

Json.NET makes working with JSON formatted data in .NET simple. Quickly read and write JSON using LINQ to JSON or serialize your .NET objects with a single method call using the JsonSerializer.

## Features

- Flexible JSON serializer to convert .NET objects to JSON and back again
- LINQ to JSON for reading and writing JSON
- Writes indented, easy to read JSON
- Convert JSON to and from XML
- Supports Silverlight and the Compact Framework

The JSON serializer is a good choice when the JSON you are reading or writing maps closely to a .NET class. The serializer automatically reads and writes JSON for the class.

For situations where you are only interested in getting values from JSON, don't have a class to serialize or deserialize to, or the JSON is radically different from your class and you need to manually read and write from your objects then LINQ to JSON is what you should use. LINQ to JSON allows you to easily read, create and modify JSON in .NET.

## History

Json.NET grew out of projects I was working on in late 2005 involving JavaScript, AJAX and .NET. At the time there were no libraries for working with JavaScript in .NET so I began to grow my own.

Starting out as a couple of static methods for escaping JavaScript strings, Json.NET evolved as features were added. To add support for reading JSON a major refactor was required and Json.NET will split into the three major classes it still uses today, JsonReader, JsonWriter and JsonSerializer.

Json.NET was first released in June 2006. Since then Json.NET has been downloaded thousands of times by developers and is used in a number of major projects open source projects including [MonoRail](#), Castle Project's MVC web framework, and [Mono](#), an open source implementation of the .NET framework.

~ *James Newton-King*

## Donate

Json.NET is a free open source project that I have developed in my personal time.

I really appreciate your feedback and support for Json.NET and its future development.



## Serializing and Deserializing JSON

The quickest method of converting between JSON text and a .NET object is using the [JsonSerializer](#). The JsonSerializer converts .NET objects into their JSON equivalent and back again.

For simple scenarios where you want to convert to and from a JSON string the [SerializeObject](#) and [DeserializeObject](#) methods on [JsonConvert](#) provide an easy to use wrapper over JsonSerializer.

```
Product product = new Product();

product.Name = "Apple";
product.Expiry = new DateTime(2008, 12, 28);
product.Price = 3.99M;
product.Sizes = new string[] { "Small", "Med

string output = JsonConvert.SerializeObject(
//{
//  "Name": "Apple",
//  "Expiry": "\/Date(1230375600000+1300)\/"
//  "Price": 3.99,
//  "Sizes": [
//    "Small",
//    "Medium",
//    "Large"
//  ]
//}

Product deserializedProduct = JsonConvert.De
```

SerializeObject and DeserializeObject both have overloads that take a [JsonSerializerSettings](#) object. JsonSerializerSettings lets you use many of the JsonSerializer settings listed below while still using the simple serialization methods.

## JsonSerializer

For more control over how an object is serialized the JsonSerializer can be used directly. The JsonSerializer is able to read and write JSON text directly to a stream via [JsonTextReader](#) and [JsonTextWriter](#). Other kinds of JsonWriters can also be used such as [JTokenReader/JTokenWriter](#) to convert your object to and from LINQ to JSON objects or [BsonReader/BsonWriter](#) to convert to and from BSON.

```
Product product = new Product();
product.Expiry = new DateTime(2008, 12, 28);

JsonSerializer serializer = new JsonSerializer()
serializer.Converters.Add(new JavaScriptDate
serializer.NullValueHandling = NullValueHand

using (StreamWriter sw = new StreamWriter(@"
using (JsonWriter writer = new JsonTextWrite
{
    serializer.Serialize(writer, product);
    // {"Expiry":new Date(1230375600000),"Pric
}
```

JsonSerializer has a number of properties on it to customize how it serializes JSON. These can also be used with the methods on JsonConvert via the JsonSerializerSettings overloads.

Read more about the available JsonSerializer settings here:

## Serialization Settings

---

JsonSerializer has a number of properties on it to customize how it serializes JSON. These can also be used with the methods on JsonConvert via the JsonSerializerSettings overloads.

## DateFormatHandling

DateFormatHandling controls how dates are serialized.

Member	Description
<b>IsoDateFormat</b>	By default Json.NET writes dates in the ISO 8601 format, e.g. "2012-03-21T05:40Z".
<b>MicrosoftDateFormat</b>	Dates are written in the Microsoft JSON format, e.g. "VDate(1198908717056)V".

## MissingMemberHandling

MissingMemberHandling controls how missing members, e.g. JSON contains a property that isn't a member on the object, are handled during deserialization.

Member	Description
<b>Ignore</b>	By default Json.NET ignores JSON if there is no field or property for its value to be set to during deserialization.
<b>Error</b>	Json.NET errors when there is a missing member during deserialization.

## ReferenceLoopHandling

ReferenceLoopHandling controls how circular referencing objects, e.g. a Person object referencing itself via a Manager property, are serialized.

Member	Description
<b>Error</b>	By default Json.NET will error if a reference loop is encountered (otherwise the serializer will get into an infinite loop).
<b>Ignore</b>	Json.NET will ignore objects in reference loops and not serialize them. The first time an object is encountered it will be serialized as usual but if the object is encountered as a child object of itself the serializer will skip serializing it.
<b>Serialize</b>	This option forces Json.NET to serialize objects in reference loops. This is useful if objects are nested but not indefinitely.

ReferenceLoopHandling can also be customized on individual properties with JsonPropertyAttribute.

## NullValueHandling

NullValueHandling controls how null values on .NET objects are handled during serialization and how null values in JSON are handled during deserialization.

Member	Description
<b>Include</b>	By default Json.NET writes null values to JSON when serializing and sets null values to fields/properties when deserializing.
<b>Ignore</b>	Json.NET will skip writing JSON properties if the .NET value is null when serializing and will skip setting fields/properties if the JSON property is null when deserializing.

NullValueHandling can also be customized on individual properties with JsonPropertyAttribute.

## DefaultValueHandling

DefaultValueHandling controls how Json.NET uses default values set using the .NET DefaultValueAttribute when serializing and deserializing.

Member	Description
<b>Include</b>	By default Json.NET will write a field/property value to JSON when serializing if the value is the same as the field/property's default value. The Json.NET deserializer will continue setting a field/property if the JSON value is the same as the default value.
<b>Ignore</b>	Json.NET will skip writing a field/property

value to JSON if the value is the same as the field/property's default value. The Json.NET deserializer will skip setting a .NET object's field/property if the JSON value is the same as the default value.

DefaultValueHandling can also be customized on individual properties with JsonPropertyAttribute.

## ObjectCreationHandling

ObjectCreationHandling controls how objects are created and deserialized to during deserialization.

Member	Description
<b>Auto</b>	By default Json.NET will attempt to set JSON values onto existing objects and add JSON values to existing collections during deserialization.
<b>Reuse</b>	Same behaviour as auto.
<b>Replace</b>	Json.NET will always recreate objects and collections before setting values to them during deserialization.

ObjectCreationHandling can also be customized on individual properties with JsonPropertyAttribute.

## TypeNameHandling

TypeNameHandling controls whether Json.NET includes .NET type names during serialization with a \$type property and reads .NET type names from that property to determine what type to create during deserialization.

<b>Member</b>	<b>Description</b>
<b>None</b>	By default Json.NET does not read or write type names during deserialization.
<b>Objects</b>	Json.NET will write and use type names for objects but not collections.
<b>Arrays</b>	Json.NET will write and use type names for collections but not objects.
<b>Auto</b>	Json.NET will check whether an object/collection matches its declared property and writes the type name if they do not match, e.g. a property with a type of Mammal has a derived instance of Dog assigned. Auto will ensure that type information isn't lost when serializing/deserializing automatically without having to write type names for every object.
<b>All</b>	Json.NET will write and use type names for objects and collections.

TypeNameHandling can also be customized on individual properties with `JsonPropertyAttribute`.

## TypeNameAssemblyFormat

TypeNameAssemblyFormat controls how type names are written during serialization.

Member	Description
<b>Simple</b>	By default Json.NET writes the partial assembly name with the type, e.g. <code>System.Data.DataSet</code> , <code>System.Data</code> . Note that Silverlight and Windows Phone are not able to use this format.
<b>Full</b>	Json.NET will write the full assembly name, including version number, culture and public key token.

Read more about the valid values at [FormatterAssemblyStyle](#).

## Binder

The serialization binder is used to resolve type names to a .NET type.

Read more about the serialization binder here: [SerializationBinder](#)

## ConstructorHandling

ConstructorHandling controls how constructors are used when initializing objects during deserialization.

Member	Description
--------	-------------

<b>Default</b>	By default Json.NET will first look for a constructor marked with the <code>JsonConstructorAttribute</code> , then look for a public default constructor (a constructor that doesn't take any arguments), then check if the class has a single public constructor with arguments and finally check for a non-public default constructor. If the class has multiple public constructors with arguments an error will be thrown. This can be fixed by marking one of the constructors with the <code>JsonConstructorAttribute</code> .
<b>AllowNonPublicDefaultConstructor</b>	Json.NET will use a classes private default constructor before constructors with arguments if available.

## Converters

This is the collection of `JsonConverters` that will be used during serialization and deserialization.

`JsonConverters` allows JSON to be manually written during serialization and read during deserialization. This is useful for particularly complex JSON structures or for when you want to change how a type is serialized.

When a `JsonConverter` has been added to a `JsonSerializer` it will be checked for every value that is being serialized/deserialized using its `CanConvert` to see if it should be used. If `CanConvert` returns true then the `JsonConverter` will be used to read or write the JSON for that value. Note that while a `JsonConverter` gives you complete control over that values JSON, many `Json.NET` serialization features are no longer available like type name and reference handling.

To create your own custom converter inherit from the `JsonConverter` class. Read more about the built-in `JsonConverters` below:

- [Serializing Dates in JSON](#)
- [Converting between JSON and XML](#)
- [CustomCreationConverter](#)
- [StringEnumConverter](#)

## ContractResolver

Internally for every .NET type the `JsonSerializer` will create a contract of how the type should be serialized and deserialized, based on type metadata and attributes applied to the class. Specifying a custom `IContractResolver` allows the creation of contracts to be customized.

Read more about Contract Resolvers here: [Contract Resolvers](#)

## Error

The `Error` event can catch errors during serialization and either handle the event and continue with serialization or let the error bubble up and be thrown to the application.

Read more about error handling here: [Serialization Error Handling](#)

---

The Json.NET serializer can serialize a wide variety of .NET objects. This guide looks at how it works at a high level and in more detail.

## Summary

At a high level, the Json.NET serializer will convert primitive .NET values into primitive JSON values, .NET arrays and collections to JSON arrays and everything else to JSON objects.

Json.NET will throw an error if it encounters incorrect JSON when deserializing a value. For example if the serializer encounters a JSON property with an array of values and the type of matching .NET property is not a collection then an error will be thrown, and vice-versa.

## Complex Types

<b>.NET</b>	<b>JSON</b>
<b>ICollection, IEnumerable, IList&lt;T&gt;, Array</b>	Array
<b>IDictionary, IDictionary&lt;TKey, TValue&gt;</b>	Object
<b>Object (more detail below)</b>	Object

## Primitive Types

<b>.NET</b>	<b>JSON</b>
<b>String</b>	String

<b>Byte</b> <b>SByte</b> <b>UInt16</b> <b>Int16</b> <b>UInt32</b> <b>Int32</b> <b>UInt64</b> <b>Int64</b>	Integer
<b>Float</b> <b>Double</b> <b>Decimal</b>	Float
<b>Enum</b>	Integer (can be string with <a href="#">StringEnumConverter</a> )
<b>DateTime</b>	String ( <a href="#">Serializing Dates in JSON</a> )
<b>Byte[]</b>	String (base 64 encoded)
<b>Type</b>	String (type name)
<b>Guid</b>	String
<b><a href="#">TypeConverter</a></b> <b>(convertible to String)</b>	String

# Breakdown of Type Serialization

## Objects

.NET types that don't fall into any other category listed below (i.e. aren't lists, dictionaries, dynamic, implement ISerializable, etc) are serialized as JSON objects.

By default types are serialized in opt-out mode. What that means is all public fields and properties with getters are automatically serialized to JSON and members that shouldn't be serialized are opted-out using the `JsonIgnoreAttribute`. To serialize private members the `JsonPropertyAttribute` can be placed on private fields and properties or the `DefaultMembersSearchFlags` can be changed on `DefaultContractResolver` to change how members are serialized on all types.

Types can also be serialized using opt-in mode. Only properties and fields that have a `JsonPropertyAttribute` or `DataMemberAttribute` on them will be serialized. Opt-in mode for an object is specified using the `JsonObjectAttribute` on the class.

Finally types can be serialized using a fields mode. All fields, both public and private, are serialized and properties are ignored. This can be specified by setting `MemberSerialization.Fields` on a type with the `JsonObject` attribute or by the .NET [Serializable](#) attribute and setting `IgnoreSerializableAttribute` on `DefaultContractResolver` to false.

## IEnumerable, Lists and Arrayss

.NET lists (types that inherit from `IEnumerable`) and .NET arrays are converted to JSON arrays. Because JSON arrays only support a range of values and not properties, any additional properties and fields declared on .NET collections are not serialized. In situations where a JSON array is not wanted the `JsonObjectAttribute` can be placed on a .NET type that implements `IEnumerable` to force the type to be serialized as a JSON object instead.

JsonArrayAttribute has options on it to customize the JsonConverter, type name handling and reference handling that are applied to collection items.

Note that if TypeNameHandling or PreserveReferencesHandling has been enabled for JSON arrays on the serializer then JSON arrays are wrapped in a containing object. The object will have the type name/reference properties and a \$values property which will have the collection data.

When deserializing if a member is typed as the interface IList<T> then it will be deserialized as a List<T>.

Read more about serializing collections here: [Serializing Collections](#)

## Dictionarys and Hashtables

.NET dictionaries (types that inherit from IDictionary) are converted to JSON objects. Note that only the dictionary name/values will be written to the JSON object when serializing and properties on the JSON object will be added to the dictionary's name/values when deserializing. Additional members on the .NET dictionary are ignored during serialization.

JsonDictionaryAttribute has options on it to customize the JsonConverter, type name handling and reference handling that are applied to collection items.

When deserializing if a member is typed as the interface IDictionary<TKey, TValue> then it will be deserialized as a Dictionary<TKey, TValue>.

Read more about serializing collections here: [Serializing Collections](#)

## Untyped Objects

.NET properties on a class that don't specify a type (i.e. they are just object) are serialized as usual. When untyped properties are

deserialized the serializer has no way to know what type to create (unless type name handling is enabled and the JSON contains the type names).

For these untyped properties the Json.NET serializer will read the JSON into LINQ to JSON objects and set them to the property. JObject will be created for JSON objects, JArray will be created for JSON arrays and JValue for primitive JSON values.

## Dynamic

There are two different usages of dynamic (introduced in .NET 4) in .NET. The first are .NET properties with a type of dynamic. Dynamic properties behave like properties declared as object, any value can be assigned to it, but the difference being that properties and methods can be called on a dynamic property without casting. In Json.NET dynamic properties are serialized and deserialized exactly the same as untyped objects: because dynamic isn't an actual type Json.NET falls back to deserializing the JSON as LINQ to JSON objects.

The second usage of dynamic in .NET are classes that implement [IDynamicMetaObjectProvider](#). This interface lets the implementor create dynamic objects that intercept the property and method calls on an object and use them. [ExpandoObject](#) is a good example of a dynamic object.

Dynamic objects are serialized as JSON objects. A property is written for every member name returned by [DynamicMetaObject.GetDynamicMemberNames](#).

When deserializing dynamic objects the serializer first attempts to set JSON property values on a normal .NET member with the matching name. If no .NET member is found with the property name then the serializer will call SetMember on the dynamic object. Because there is no type information for dynamic members on a dynamic object the values assigned to them will be LINQ to JSON objects

## **ISerializable**

Types that implement `ISerializable` are serialized as JSON objects. When serializing only the values returned from `ISerializable.GetObjectData` are used; members on the type are ignored. When deserializing the constructor with a `SerializationInfo` and `StreamingContext` is called, passing the JSON object's values.

In situations where this behavior is not wanted the `JsonObjectAttribute` can be placed on a .NET type that implements `ISerializable` to force it to be serialized as a normal JSON object.

## **LINQ to JSON**

LINQ to JSON types (e.g. `JObject`, `JArray`) are automatically serialized and deserialized to their equivalent JSON when encountered by the `Json.NET` serializer.

## **JsonConverter**

Serialization of values that are convertible by a `JsonConverter` (i.e. `CanConvert` returns true for its type) is completely overridden by the `JsonConverter`. The test to see whether a value can be converted by a `JsonSerializer` takes precedence over all other tests.

`JsonConverters` can be defined and specified in a number of places: in an attribute on a member, in an attribute on a class and added to the `JsonSerializer`'s converters collection. The priority of which `JsonConverter` is used is the `JsonConverter` defined by attribute on a member then the `JsonConverter` defined by an attribute on a class and finally any converters passed to the `JsonSerializer`.

---

Attributes can be used to control how Json.NET serializes and deserializes .NET objects.

- [JsonObjectAttribute](#) - Placed on classes to control how it should be serialized as a JSON object.
- [JsonArrayAttribute](#) - Placed on collections to control how it should be serialized as a JSON array.
- [JsonDictionaryAttribute](#) - Placed on dictionaries to control how it should be serialized as a JSON object.
- [JsonPropertyAttribute](#) - Placed on fields and properties to control how it should be serialized as a property in a JSON object.
- [JsonConverterAttribute](#) - Placed on either classes or fields and properties to specify which `JsonConverter` should be used during serialization.

As well as using the built-in Json.NET attributes, Json.NET also looks for the [Serializable](#) (if `IgnoreSerializableAttribute` on `DefaultContractResolver` is set to `false`) [DataContract](#), [DataMember](#) and [NotSerialized](#) and attributes when determining how JSON is to be serialized and deserialized. If both are present the Json.NET serialization attributes take precedence.

```
[JsonObject(MemberSerialization.OptIn)]
public class Person
{
    // "John Smith"
    [JsonProperty]
    public string Name { get; set; }

    // "2000-12-15T22:11:03"
    [JsonProperty]
    public DateTime BirthDate { get; set; }
}
```

```
// new Date(976918263055)
[JsonProperty]
[JsonConverter(typeof(JavaScriptDateTimeCo
public DateTime LastModified { get; set; }

// not serialized because mode is opt-in
public string Department { get; set; }
}
```

### JsonObjectAttribute

The MemberSerialization flag on this attribute specifies whether member serialization is opt-in (a member must have the JsonProperty or DataMember attribute to be serialized), opt-out (everything is serialized by default but can be ignored with the.JsonIgnoreAttribute, Json.NET's default behavior) or fields (all public and private fields are serialized, properties are ignored).

Json.NET serializes .NET classes that implement IEnumerable as an JSON array populated with the IEnumerable values. Placing the JsonPropertyAttribute overrides this behavior and forces the serializer to serialize the class's fields and properties.

The DataContractAttribute can be used as substitute for JsonObjectAttribute. The DataContractAttribute will default member serialization to opt-in.

### JsonArrayAttribute/JsonDictionaryAttribute

The JsonArrayAttribute and JsonDictionaryAttributes are used to specified whether a class is serialized as that collection type.

The collection attributes have options to customize the JsonConverter, type name handling and reference handling that are applied to collection items.

## **JsonPropertyAttribute**

JsonPropertyAttribute has a number of uses:

- By default the JSON property will have the same name as the .NET property. This attribute allows the name to be customized.
- Indicates that a property should be serialized when member serialization is set to opt-in.
- Includes non-public properties in serialization and deserialization.
- Customize type name, reference, null and default value handling for the property value.
- Customize the property's collection items JsonSerializer, type name handling and reference handling.

The DataMemberAttribute can be used as substitute for JsonPropertyAttribute.

## **JsonIgnoreAttribute**

Excludes a field or property from serialization.

The NotSerializedAttribute can be used as substitute for JsonIgnoreAttribute.

## **JsonConverterAttribute**

The JsonConverterAttribute specifies which JsonSerializer is used to convert an object.

The attribute can be placed on a class or a member. When placed on a class the JsonSerializer specified by the attribute will be the default way of serializing that class. When the attribute is on a field or property then the specified JsonSerializer will always be used to serialize that value.

The priority of which JsonSerializer is used is member attribute then class attribute and finally any converters passed to the JsonSerializer.

```
public class MemberConverterClass
{
    public DateTime DefaultConverter { get; se
    [JsonConverter(typeof(IsoDateTimeConverter
    public DateTime MemberConverter { get; set
}
```

This example shows the `JsonConverterAttribute` being applied to a property.

```
DateTime date = Convert.ToDateTime("1970-01-

MemberConverterClass c = new MemberConverter
{
    DefaultConverter = date,
    MemberConverter = date
};

string json = JsonConvert.SerializeObject(c,

Console.WriteLine(json);
//{
//  "DefaultConverter": "\/Date(0)\/",
//  "MemberConverter": "1970-01-01T00:00:00Z
//}
```

To apply a `JsonConverter` to the items in a collection use either [JsonArrayAttribute](#), [JsonDictionaryAttribute](#) or [JsonPropertyAttribute](#) and set the `ItemConverterType` property to the converter type you want to use.

---

Json.NET supports serialization callback methods. A callback can be used to manipulate an object before and after its serialization and deserialization by the JsonSerializer.

- **OnSerializing**
- **OnSerialized**
- **OnDeserializing**
- **OnDeserialized**

To tell the serializer which methods should be called during the object's serialization lifecycle, decorate a method with the appropriate attribute ([OnSerializingAttribute](#), [OnSerializedAttribute](#), [OnDeserializingAttribute](#), [OnDeserializedAttribute](#)).

Example object with serialization callback methods:

```
public class SerializationEventTestObject
{
    // This member is serialized and deseriali
    public int Member1 { get; set; }

    // The value of this field is set and rese
    // after serialization.
    public string Member2 { get; set; }

    // This field is not serialized. The OnDes
    // is used to set the member value after s
    [JsonIgnore]
    public string Member3 { get; set; }

    // This field is set to null, but populate
    public string Member4 { get; set; }
```

```
public SerializationEventTestObject()
{
    Member1 = 11;
    Member2 = "Hello World!";
    Member3 = "This is a nonserialized value";
    Member4 = null;
}

[OnSerializing]
internal void OnSerializingMethod(StreamingSerializer s)
{
    Member2 = "This value went into the data stream";
}

[OnSerialized]
internal void OnSerializedMethod(StreamingSerializer s)
{
    Member2 = "This value was reset after serialization";
}

[OnDeserializing]
internal void OnDeserializingMethod(StreamingDeserializer d)
{
    Member3 = "This value was set during deserialization";
}

[OnDeserialized]
internal void OnDeserializedMethod(StreamingDeserializer d)
{
    Member4 = "This value was set after deserialization";
}
}
```

---

The example object being serialized and deserialized by Json.NET:

```
SerializationEventTestObject obj = new Serializ

Console.WriteLine(obj.Member1);
// 11
Console.WriteLine(obj.Member2);
// Hello World!
Console.WriteLine(obj.Member3);
// This is a nonserialized value
Console.WriteLine(obj.Member4);
// null

string json = JsonConvert.SerializeObject(obj)
// {
//   "Member1": 11,
//   "Member2": "This value went into the da
//   "Member4": null
// }

Console.WriteLine(obj.Member1);
// 11
Console.WriteLine(obj.Member2);
// This value was reset after serialization.
Console.WriteLine(obj.Member3);
// This is a nonserialized value
Console.WriteLine(obj.Member4);
// null

obj = JsonConvert.DeserializeObject<Serializ

Console.WriteLine(obj.Member1);
```

```
// 11
Console.WriteLine(obj.Member2);
// This value went into the data file during
Console.WriteLine(obj.Member3);
// This value was set during deserialization
Console.WriteLine(obj.Member4);
// This value was set after deserialization.
```



## Serialization Error Handling

Json.NET supports error handling during serialization and deserialization. Error handling lets you catch an error and choose whether to handle it and continue with serialization or let the error bubble up and be thrown in your application.

Error handling is defined through two methods: the Error event on `JsonSerializer` and the `OnErrorAttribute`.

### Error Event

The `Error` event is an event handler found on `JsonSerializer`. The error event is raised whenever an exception is thrown while serializing or deserializing JSON. Like all settings found on `JsonSerializer` it can also be set on `JsonSerializerSettings` and passed to the serialization methods on `JsonConvert`.

```
List<string> errors = new List<string>();

List<DateTime> c = JsonConvert.DeserializeOb
    ""2009-09-09T00:00:00Z"",
    ""I am not a date and will error!"",
    [
        1
    ],
    ""1977-02-20T00:00:00Z"",
    null,
    ""2000-12-01T00:00:00Z""
    ]",
    new JsonSerializerSettings
    {
        Error = delegate(object sender, ErrorE
```

```
        {
            errors.Add(args.ErrorContext.Error
                args.ErrorContext.Handled = true;
        },
        Converters = { new IsoDateTimeConverte
    });
```

```
// 2009-09-09T00:00:00Z
```

```
// 1977-02-20T00:00:00Z
```

```
// 2000-12-01T00:00:00Z
```

```
// The string was not recognized as a valid
// Unexpected token parsing date. Expected S
// Cannot convert null value to System.DateT
```

In this example we are deserializing a JSON array to a collection of DateTimes. On the JsonSerializerSettings a handler has been assigned to the Error event which will log the error message and mark the error as handled.

The result of deserializing the JSON is three successfully deserialized dates and three error messages: one for the badly formatted string, "I am not a date and will error!", one for the nested JSON array and one for the null value since the list doesn't allow nullable DateTimes. The event handler has logged these messages and Json.NET has continued on deserializing the JSON because the errors were marked as handled.

One thing to note with error handling in Json.NET is that an unhandled error will bubble up and raise the event on each of its parents, e.g. an unhandled error when serializing a collection of objects will be raised twice, once against the object and then again on the collection. This will let you handle an error either where it occurred or on one of its parents.

```
JsonSerializer serializer = new JsonSerializer()
serializer.Error += delegate(object sender,
{
    // only log an error once
    if (args.CurrentObject == args.ErrorContext.OriginalObject)
        errors.Add(args.ErrorContext.Error.Message);
};
```

If you aren't immediately handling an error and only want to perform an action against it once then you can check to see whether the [ErrorEventArgs](#)'s `CurrentObject` is equal to the `OriginalObject`. `OriginalObject` is the object that threw the error and `CurrentObject` is the object that the event is being raised against. They will only equal the first time the event is raised against the `OriginalObject`.

## OnErrorAttribute

The [OnErrorAttribute](#) works much like the other [.NET serialization attributes](#) that Json.NET supports. To use it you simply place the attribute on a method which takes the correct parameters: a `StreamingContext` and a `ErrorContext`. The name of the method doesn't matter.

```
public class PersonError
{
    private List<string> _roles;

    public string Name { get; set; }
    public int Age { get; set; }
    public List<string> Roles
    {
        get
        {
```

```

        if (_roles == null)
            throw new Exception("Roles not loaded");

        return _roles;
    }
    set { _roles = value; }
}
public string Title { get; set; }

[OnError]
internal void OnError(StreamingContext context)
{
    context.ErrorContext.Handled = true;
}
}

```

In this example accessing the the Roles property will throw an exception when no roles have been set. The HandleError method will set the error when serializing Roles as handled and allow Json.NET to continue serializing the class.

```

PersonError person = new PersonError
{
    Name = "George Michael Bluth",
    Age = 16,
    Roles = null,
    Title = "Mister Manager"
};

string json = JsonConvert.SerializeObject(person);
Console.WriteLine(json);

```

```
//{  
//  "Name": "George Michael Bluth",  
//  "Age": 16,  
//  "Title": "Mister Manager"  
//}
```



## Serialization and Preserving Object References

By default Json.NET will serialize all objects it encounters by value. If a list contains two Person references, and both references point to the same object then the JsonSerializer will write out all the names and values for each reference.

```
Person p = new Person
{
    BirthDate = new DateTime(1980, 12, 23, 0
    LastModified = new DateTime(2009, 2, 20,
    Name = "James"
};

List<Person> people = new List<Person>();
people.Add(p);
people.Add(p);

string json = JsonConvert.SerializeObject(pe
//[
// {
//     "Name": "James",
//     "BirthDate": "\/Date(346377600000)\/",
//     "LastModified": "\/Date(1235134761000)
// },
// {
//     "Name": "James",
//     "BirthDate": "\/Date(346377600000)\/",
//     "LastModified": "\/Date(1235134761000)
// }
//]
```



In most cases this is the desired result but in certain scenarios writing the second item in the list as a reference to the first is a better solution. If the above JSON was deserialized now then the returned list would contain two completely separate Person objects with the same values. Writing references by value will also cause problems on objects where a circular reference occurs.

## PreserveReferencesHandling

Settings PreserveReferencesHandling will track object references when serializing and deserializing JSON.

```
string json = JsonConvert.SerializeObject(pe
    new JsonSerializerSettings { PreserveRefer
//[
// {
//     "$id": "1",
//     "Name": "James",
//     "BirthDate": @"\Date(346377600000)\/",
//     "LastModified": @"\Date(1235134761000)
// },
// {
//     "$ref": "1"
// }
//]

List<Person> deserializedPeople = JsonConver
    new JsonSerializerSettings { PreserveRefer

Console.WriteLine(deserializedPeople.Count);
// 2

Person p1 = deserializedPeople[0];
```

```
Person p2 = deserializedPeople[1];

Console.WriteLine(p1.Name);
// James
Console.WriteLine(p2.Name);
// James

bool equal = Object.ReferenceEquals(p1, p2);
// true
```

The first Person in the list is serialized with the addition of an object Id. The second Person in JSON is now only a reference to the first.

With PreserveReferencesHandling on now only one Person object is created on deserialization and the list contains two references to it, mirroring what we started with.

## IsReference on JsonObjectAttribute, JsonArrayAttribute and JsonPropertyAttribute

The PreserveReferencesHandling setting on the JsonSerializer will change how all objects are serialized and deserialized. For fine grain control over which objects and members should be serialized as a reference there is the IsReference property on the JsonObjectAttribute, JsonArrayAttribute and JsonPropertyAttribute.

Setting IsReference on JsonObjectAttribute or JsonArrayAttribute to true will mean the JsonSerializer will always serialize the type the attribute is against as a reference. Setting IsReference on the JsonPropertyAttribute to true will serialize only that property as a reference.

```
[JsonObject(IsReference = true)]
public class EmployeeReference
```

```
{  
    public string Name { get; set; }  
    public EmployeeReference Manager { get; se  
}
```

## **IReferenceResolver**

To customize how references are generated and resolved the [IReferenceResolver](#) interface is available to inherit from and use with the JsonSerializer.

---

## CustomCreationConverter

The CustomCreationConverter is a JsonSerializer that provides a way to customize how an object is created during JSON deserialization. Once the object has been created it will then have values populated onto it by the serializer.

```
public interface IPerson
{
    string FirstName { get; set; }
    string LastName { get; set; }
    DateTime BirthDate { get; set; }
}

public class Employee : IPerson
{
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public DateTime BirthDate { get; set; }

    public string Department { get; set; }
    public string JobTitle { get; set; }
}

public class PersonConverter : CustomCreatio
{
    public override IPerson Create(Type object
    {
        return new Employee();
    }
}
```

This is an extremely simple example. A more complicated scenario could involve an object factory or service locator which resolves the object at runtime.

```
//[  
//  {  
//    "FirstName": "Maurice",  
//    "LastName": "Moss",  
//    "BirthDate": "\/Date(252291661000)\/",  
//    "Department": "IT",  
//    "JobTitle": "Support"  
//  },  
//  {  
//    "FirstName": "Jen",  
//    "LastName": "Barber",  
//    "BirthDate": "\/Date(258771661000)\/",  
//    "Department": "IT",  
//    "JobTitle": "Manager"  
//  }  
//]
```

```
List<IPerson> people = JsonConvert.Deseriali
```

```
IPerson person = people[0];
```

```
Console.WriteLine(person.GetType());  
// Newtonsoft.Json.Tests.Employee
```

```
Console.WriteLine(person.FirstName);  
// Maurice
```

```
Employee employee = (Employee)person;
```

```
Console.WriteLine(employee.JobTitle);  
// Support
```



The [IContractResolver](#) interface provides a way to customize how the JsonSerializer serializes and deserializes .NET objects to JSON.

Implementing the IContractResolver interface and then assigning an instance to a JsonSerializer lets you control whether the object is serialized as a JSON object or JSON array, what object members should be serialized, how they are serialized and what they are called.

## DefaultContractResolver

The [DefaultContractResolver](#) is the default resolver used by the serializer. It provides many avenues of extensibility in the form of virtual methods that can be overridden.

## CamelCasePropertyNamesContractResolver

[CamelCasePropertyNamesContractResolver](#) inherits from DefaultContractResolver and simply overrides the JSON property name to be written in [camelcase](#).

```
Product product = new Product
    {
        ExpiryDate = new DateTime
        Name = "Widget",
        Price = 9.99m,
        Sizes = new[] { "Small"
    };

string json =
    JsonConvert.SerializeObject(
        product,
        Formatting.Indented,
```

```
new JsonSerializerSettings { ContractRes  
);
```

```
//{  
// "name": "Widget",  
// "expiryDate": @"\Date(1292868060000)\",  
// "price": 9.99,  
// "sizes": [  
//     "Small",  
//     "Medium",  
//     "Large"  
// ]  
//}
```



The [JsonSerializer](#) has great support for serializing and deserializing collections of objects.

## Serializing

To serialize a collection - a generic list, array, dictionary, or your own custom collection - simply call the serializer with the object you want to get JSON for. Json.NET will serialize the collection and all of the values it contains.

```
Product p1 = new Product
{
    Name = "Product 1",
    Price = 99.95m,
    ExpiryDate = new DateTime(2000, 12, 29,
};
Product p2 = new Product
{
    Name = "Product 2",
    Price = 12.50m,
    ExpiryDate = new DateTime(2009, 7, 31, 0,
};

List<Product> products = new List<Product>()
products.Add(p1);
products.Add(p2);

string json = JsonConvert.SerializeObject(pr
//[
// {
//     "Name": "Product 1",
```

```
//      "ExpiryDate": "\/Date(978048000000)\/"
//      "Price": 99.95,
//      "Sizes": null
//    },
//    {
//      "Name": "Product 2",
//      "ExpiryDate": "\/Date(1248998400000)\/"
//      "Price": 12.50,
//      "Sizes": null
//    }
//  ]
```

## Deserializing

To deserialize JSON into a .NET collection just specify the collection type you want to deserialize to. Json.NET supports a wide range of collection types.

```
string json = @"[
  {
    ""Name"": ""Product 1"",
    ""ExpiryDate"": ""\/Date(978048000000)\/"
    ""Price"": 99.95,
    ""Sizes"": null
  },
  {
    ""Name"": ""Product 2"",
    ""ExpiryDate"": ""\/Date(1248998400000)\/"
    ""Price"": 12.50,
    ""Sizes"": null
  }
]";
```

```
List<Product> products = JsonConvert.Deseria  
  
Console.WriteLine(products.Count);  
// 2  
  
Product p1 = products[0];  
  
Console.WriteLine(p1.Name);  
// Product 1
```

## Deserializing Dictionaries

Using Json.NET you can also deserialize a JSON object into a .NET generic dictionary. The JSON object's property names and values will be added to the dictionary.

```
string json = @"{"key1": "value1", "key2"  
  
Dictionary<string, string> values = JsonConv  
  
Console.WriteLine(values.Count);  
// 2  
  
Console.WriteLine(values["key1"]);  
// value1
```

## Serializing Dates in JSON

DateTimes in JSON are hard.

The problem comes from the [JSON spec](#) itself: there is no literal syntax for dates in JSON. The spec has objects, arrays, strings, integers and floats, but it defines no standard for what a date looks like.

The default format used by [Json.NET](#) is the ISO 8601 standard: 2012-03-19T07:22Z. You can read more about it [here](#).

Prior to [Json.NET 4.5](#) dates were written using the Microsoft format: "VDate(1198908717056)V". If you want to use this format, or you want to maintain compatibility with Microsoft JSON serializers or older versions of [Json.NET](#) then change the [DateFormatHandling](#) setting to [MicrosoftDateFormat](#).

[Json.NET](#) also has the [DateTimeZoneHandling](#) setting. This can be used to convert [DateTime](#)'s kind when serializing, e.g. set [DateTimeZoneHandling](#) to [Utc](#) to serialize all [DateTimes](#) as UTC dates.

### DateTime JsonConverters

With no standard for dates in JSON, the number of possible different formats when interoping with other systems is endless. Fortunately [Json.NET](#) has a solution to deal with reading and writing custom dates: [JsonConverters](#). A [JsonConverter](#) is used to override how a type is serialized.

```
public class LogEntry
{
    public string Details { get; set; }
    public DateTime LogDate { get; set; }
}
```

```

[Test]
public void WriteJsonDates()
{
    LogEntry entry = new LogEntry
    {
        LogDate = new DateTime(2009, 2, 15, 0, 0)
        Details = "Application started."
    };

    // default as of Json.NET 4.5
    string isoJson = JsonConvert.SerializeObject(
    // {"Details":"Application started.,"LogD

    JsonSerializerSettings microsoftDateFormat
    {
        DateFormatHandling = DateFormatHandling.
    };
    string microsoftJson = JsonConvert.Seriali
    // {"Details":"Application started.,"LogD

    string javascriptJson = JsonConvert.Serial
    // {"Details":"Application started.,"LogD
}

```

Simply pass the `JsonConverter` you wish to use to the `Json.NET` serializer.

### JavaScriptDateTimeConverter

The `JavaScriptDateTimeConverter` class is one of the two `DateTime` `JsonConverters` that come with `Json.NET`. This converter serializes a `DateTime` as a [JavaScript Date object](#).

```
new Date(1234656000000)
```

Technically this is invalid JSON according to the spec but all browsers, and some JSON frameworks, including Json.NET, support it.

## **IsoDateTimeConverter**

IsoDateTimeConverter serializes a DateTime to an [ISO 8601](#) formatted string. Note that since Json.NET 4.5 dates are written using the ISO 8601 format by default and using this converter is unnecessary.

```
"2009-02-15T00:00:00Z"
```

The IsoDateTimeConverter class has a property, DateTimeFormat, to further customize the formatted string.

---

## Reducing Serialized JSON Size

One of the common problems encountered when serializing .NET objects to JSON is that the JSON ends up containing a lot of unwanted properties and values. This can be especially important when returning JSON to the client. More JSON means more bandwidth and a slower website.

To solve the issue of unwanted JSON Json.NET has a range of built in options to fine tune what gets written from a serialized object.

### JsonIgnoreAttribute and DataMemberAttribute

By default Json.NET will include all of a classes public properties and fields in the JSON it creates. Adding the [JsonIgnoreAttribute](#) to a property tells the serializer to always skip writing it to the JSON result.

```
public class Car
{
    // included in JSON
    public string Model { get; set; }
    public DateTime Year { get; set; }
    public List<string> Features { get; set; }

    // ignored
    [JsonIgnore]
    public DateTime LastModified { get; set; }
}
```

If a class has many properties and you only want to serialize a small subset of them then adding `JsonIgnore` to all the others will be tedious and error prone. The way to tackle this scenario is to add the [DataContractAttribute](#) to the class and `DataMemberAttributes` to the

properties to serialize. This is opt-in serialization, only the properties you mark up with be serialized, compared to opt-out serialization using `JsonIgnoreAttribute`.

```
[DataContract]
public class Computer
{
    // included in JSON
    [DataMember]
    public string Name { get; set; }
    [DataMember]
    public decimal SalePrice { get; set; }

    // ignored
    public string Manufacture { get; set; }
    public int StockCount { get; set; }
    public decimal WholesalePrice { get; set; }
    public DateTime NextShipmentDate { get; se
}
```

## Formatting

JSON written by the serializer with an option of `Formatting.Indented` produces nicely formatted, easy to read JSON – great when you are developing. `Formatting.None` on the other hand keeps the JSON result small, skipping all unnecessary spaces and line breaks to produce the most compact and efficient JSON possible.

## NullValueHandling

`NullValueHandling` is an option on the `JsonSerializer` and controls how the serializer handles properties with a null value. By setting a value of `NullValueHandling.Ignore` the `JsonSerializer` skips writing any properties that have a value of null.

```
public class Movie
{
    public string Name { get; set; }
    public string Description { get; set; }
    public string Classification { get; set; }
    public string Studio { get; set; }
    public DateTime? ReleaseDate { get; set; }
    public List<string> ReleaseCountries { get
}
}
```

```
Movie movie = new Movie();
movie.Name = "Bad Boys III";
movie.Description = "It's no Bad Boys";

string included = JsonConvert.SerializeObject(
    Formatting.Indented,
    new JsonSerializerSettings { });

// {
//   "Name": "Bad Boys III",
//   "Description": "It's no Bad Boys",
//   "Classification": null,
//   "Studio": null,
//   "ReleaseDate": null,
//   "ReleaseCountries": null
// }

string ignored = JsonConvert.SerializeObject(
    Formatting.Indented,
    new JsonSerializerSettings { NullValueHand
```

```
// {  
//   "Name": "Bad Boys III",  
//   "Description": "It's no Bad Boys"  
// }
```

NullValueHandling can also be customized on individual properties using the a [JsonPropertyAttribute](#). The JsonPropertyAttribute value of NullValueHandling will override the setting on the JsonSerializer for that property.

## DefaultValueHandling

[DefaultValueHandling](#) is an option on the JsonSerializer and controls how the serializer handles properties with a default value. Setting a value of DefaultValueHandling.Ignore will make the JsonSerializer skip writing any properties that have a default value to the JSON result. For object references this will be null. For value types like int and DateTime the serializer will skip the default uninitialized value for that value type.

Json.NET also allows you to customize what the default value of an individual property is using the [DefaultValueAttribute](#). For example if a string property called Department always returns an empty string in its default state and you didn't want that empty string in your JSON then placing the DefaultValueAttribute on Department with that value will mean Department is no longer written to JSON unless it has a value.

```
public class Invoice  
{  
    public string Company { get; set; }  
    public decimal Amount { get; set; }  
  
    // false is default value of bool  
    public bool Paid { get; set; }  
    // null is default value of nullable
```

```
public DateTime? PaidDate { get; set; }

// customize default values
[DefaultValue(30)]
public int FollowUpDays { get; set; }
[DefaultValue("")]
public string FollowUpEmailAddress { get;
}
}
```

```
Invoice invoice = new Invoice
{
    Company = "Acme Ltd.",
    Amount = 50.0m,
    Paid = false,
    FollowUpDays = 30,
    FollowUpEmailAddress = string.Empty,
    PaidDate = null
};

string included = JsonConvert.SerializeObject(
    Formatting.Indented,
    new JsonSerializerSettings { });

// {
//   "Company": "Acme Ltd.",
//   "Amount": 50.0,
//   "Paid": false,
//   "PaidDate": null,
//   "FollowUpDays": 30,
//   "FollowUpEmailAddress": ""
// }
```

```
string ignored = JsonConvert.SerializeObject
    Formatting.Indented,
    new JsonSerializerSettings { DefaultValueH

// {
//     "Company": "Acme Ltd.",
//     "Amount": 50.0
// }
```

DefaultValueHandling can also be customized on individual properties using the a [JsonPropertyAttribute](#). The JsonPropertyAttribute value of DefaultValueHandling will override the setting on the JsonSerializer for that property.

## IContractResolver

For more flexibility the [IContractResolver](#) provides an interface to customize almost every aspect of how a .NET object gets serialized to JSON, including changing serialization behavior at runtime.

```
public class DynamicContractResolver : Defau
{
    private readonly char _startingWithChar;
    public DynamicContractResolver(char starti
    {
        _startingWithChar = startingWithChar;
    }

    protected override IList<JsonProperty> Cre
    {
        IList<JsonProperty> properties = base.Cr
```

```
// only serializer properties that start
properties =
    properties.Where(p => p.PropertyName.S

return properties;
}
}

public class Book
{
    public string BookName { get; set; }
    public decimal BookPrice { get; set; }
    public string AuthorName { get; set; }
    public int AuthorAge { get; set; }
    public string AuthorCountry { get; set; }
}
```

```
Book book = new Book
{
    BookName = "The Gathering St
    BookPrice = 16.19m,
    AuthorName = "Brandon Sander
    AuthorAge = 34,
    AuthorCountry = "United Stat
};

string startingWithA = JsonConvert.Serialize
    new JsonSerializerSettings { ContractResol

// {
//     "AuthorName": "Brandon Sanderson",
```

```
// "AuthorAge": 34,  
// "AuthorCountry": "United States of Amer  
// }
```

```
string startingWithB = JsonConvert.Serialize  
    new JsonSerializerSettings { ContractResol
```

```
// {  
// "BookName": "The Gathering Storm",  
// "BookPrice": 16.19  
// }
```



## Serializing Partial JSON Fragments

Often when working with large JSON documents you're only interested in a small fragment of information. This scenario can be annoying when you want to serialize that JSON into .NET objects because you have to define .NET classes for the entire JSON result.

With Json.NET it is easy to get around this problem. Using LINQ to JSON you can extract the pieces of JSON you want to serialize before passing them to the Json.NET serializer.

```
public class SearchResult
{
    public string Title { get; set; }
    public string Content { get; set; }
    public string Url { get; set; }
}
```

```
string googleSearchText = @"{
  ""responseData"": {
    ""results"": [
      {
        ""GsearchResultClass"": ""GwebSearch
        ""unescapedUrl"": ""http://en.wikipe
        ""url"": ""http://en.wikipedia.org/w
        ""visibleUrl"": ""en.wikipedia.org""
        ""cacheUrl"": ""http://www.google.co
        ""title"": ""<b>Paris Hilton</b> - W
        ""titleNoFormatting"": ""Paris Hilto
        ""content"": ""[1] In 2006, she rele
      },
```

```
{
  "GsearchResultClass": "GwebSearch
  "unescapedUrl": "http://www.imdb.
  "url": "http://www.imdb.com/name/
  "visibleUrl": "www.imdb.com",
  "cacheUrl": "http://www.google.co
  "title": "<b>Paris Hilton</b>",
  "titleNoFormatting": "Paris Hilto
  "content": "Self: Zoolander. Soci
}
],
"cursor": {
  "pages": [
    {
      "start": "0",
      "label": 1
    },
    {
      "start": "4",
      "label": 2
    },
    {
      "start": "8",
      "label": 3
    },
    {
      "start": "12",
      "label": 4
    }
  ]
},
"estimatedResultCount": "59600000"
"currentPageIndex": 0,
"moreResultsUrl": "http://www.googl
```

```
    }  
  },  
  ""responseDetails"": null,  
  ""responseStatus"": 200  
}";
```

```
JObject googleSearch = JObject.Parse(googleS
```

```
// get JSON result objects into a list
```

```
IList<JToken> results = googleSearch["respon
```

```
// serialize JSON results into .NET objects
```

```
IList<SearchResult> searchResults = new List
```

```
foreach (JToken result in results)
```

```
{
```

```
    SearchResult searchResult = JsonConvert.De
```

```
    searchResults.Add(searchResult);
```

```
}
```

```
// Title = <b>Paris Hilton</b> - Wikipedia,
```

```
// Content = [1] In 2006, she released her d
```

```
// Url = http://en.wikipedia.org/wiki/Paris_
```

```
// Title = <b>Paris Hilton</b>
```

```
// Content = Self: Zoolander. Socialite <b>P
```

```
// Url = http://www.imdb.com/name/nm0385296/
```



LINQ to JSON is a programming API for working with JSON objects. The API has been designed with LINQ in mind to enable to quick querying and creation of JSON objects. LINQ to JSON sits under the [Newtonsoft.Json.Linq](#) namespace.

## Creating JSON

There are a number of different options when it comes to creating JSON using LINQ to JSON. The first to create objects imperatively. You have total control but it is more verbose than other options.

```
JArray array = new JArray();
JValue text = new JValue("Manual text");
JValue date = new JValue(new DateTime(2000,

array.Add(text);
array.Add(date);

string json = array.ToString();
// [
//   "Manual text",
//   "\/Date(958996800000+1200)\/"
// ]
```

Another option is to create JSON objects declaratively.

```
List<Post> posts = GetPosts();

JObject rss =
    new JObject(
```

```

new JObject(
    new JProperty("channel",
        new JObject(
            new JProperty("title", "James Newton
            new JProperty("link", "http://james.
            new JProperty("description", "James
            new JProperty("item",
                new JSONArray(
                    from p in posts
                    orderby p.Title
                    select new JObject(
                        new JProperty("title", p.Title
                        new JProperty("description", p
                        new JProperty("link", p.Link),
                        new JProperty("category",
                            new JSONArray(
                                from c in p.Categories
                                select new JValue(c)))))))))

```

```

Console.WriteLine(rss.ToString());

```

```

//{
//  "channel": {
//    "title": "James Newton-King",
//    "link": "http://james.newtonking.com",
//    "description": "James Newton-King's bl
//    "item": [
//      {
//        "title": "Json.NET 1.3 + New licen
//        "description": "Annoucing the rele
//        "link": "http://james.newtonking.c
//        "category": [
//          "Json.NET",
//          "CodePlex"

```

```

//      ]
//    },
//    {
//      "title": "LINQ to JSON beta",
//      "description": "Announcing LINQ to
//      "link": "http://james.newtonking.c
//      "category": [
//        "Json.NET",
//        "LINQ"
//      ]
//    }
//  ]
// }
//}

```

You can create a JSON object from a non-JSON type using the [FromObject](#) method.

```

JObject o = JObject.FromObject(new
{
    channel = new
    {
        title = "James Newton-King",
        link = "http://james.newtonking.com",
        description = "James Newton-King's blog.",
        item =
            from p in posts
            orderby p.Title
            select new
            {
                title = p.Title,
                description = p.Description,

```

```
        link = p.Link,
        category = p.Categories
    }
}
});
```

Finally JSON objects can be created from a string use the [Parse](#) method.

```
string json = @"{
    CPU: 'Intel',
    Drives: [
        'DVD read/writer',
        ""500 gigabyte hard drive""
    ]
}";

JObject o = JObject.Parse(json);
```

## Querying JSON

The properties methods that are the most useful when querying JSON objects are the [Children](#) method and the [property index](#).

Children returns all the children of that object. If it is a JObject it will return a collection of properties to work with and if it is a JArray you will get a collection of the array's values.

The property index is used to get a specific child, either by index position for JSON arrays or property name for JSON objects.

```
var postTitles =
    from p in rss["channel"]["item"].Children(
```

```

    select (string)p["title"];

foreach (var item in postTitles)
{
    Console.WriteLine(item);
}

//LINQ to JSON beta
//Json.NET 1.3 + New license + Now on CodePl

var categories =
    from c in rss["channel"]["item"].Children(
    group c by c into g
    orderby g.Count() descending
    select new { Category = g.Key, Count = g.C

foreach (var c in categories)
{
    Console.WriteLine(c.Category + " - Count:

//Json.NET - Count: 2
//LINQ - Count: 1
//CodePlex - Count: 1

```

LINQ to JSON can also be used to manually convert from JSON to a .NET object.

```

public class Shortie
{
    public string Original { get; set; }
    public string Shortened { get; set; }
}

```

```

    public string Short { get; set; }
    public ShortieException Error { get; set; }
}

public class ShortieException
{
    public int Code { get; set; }
    public string ErrorMessage { get; set; }
}

```

Manually serializing and deserializing between .NET objects is most useful when working with JSON that doesn't closely match your .NET objects.

```

string jsonText = @"{
    ""short"":{
        ""original"":""http://www.foo.com/"",
        ""short"":""krehqk"",
        ""error"":{
            ""code"":0,
            ""msg"":""No action taken""}
    }";

JObject json = JObject.Parse(jsonText);

Shortie shortie = new Shortie
{
    Original = (string)json[
    Short = (string)json["sh
    Error = new ShortieExcep
        {
            Code = (int)js

```

```
                ErrorMessage =  
            }  
        };  
  
Console.WriteLine(shortie.Original);  
// http://www.foo.com/  
  
Console.WriteLine(shortie.Error.ErrorMessage  
// No action taken
```



## Querying LINQ to JSON with SelectToken

`SelectToken` provides a method to query LINQ to JSON using a single string path to a desired `JToken`. `SelectToken` makes dynamic queries easy because the entire query is defined in a string.

```
string name = (string)o.SelectToken("Manufac
```

### SelectToken

`SelectToken` is a method on `JToken` and takes a string path to a child token. `SelectToken` returns the child token or a null reference if a token couldn't be found at the path's location.

The path is made up of property names and array indexes separated by periods. Array indexes can use either square or round brackets. Both of the following are valid paths and are equivalent to each other: `Manufacturers[0].Name` and `Manufacturers(0).Name`.

```
JObject o = JObject.Parse(@"{
  ""Stores"": [
    ""Lambton Quay"",
    ""Willis Street""
  ],
  ""Manufacturers"": [
    {
      ""Name"": ""Acme Co"",
      ""Products"": [
        {
          ""Name"": ""Anvil"",
          ""Price"": 50
        }
      ]
    }
  ]
}
```

```

        }
    ],
},
{
    ""Name"": ""Contoso"",
    ""Products"": [
        {
            ""Name"": ""Elbow Grease"",
            ""Price"": 99.95
        },
        {
            ""Name"": ""Headlight Fluid"",
            ""Price"": 4
        }
    ]
}
]
}");

string name = (string)o.SelectToken("Manufac
// Acme Co

decimal productPrice = (decimal)o.SelectToke
// 50

string productName = (string)o.SelectToken("
// Elbow Grease

```

## SelectToken with LINQ

SelectToken can be used in combination with standard LINQ methods.

```
IList<string> storeNames = o.SelectToken("St  
// Lambton Quay  
// Willis Street  
  
IList<string> firstProductNames = o["Manufac  
// null  
// Headlight Fluid  
  
decimal totalPrice = o["Manufacturers"].Sum(  
// 149.95
```



## JSON

To manually read and write JSON Json.NET provides the [JsonReader](#) and [JsonWriter](#) classes.

### JsonTextReader and JsonTextWriter

[JsonTextReader](#) and [JsonTextWriter](#) are used to read and write JSON text. The [JsonTextWriter](#) has a number of settings on it to control how JSON is formatted when it is written. These options include formatting, indentation character, indent count and quote character.

```
StringBuilder sb = new StringBuilder();
StringWriter sw = new StringWriter(sb);

using (JsonWriter jsonWriter = new JsonTextw
{
    jsonWriter.Formatting = Formatting.Indente

    jsonWriter.WriteStartObject();
    jsonWriter.WritePropertyName("CPU");
    jsonWriter.WriteValue("Intel");
    jsonWriter.WritePropertyName("PSU");
    jsonWriter.WriteValue("500W");
    jsonWriter.WritePropertyName("Drives");
    jsonWriter.WriteStartArray();
    jsonWriter.WriteValue("DVD read/writer");
    jsonWriter.WriteComment("(broken)");
    jsonWriter.WriteValue("500 gigabyte hard d
    jsonWriter.WriteValue("200 gigabyte hard d
    jsonWriter.WriteEnd();
    jsonWriter.WriteEndObject());
```

```
}

// {
//   "CPU": "Intel",
//   "PSU": "500W",
//   "Drives": [
//     "DVD read/writer"
//     /*(broken)*/,
//     "500 gigabyte hard drive",
//     "200 gigabyte hard drive"
//   ]
// }
```

## JTokenReader and JTokenWriter

[JTokenReader](#) and [JTokenWriter](#) read and write LINQ to JSON objects. They are located in the [Newtonsoft.Json.Linq](#) namespace. These objects allow you to use LINQ to JSON objects with objects that read and write JSON such as the `JsonSerializer`. For example you can deserialize from a LINQ to JSON object into a regular .NET object and vice versa.

```
JObject o = new JObject(
    new JProperty("Name", "John Smith"),
    new JProperty("BirthDate", new DateTime(19
));

JsonSerializer serializer = new JsonSerializer
Person p = (Person)serializer.Deserialize(ne

Console.WriteLine(p.Name);
// John Smith
```



## Converting between JSON and XML

Json.NET supports converting JSON to XML and vice versa using the [XmlNodeConverter](#).

Elements, attributes, text, comments, character data, processing instructions, namespaces and the XML declaration are all preserved when converting between the two. The only caveat is that it is possible to lose the order of differently named nodes at the same level when they are grouped together into an array.

### Conversion Rules

- Elements remain unchanged.
- Attributes are prefixed with an @ and should be at the start of the object.
- Single child text nodes are a value directly against an element, otherwise they are accessed via #text.
- The XML declaration and processing instructions are prefixed with ?.
- Character data, comments, whitespace and significant whitespace nodes are accessed via #cdata-section, #comment, #whitespace and #significant-whitespace respectively.
- Multiple nodes with the same name at the same level are grouped together into an array.
- Empty elements are null.

### SerializeXmlNode

The `JsonConvert` has two helper methods for converting between JSON and XML. The first is [SerializeXmlNode](#). This method takes an `XmlNode` and serializes it to JSON text.

```
string xml = @"<?xml version=""1.0"" standal
<root>
```

```
<person id=""1"">
<name>Alan</name>
<url>http://www.google.com</url>
</person>
<person id=""2"">
<name>Louis</name>
<url>http://www.yahoo.com</url>
</person>
</root>;
```

```
XmlDocument doc = new XmlDocument();
doc.LoadXml(xml);
```

```
string jsonText = JsonConvert.SerializeXmlNode
//{
//  "?xml": {
//    "@version": "1.0",
//    "@standalone": "no"
//  },
//  "root": {
//    "person": [
//      {
//        "@id": "1",
//        "name": "Alan",
//        "url": "http://www.google.com"
//      },
//      {
//        "@id": "2",
//        "name": "Louis",
//        "url": "http://www.yahoo.com"
//      }
//    ]
//  }
// }
```

```
//}
```

## DeserializeXmlNode

The second helper method on `JsonConvert` is [DeserializeXmlNode](#). This method takes JSON text and deserializes it into a `XmlNode`.

Because valid XML must have one root element the JSON passed to `DeserializeXmlNode` should have one property in the root JSON object. If the root JSON object has multiple properties then the overload that also takes an element name should be used. A root element with that name will be inserted into the deserialized `XmlNode`.

```
string json = @"{
  ""?xml"": {
    ""@version"": ""1.0"",
    ""@standalone"": ""no""
  },
  ""root"": {
    ""person"": [
      {
        ""@id"": ""1"",
        ""name"": ""Alan"",
        ""url"": ""http://www.google.com""
      },
      {
        ""@id"": ""2"",
        ""name"": ""Louis"",
        ""url"": ""http://www.yahoo.com""
      }
    ]
  }
}";
```

```
XmlDocument doc = (XmlDocument)JsonConvert.D
// <?xml version="1.0" standalone="no"?>
// <root>
//   <person id="1">
//     <name>Alan</name>
//     <url>http://www.google.com</url>
//   </person>
//   <person id="2">
//     <name>Louis</name>
//     <url>http://www.yahoo.com</url>
//   </person>
// </root>
```

## Namespace

[Namespaces](#) ► **Newtonsoft.Json**

C#

The **Newtonsoft.Json** namespace provides classes that are used to implement the core services of the framework.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<code>namespace Newtonsoft.Json</code>		

```
Namespace Newtonsoft.Json
```

```
namespace Newtonsoft.Json
```

### ▣ Types

All Types	Classes	Interfaces	Enumerations
-----------	---------	------------	--------------

Icon	Type	Description
	<a href="#"><b>ConstructorHandling</b></a>	Specifies how constructors are used when initializing objects during deserialization by the <a href="#">JsonSerializer</a> .
	<a href="#"><b>DateFormatHandling</b></a>	Specifies how dates are formatted when writing JSON text.
	<a href="#"><b>DateParseHandling</b></a>	Specifies how date formatted strings, e.g. "VDate(1198908717056)V" and

		"2012-03-21T05:40Z", are parse when reading JSON text.
	<a href="#"><b><u>DateTimeZoneHandling</u></b></a>	Specifies how to treat the time value when converting between string and <a href="#">DateTime</a> .
	<a href="#"><b><u>DefaultValueHandling</u></b></a>	Specifies default value handling options for the <a href="#">JsonSerializer</a> .
	<a href="#"><b><u>Formatting</u></b></a>	Specifies formatting options for the <a href="#">JsonTextWriter</a> .
	<a href="#"><b><u>IJsonLineInfo</u></b></a>	Provides an interface to enable a class to return line and position information.
	<a href="#"><b><u>JsonArrayAttribute</u></b></a>	Instructs the <a href="#">JsonSerializer</a> how to serialize the collection.
	<a href="#"><b><u>JsonConstructorAttribute</u></b></a>	Instructs the <a href="#">JsonSerializer</a> to use the specified constructor when deserializing that object.
	<a href="#"><b><u>JsonContainerAttribute</u></b></a>	Instructs the <a href="#">JsonSerializer</a> how to serialize the object.

	<a href="#"><u>JsonConvert</u></a>	Provides methods for converting between common language runtime types and JSON types.
	<a href="#"><u>JsonConverter</u></a>	Converts an object to and from JSON.
	<a href="#"><u>JsonConverterAttribute</u></a>	Instructs the <a href="#"><u>JsonSerializer</u></a> to use the specified <a href="#"><u>JsonConverter</u></a> when serializing the member or class.
	<a href="#"><u>JsonConverterCollection</u></a>	Represents a collection of <a href="#"><u>JsonConverter</u></a> .
	<a href="#"><u>JsonDictionaryAttribute</u></a>	Instructs the <a href="#"><u>JsonSerializer</u></a> how to serialize the collection.
	<a href="#"><u>JsonException</u></a>	The exception thrown when an error occurs during Json serialization or deserialization.
	<a href="#"><u>JsonIgnoreAttribute</u></a>	Instructs the <a href="#"><u>JsonSerializer</u></a> not to serialize the public field or public read/write property value.
	<a href="#"><u>JsonObjectAttribute</u></a>	Instructs the <a href="#"><u>JsonSerializer</u></a> how to serialize the object.

	<a href="#"><b>JsonPropertyAttribute</b></a>	Instructs the <a href="#">JsonSerializer</a> to always serialize the member with the specified name.
	<a href="#"><b>JsonReader</b></a>	Represents a reader that provides fast, non-cached, forward-only access to serialized Json data.
	<a href="#"><b>JsonReaderException</b></a>	The exception thrown when an error occurs while reading Json text.
	<a href="#"><b>JsonSerializationException</b></a>	The exception thrown when an error occurs during Json serialization or deserialization.
	<a href="#"><b>JsonSerializer</b></a>	Serializes and deserializes objects into and from the JSON format. The <a href="#">JsonSerializer</a> enables you to control how objects are encoded into JSON.
	<a href="#"><b>JsonSerializerSettings</b></a>	Specifies the settings on a <a href="#">JsonSerializer</a> object.
	<a href="#"><b>JsonTextReader</b></a>	Represents a reader that

		provides fast, non-cached, forward-only access to JSON text data.
	<a href="#"><u>JsonTextWriter</u></a>	Represents a writer that provide a fast, non-cached, forward-only way of generating Json data.
	<a href="#"><u>JsonToken</u></a>	Specifies the type of Json token
	<a href="#"><u>JsonValidatingReader</u></a>	Represents a reader that provides <a href="#"><u>JsonSchema</u></a> validation
	<a href="#"><u>JsonWriter</u></a>	Represents a writer that provide a fast, non-cached, forward-only way of generating Json data.
	<a href="#"><u>JsonWriterException</u></a>	The exception thrown when an error occurs while reading Json text.
	<a href="#"><u>MemberSerialization</u></a>	Specifies the member serialization options for the <a href="#"><u>JsonSerializer</u></a> .
	<a href="#"><u>MissingMemberHandling</u></a>	Specifies missing member handling options for the

		<a href="#">JsonSerializer</a> .
	<a href="#">NullValueHandling</a>	Specifies null value handling options for the <a href="#">JsonSerializer</a> .
	<a href="#">ObjectCreationHandling</a>	Specifies how object creation is handled by the <a href="#">JsonSerializer</a> .
	<a href="#">PreserveReferencesHandling</a>	Specifies reference handling options for the <a href="#">JsonSerializer</a> .
	<a href="#">ReferenceLoopHandling</a>	Specifies reference loop handling options for the <a href="#">JsonSerializer</a> .
	<a href="#">Required</a>	Indicating whether a property is required.
	<a href="#">JsonReader.State</a>	Specifies the state of the reader.
	<a href="#">TypeNameHandling</a>	Specifies type name handling options for the <a href="#">JsonSerializer</a> .
	<a href="#">WriteState</a>	Specifies the state of the <a href="#">JsonWriter</a> .



## Enumeration

[Namespaces](#) ► [Newtonsoft.Json](#) ► **ConstructorHandling**

Specifies how constructors are used when initializing objects during deserialization by the [JsonSerializer](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum ConstructorHandling
```

```
Public Enumeration ConstructorHandling
```

```
public enum class ConstructorHandling
```

### Members

Member	Description
<b>Default</b>	First attempt to use the public default constructor, then fall back to single paramatized constructor, then the non-public default constructor.
<b>AllowNonPublicDefaultConstructor</b>	Json.NET will use a non-public default constructor before falling back to a paramatized constructor.

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Enumeration

[Namespaces](#) ► [Newtonsoft.Json](#) ► **DateFormatHandling**

Specifies how dates are formatted when writing JSON text.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum DateFormatHandling
```

```
Public Enumeration DateFormatHandling
```

```
public enum class DateFormatHandling
```

### ▣ Members

Member	Description
<b>IsoDateFormat</b>	Dates are written in the ISO 8601 format, e.g. "2012-03-21T05:40Z".
<b>MicrosoftDateFormat</b>	Dates are written in the Microsoft JSON format, e.g. "VDate(1198908717056)V".

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Enumeration

[Namespaces](#) ► [Newtonsoft.Json](#) ► **DateParseHandling**

Specifies how date formatted strings, e.g. "VDate(1198908717056)V" and "2012-03-21T05:40Z", are parsed when reading JSON text.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum DateParseHandling
```

```
Public Enumeration DateParseHandling
```

```
public enum class DateParseHandling
```

### Members

Member	Description
<b>None</b>	Date formatted strings are not parsed to a date type and are read as strings.
<b>DateTime</b>	Date formatted strings, e.g. "VDate(1198908717056)V" and "2012-03-21T05:40Z", are parsed to <b>DateTime</b> .
<b>DateTimeOffset</b>	Date formatted strings, e.g. "VDate(1198908717056)V" and "2012-03-21T05:40Z", are parsed to <b>DateTimeOffset</b> .

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Enumeration

[Namespaces](#) ► [Newtonsoft.Json](#) ► **DateTimeZoneHandling**

C#

Specifies how to treat the time value when converting between string and [DateTime](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum DateTimeZoneHandling
```

```
Public Enumeration DateTimeZoneHandling
```

```
public enum class DateTimeZoneHandling
```

### Members

Member	Description
<b>Local</b>	Treat as local time. If the <a href="#">DateTime</a> object represents a Coordinated Universal Time (UTC), it is converted to the local time.
<b>Utc</b>	Treat as a UTC. If the <a href="#">DateTime</a> object represents a local time, it is converted to UTC.
<b>Unspecified</b>	Treat as a local time if a <a href="#">DateTime</a> is being converted to a string. If a string is being converted to <a href="#">DateTime</a> , convert to a local time if a time zone is specified.
<b>RoundtripKind</b>	Time zone information should be preserved when converting.

(4.5.6.14930)

## Enumeration

[Namespaces](#) ► [Newtonsoft.Json](#) ► **DefaultValueHandling**

Specifies default value handling options for the [JsonSerializer](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
[FlagsAttribute]
public enum DefaultValueHandling
```

```
<FlagsAttribute> _
Public Enumeration DefaultValueHandling
```

```
[FlagsAttribute]
public enum class DefaultValueHandling
```

### Members

Member	Description
<b>Include</b>	Include members where the member value is the same as the member's default value when serializing objects. Included members are written to JSON. Has no effect when deserializing.
<b>Ignore</b>	Ignore members where the member value is the same as the member's default value when serializing objects so that is is not written to JSON, and ignores setting members when the JSON value equals the member's default value.
<b>Populate</b>	Members with a default value but no JSON will be set to their default value

	when deserializing.
<b>IgnoreAndPopulate</b>	Ignore members where the member value is the same as the member's default value when serializing objects and sets members to their default value when deserializing.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Specifies formatting options for the [JsonTextWriter](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum Formatting
```

```
Public Enumeration Formatting
```

```
public enum class Formatting
```

### ▣ Members

Member	Description
<b>None</b>	No special formatting is applied. This is the default.
<b>Indented</b>	Causes child objects to be indented according to the <a href="#">Indentation</a> and <a href="#">IndentChar</a> settings.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Provides an interface to enable a class to return line and position information.

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
----	--------------	------------

```
public interface IJsonLineInfo
```

```
Public Interface IJsonLineInfo
```

```
public interface class IJsonLineInfo
```

▣ **Members**

All Members	Methods	Properties	
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">HasLineInfo()</a>	Gets a value indicating whether the class can return line information.
	<a href="#">LineNumber</a>	Gets the current line number.
	<a href="#">LinePosition</a>	Gets the current line position.

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [IJsonLineInfo](#) ► **HasLineInfo()**

Gets a value indicating whether the class can return line information.

C# ▼

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
bool HasLineInfo()
```

```
Function HasLineInfo As Boolean
```

```
bool HasLineInfo()
```

## ▣ Return Value

`true` if `LineNumber` and `LinePosition` can be provided; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets the current line number.

## Declaration Syntax

C#	Visual Basic	Visual C++
<pre><code>int LineNumber { get; }</code></pre>		
<pre><code>ReadOnly Property LineNumber As Integer Get</code></pre>		
<pre><code>property int LineNumber {     int get (); }</code></pre>		

## Value

The current line number or 0 if no line information is available (for example, HasLineInfo returns false).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets the current line position.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre><code>int LinePosition { get; }</code></pre>		
<pre><code>ReadOnly Property LinePosition As Integer Get</code></pre>		
<pre><code>property int LinePosition {     int get (); }</code></pre>		

## ▣ Value

The current line position or 0 if no line information is available (for example, HasLineInfo returns false).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Instructs the [JsonSerializer](#) how to serialize the collection.

### [-] Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public sealed class JsonArrayAttribute : JsonContainerAttribute
```

```
Public NotInheritable Class JsonArrayAttribute _
    Inherits JsonContainerAttribute
```

```
public ref class JsonArrayAttribute sealed : public
```

### [-] Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonArrayAttribute()</a>	Initializes a new instance of the <b>JsonArrayAttribute</b> class.
	<a href="#">JsonArrayAttribute(Boolean)</a>	Initializes a new instance of the <a href="#">JsonObjectAttribute</a> class with a flag indicating whether the array can contain null items
	<a href="#">JsonArrayAttribute(String)</a>	Initializes a new instance of the <b>JsonArrayAttribute</b> class with the

		specified container Id.
	<a href="#">AllowNullItems</a>	Gets or sets a value indicating whether null items are allowed in the collection.
	<a href="#">Description</a>	Gets or sets the description.  (Inherited from <a href="#">JsonContainerAttribute</a> .)
	<a href="#">Equals(Object)</a>	Returns a value that indicates whether this instance is equal to a specified object.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Returns the hash code for this instance.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">Id</a>	Gets or sets the id.  (Inherited from

		<a href="#">JsonContainerAttribute.</a> )
	<a href="#">IsDefaultAttribute()</a>	<p>When overridden in a derived class indicates whether the value of this instance is the default value for the derived class.</p> <p>(Inherited from <a href="#">Attribute.</a>)</p>
	<a href="#">IsReference</a>	<p>Gets or sets a value that indicates whether to preserve object references.</p> <p>(Inherited from <a href="#">JsonContainerAttribute.</a>)</p>
	<a href="#">ItemConverterType</a>	<p>Gets the collection's items converter.</p> <p>(Inherited from <a href="#">JsonContainerAttribute.</a>)</p>
	<a href="#">ItemsIsReference</a>	<p>Gets or sets a value that indicates whether to preserve collection's items references.</p> <p>(Inherited from <a href="#">JsonContainerAttribute.</a>)</p>
	<a href="#">ItemReferenceLoopHandling</a>	<p>Gets or sets the reference loop handling used when serializing the collection's items.</p> <p>(Inherited from <a href="#">JsonContainerAttribute.</a>)</p>
	<a href="#">ItemTypePropertyNameHandling</a>	<p>Gets or sets the type name handling used when serializing the collection's items.</p> <p>(Inherited from</p>

		<a href="#">JsonContainerAttribute.</a> )
	<a href="#">Match(Object)</a>	When overridden in a derived class returns a value that indicates whether this instance equals a specified object.  (Inherited from <a href="#">Attribute.</a> )
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">Title</a>	Gets or sets the title.  (Inherited from <a href="#">JsonContainerAttribute.</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">TypeId</a>	When implemented in a derived class, gets a unique identifier for this <a href="#">Attribute</a> .  (Inherited from <a href="#">Attribute.</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [Attribute](#)

└─ [JsonContainerAttribute](#)

└─ **JsonArrayAttribute**



## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonArrayAttribute](#) ►

### JsonArrayAttribute()

#### Members

Icon	Member	Description
	<a href="#">JsonArrayAttribute()</a>	Initializes a new instance of the <a href="#">JsonArrayAttribute</a> class.
	<a href="#">JsonArrayAttribute(Boolean)</a>	Initializes a new instance of the <a href="#">JsonObjectAttribute</a> class with a flag indicating whether the array can contain null items
	<a href="#">JsonArrayAttribute(String)</a>	Initializes a new instance of the <a href="#">JsonArrayAttribute</a> class with the specified container Id.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonArrayAttribute](#) ►

### JsonArrayAttribute()

Initializes a new instance of the [JsonArrayAttribute](#) class.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonArrayAttribute()</pre>		
	<pre>Public Sub New</pre>	
		<pre>public: JsonArrayAttribute()</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonArrayAttribute Constructor (allowNullItems)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonArrayAttribute](#) ►

## JsonArrayAttribute(Boolean)

Initializes a new instance of the [JsonObjectAttribute](#) class with a flag indicating whether the array can contain null items

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonArrayAttribute(  
    bool allowNullItems  
)
```

```
Public Sub New ( _  
    allowNullItems As Boolean _  
)
```

```
public:  
JsonArrayAttribute(  
    bool allowNullItems  
)
```

### Parameters

#### ***allowNullItems*** ([Boolean](#))

A flag indicating whether the array can contain null items.

## Constructor (id)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonArrayAttribute](#) ►

### JsonArrayAttribute(String)

Initializes a new instance of the [JsonArrayAttribute](#) class with the specified container Id.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonArrayAttribute(     string id )</pre>		
	<pre>Public Sub New ( _     id As String _ )</pre>	
		<pre>public: JsonArrayAttribute(     String^ id )</pre>

#### ▣ Parameters

***id*** ([String](#))

The container Id.

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonArrayAttribute](#) ►

## AllowNullItems

Gets or sets a value indicating whether null items are allowed in the collection.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool AllowNullItems { get; set; }
```

```
Public Property AllowNullItems As Boolean  
    Get  
    Set
```

```
public:  
property bool AllowNullItems {  
    bool get ();  
    void set (bool value);  
}
```

### Value

`true` if null items are allowed in the collection; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonConstructorAttribute**

C# ▼

Instructs the [JsonSerializer](#) to use the specified constructor when deserializing that object.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public sealed class JsonConstructorAttribute : Attribute
```

```
Public NotInheritable Class JsonConstructorAttribute
    Inherits Attribute
```

```
public ref class JsonConstructorAttribute sealed : p
```

### Members

All Members

Constructors

Methods

Properties

Public  
 Protected

Instance  
 Static

Declared  
 Inherited

Icon	Member	Description
	<a href="#">JsonConstructorAttribute()</a>	Initializes a new instance of the <b>JsonConstructorAttribute</b> class
	<a href="#">Equals(Object)</a>	Returns a value that indicates whether this instance is equal to a specified object.  (Inherited from <a href="#">Attribute</a> .)

	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Returns the hash code for this instance.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">IsDefaultAttribute()</a>	When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">Match(Object)</a>	When overridden in a derived class, returns a value that indicates whether this instance equals a specified object.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">TypeId</a>	When implemented in a derived

class, gets a unique identifier for this [Attribute](#).

(Inherited from [Attribute](#).)

## ▣ Inheritance Hierarchy

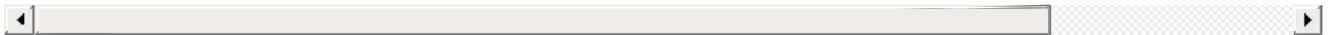
[Object](#)

└─ [Attribute](#)

└─ **JsonConstructorAttribute**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



# JsonConstructorAttribute Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConstructorAttribute](#) ►  
**JsonConstructorAttribute()**

Initializes a new instance of the [JsonConstructorAttribute](#) class

C#

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonConstructorAttribute()
```

```
Public Sub New
```

```
public:  
JsonConstructorAttribute()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonContainerAttribute**

Instructs the [JsonSerializer](#) how to serialize the object.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract class JsonContainerAttribute : Attribute
```

```
Public MustInherit Class JsonContainerAttribute _  
    Inherits Attribute
```

```
public ref class JsonContainerAttribute abstract : p
```

### Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declared <input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonContainerAttribute()</a>	Initializes a new instance of the <b>JsonContainerAttribute</b> class.
	<a href="#">JsonContainerAttribute(String)</a>	Initializes a new instance of the <b>JsonContainerAttribute</b> class with the specified container Id.
	<a href="#">Description</a>	Gets or sets the description.

	<a href="#">Equals(Object)</a>	Returns a value that indicates whether this instance is equal to a specified object.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Returns the hash code for this instance.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">Id</a>	Gets or sets the id.
	<a href="#">IsDefaultAttribute()</a>	When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">IsReference</a>	Gets or sets a value that indicates whether to preserve object references.

	<a href="#">ItemConverterType</a>	Gets the collection's items converter.
	<a href="#">ItemsReference</a>	Gets or sets a value that indicates whether to preserve collection's items references.
	<a href="#">ItemReferenceLoopHandling</a>	Gets or sets the reference loop handling used when serializing the collection's items.
	<a href="#">ItemTypeNameHandling</a>	Gets or sets the type name handling used when serializing the collection's items.
	<a href="#">Match(Object)</a>	When overridden in a derived class, returns a value that indicates whether this instance equals a specified object.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Title</a>	Gets or sets the title.
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .

		(Inherited from <a href="#">Object</a> .)
	<a href="#">TypeId</a>	When implemented in a derived class, gets a unique identifier for this <a href="#">Attribute</a> .  (Inherited from <a href="#">Attribute</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

└─ [Attribute](#)

└─ **JsonContainerAttribute**

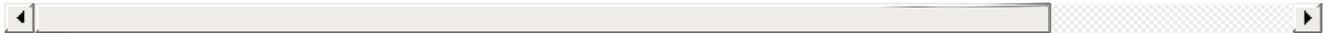
└─ [JsonArrayAttribute](#)

└─ [JsonDictionaryAttribute](#)

└─ [JsonObjectAttribute](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonContainerAttribute](#) ►

### JsonContainerAttribute()

#### Members

Icon	Member	Description
	<a href="#">JsonContainerAttribute()</a>	Initializes a new instance of the <a href="#">JsonContainerAttribute</a> class.
	<a href="#">JsonContainerAttribute(String)</a>	Initializes a new instance of the <a href="#">JsonContainerAttribute</a> class with the specified container Id.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**Constructor**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonContainerAttribute](#) ►**JsonContainerAttribute()**

Initializes a new instance of the [JsonContainerAttribute](#) class.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
protected JsonContainerAttribute()
```

```
Protected Sub New
```

```
protected:  
JsonContainerAttribute()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Constructor (id)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonContainerAttribute](#) ►

### JsonContainerAttribute(String)

Initializes a new instance of the [JsonContainerAttribute](#) class with the specified container Id.

C#

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected JsonContainerAttribute(  
    string id  
)
```

```
Protected Sub New ( _  
    id As String _  
)
```

```
protected:  
JsonContainerAttribute(  
    String^ id  
)
```

#### Parameters

***id*** ([String](#))

The container Id.

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonContainerAttribute](#) ►

## Description

Gets or sets the description.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Description { get; set; }
```

```
Public Property Description As String  
    Get  
    Set
```

```
public:  
property String^ Description {  
    String^ get ();  
    void set (String^ value);  
}
```

### ▣ Value

The description.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets the id.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public string Id { get; set; }
```

```
Public Property Id As String
    Get
    Set
```

```
public:
property String^ Id {
    String^ get ();
    void set (String^ value);
}
```

▣ **Value**

The id.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonContainerAttribute](#) ►

## IsReference

C#

Gets or sets a value that indicates whether to preserve object reference

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool IsReference { get; set; }
```

```
Public Property IsReference As Boolean  
    Get  
    Set
```

```
public:  
property bool IsReference {  
    bool get ();  
    void set (bool value);  
}
```

### ▣ Value

`true` to keep object reference; otherwise, `false`. The default is `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonContainerAttribute](#) ►

### ItemConverterType

Gets the collection's items converter.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Type ItemConverterType { get; set; }
```

```
Public Property ItemConverterType As Type
    Get
    Set
```

```
public:
property Type^ ItemConverterType {
    Type^ get ();
    void set (Type^ value);
}
```

#### ▣ Value

The collection's items converter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonContainerAttribute](#) ►

## ItemIsReference

Gets or sets a value that indicates whether to preserve collection's item references.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool ItemIsReference { get; set; }
```

```
Public Property ItemIsReference As Boolean  
    Get  
    Set
```

```
public:  
property bool ItemIsReference {  
    bool get ();  
    void set (bool value);  
}
```

### Value

**true** to keep collection's items object references; otherwise, **false**.  
The default is **false**.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Documentation **ItemReferenceLoopHandling Property**

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonContainerAttribute](#) ►

**ItemReferenceLoopHandling**

Gets or sets the reference loop handling used when serializing the collection's items.

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
<code>public <a href="#">ReferenceLoopHandling</a> ItemReferenceLoopHandli</code>		

```
Public Property ItemReferenceLoopHandling As Referen
    Get
    Set
```

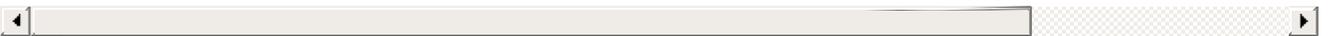
```
public:
property ReferenceLoopHandling ItemReferenceLoopHand
    ReferenceLoopHandling get ();
    void set (ReferenceLoopHandling value);
}
```

▣ **Value**

The reference loop handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonContainerAttribute](#) ►

### ItemTypeNameHandling

Gets or sets the type name handling used when serializing the collection's items.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public TypeNameHandling ItemTypeNameHandling { get;
```

```
Public Property ItemTypeNameHandling As TypeNameHand  
    Get  
    Set
```

```
public:  
property TypeNameHandling ItemTypeNameHandling {  
    TypeNameHandling get ();  
    void set (TypeNameHandling value);  
}
```

#### Value

The type name handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



Gets or sets the title.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public string Title { get; set; }</pre>		
	<pre>Public Property Title As String     Get     Set</pre>	
		<pre>public: property String^ Title {     String^ get ();     void set (String^ value); }</pre>

## ▣ Value

The title.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Provides methods for converting between common language runtime types and JSON types.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public static class JsonConvert
```

```
Public NotInheritable Class JsonConvert
```

```
public ref class JsonConvert abstract sealed
```

## ▣ Members

All Members	Methods	Fields	
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherite

Icon	Member	Description
	<a href="#">DeserializeAnonymousType&lt;T&gt;(String, T)</a>	Deserializes the JSON to the given anonymous type.
	<a href="#">DeserializeObject(String)</a>	Deserializes the JSON to a .NET object.
	<a href="#">DeserializeObject(String, JsonSerializerSettings)</a>	Deserializes the JSON to a .NET object.

	<a href="#">DeserializeObject(String, Type)</a>	Deserializes the JSON to the specified .NET type.
	<a href="#">DeserializeObject&lt;T&gt;(String)</a>	Deserializes the JSON to the specified .NET type.
	<a href="#">DeserializeObject&lt;T&gt;(String, JsonSerializer[])</a>	Deserializes the JSON to the specified .NET type.
	<a href="#">DeserializeObject&lt;T&gt;(String, JsonSerializerSettings)</a>	Deserializes the JSON to the specified .NET type.
	<a href="#">DeserializeObject(String, Type, JsonSerializer[])</a>	Deserializes the JSON to the specified .NET type.
	<a href="#">DeserializeObject(String, Type, JsonSerializerSettings)</a>	Deserializes the JSON to the specified .NET type.
	<a href="#">DeserializeObjectAsync&lt;T&gt;(Void)</a>	
	<a href="#">DeserializeObjectAsync&lt;T&gt;(Void, T)</a>	
	<a href="#">DeserializeObjectAsync(Void)</a>	
	<a href="#">DeserializeObjectAsync(Void, Object, String)</a>	
	<a href="#">DeserializeXmlNode(String)</a>	Deserializes the XmlNode from a JSON string.

	<a href="#">DeserializeXmlNode(String, String)</a>	Deserializes the XmlNode from a JSON string nested in a root element.
	<a href="#">DeserializeXmlNode(String, String, Boolean)</a>	Deserializes the XmlNode from a JSON string nested in a root element.
	<a href="#">DeserializeXmlNode(String)</a>	Deserializes the <a href="#">XmlNode</a> from a JSON string.
	<a href="#">DeserializeXmlNode(String, String)</a>	Deserializes the <a href="#">XmlNode</a> from a JSON string nested in a root element.
	<a href="#">DeserializeXmlNode(String, String, Boolean)</a>	Deserializes the <a href="#">XmlNode</a> from a JSON string nested in a root element.
	<a href="#">False</a>	Represents JavaScript's boolean value false as a string. This field is read-only.
		

	<a href="#">NaN</a>	Represents JavaScript's NaN as a string. This field is read-only.
 	<a href="#">NegativeInfinity</a>	Represents JavaScript's negative infinity as a string. This field is read-only.
 	<a href="#">Null</a>	Represents JavaScript's null as a string. This field is read-only.
 	<a href="#">PopulateObject(String, Object)</a>	Populates the object with values from the JSON string.
 	<a href="#">PopulateObject(String, Object, JsonSerializerSettings)</a>	Populates the object with values from the JSON string.
 	<a href="#">PopulateObjectAsync(String, Object, JsonSerializerSettings)</a>	Asynchronously populates the object with values from the JSON string.
 	<a href="#">PositiveInfinity</a>	Represents JavaScript's positive infinity as a string. This field is read-

		only.
	<a href="#">SerializeObject(Object)</a>	Serializes the specified object to a JSON string.
	<a href="#">SerializeObject(Object, Formatting)</a>	Serializes the specified object to a JSON string.
	<a href="#">SerializeObject(Object, JsonSerializer[])</a>	Serializes the specified object to a JSON string using a collection of <a href="#">JsonConverter</a> .
	<a href="#">SerializeObject(Object, Formatting, JsonSerializer[])</a>	Serializes the specified object to a JSON string using a collection of <a href="#">JsonConverter</a> .
	<a href="#">SerializeObject(Object, JsonSerializerSettings)</a>	Serializes the specified object to a JSON string using a collection of <a href="#">JsonConverter</a> .
	<a href="#">SerializeObject(Object, Formatting, JsonSerializerSettings)</a>	Serializes the specified object to a JSON string using a collection of <a href="#">JsonConverter</a> .

	<a href="#">SerializeObjectAsync(Void)</a>	
	<a href="#">SerializeObjectAsync(Void, String)</a>	
	<a href="#">SerializeObjectAsync(Void, String, Object)</a>	
	<a href="#">SerializeXmlNode(XmlNode)</a>	Serializes the XML node to a JSON string.
	<a href="#">SerializeXmlNode(XmlNode, Formatting)</a>	Serializes the XML node to a JSON string.
	<a href="#">SerializeXmlNode(XmlNode, Formatting, Boolean)</a>	Serializes the XML node to a JSON string.
	<a href="#">SerializeXmlNode(XObject)</a>	Serializes the <a href="#">XmlNode</a> to JSON string.
	<a href="#">SerializeXmlNode(XObject, Formatting)</a>	Serializes the <a href="#">XmlNode</a> to JSON string.
	<a href="#">SerializeXmlNode(XObject, Formatting, Boolean)</a>	Serializes the <a href="#">XmlNode</a> to JSON string.
	<a href="#">ToString(DateTime)</a>	Converts the <a href="#">DateTime</a> to its JSON string representation.

 <a href="#">ToString(DateTime, DateFormatHandling, DateTimeZoneHandling)</a>	Converts the <a href="#">DateTime</a> to its JSON string representation using the <a href="#">DateFormatHandling</a> specified.
 <a href="#">ToString(DateTimeOffset)</a>	Converts the <a href="#">DateTimeOffset</a> to its JSON string representation.
 <a href="#">ToString(DateTimeOffset, DateFormatHandling)</a>	Converts the <a href="#">DateTimeOffset</a> to its JSON string representation using the <a href="#">DateFormatHandling</a> specified.
 <a href="#">ToString(Boolean)</a>	Converts the <a href="#">Boolean</a> to its JSON string representation.
 <a href="#">ToString(Char)</a>	Converts the <a href="#">Char</a> to its JSON string representation.
 <a href="#">ToString(Enum)</a>	Converts the <a href="#">Enum</a> to its JSON string representation.

 <a href="#">ToString(Int32)</a>	Converts the <a href="#">Int32</a> to its JSON string representation.
 <a href="#">ToString(Int16)</a>	Converts the <a href="#">Int16</a> to its JSON string representation.
 <a href="#">ToString(UInt16)</a>	Converts the <a href="#">UInt16</a> to its JSON string representation.
 <a href="#">ToString(UInt32)</a>	Converts the <a href="#">UInt32</a> to its JSON string representation.
 <a href="#">ToString(Int64)</a>	Converts the <a href="#">Int64</a> to its JSON string representation.
 <a href="#">ToString(UInt64)</a>	Converts the <a href="#">UInt64</a> to its JSON string representation.
 <a href="#">ToString(Single)</a>	Converts the <a href="#">Single</a> to its JSON string representation.

 <a href="#">ToString(Double)</a>	Converts the <a href="#">Double</a> to its JSON string representation.
 <a href="#">ToString(Byte)</a>	Converts the <a href="#">Byte</a> to its JSON string representation.
 <a href="#">ToString(SByte)</a>	Converts the <a href="#">SByte</a> to its JSON string representation.
 <a href="#">ToString(Decimal)</a>	Converts the <a href="#">Decimal</a> to its JSON string representation.
 <a href="#">ToString(Guid)</a>	Converts the <a href="#">Guid</a> to its JSON string representation.
 <a href="#">ToString(TimeSpan)</a>	Converts the <a href="#">TimeSpan</a> to its JSON string representation.
 <a href="#">ToString(Uri)</a>	Converts the <a href="#">Uri</a> to its JSON string representation.

 <a href="#">ToString(String)</a>	Converts the <a href="#">String</a> to its JSON string representation.
 <a href="#">ToString(String, Char)</a>	Converts the <a href="#">String</a> to its JSON string representation.
 <a href="#">ToString(Object)</a>	Converts the <a href="#">Object</a> to its JSON string representation.
 <a href="#">True</a>	Represents JavaScript's boolean value true as a string. This field is read-only.
 <a href="#">Undefined</a>	Represents JavaScript's undefined as a string. This field is read-only.

## ▣ Inheritance Hierarchy

[Object](#)

└─ **JsonConvert**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Documentation **DeserializeAnonymousType<T> Method (value, anonymousTypeObject)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►**DeserializeAnonymousType<T>(String, T)**

Deserializes the JSON to the given anonymous type.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public static T DeserializeAnonymousType<T>(
    string value,
    T anonymousTypeObject
)
```

```
Public Shared Function DeserializeAnonymousType(Of T
    value As String, _
    anonymousTypeObject As T _
) As T
```

```
public:
generic<typename T>
static T DeserializeAnonymousType(
    String^ value,
    T anonymousTypeObject
)
```

**Generic Template Parameters****T**

The anonymous type to deserialize to. This can't be specified traditionally and must be inferred from the anonymous type passed as a parameter.

**Parameters**

**value** ([String](#))

The JSON to deserialize.

**anonymousTypeObject** (**T**)

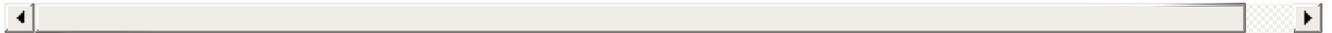
The anonymous type object.

#### ▣ **Return Value**

The deserialized anonymous type from the JSON string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## DeserializeObject()

### Members

Icon	Member	Description
	<a href="#">DeserializeObject(String)</a>	Deserializes the JSON to a .NET object.
	<a href="#">DeserializeObject(String, JsonSerializerSettings)</a>	Deserializes the JSON to a .NET object.
	<a href="#">DeserializeObject(String, Type)</a>	Deserializes the JSON to the specified .NET type.
	<a href="#">DeserializeObject&lt;T&gt;(String)</a>	Deserializes the JSON to the specified .NET type.
	<a href="#">DeserializeObject&lt;T&gt;(String, JsonSerializer[])</a>	Deserializes the JSON to the specified .NET type.
	<a href="#">DeserializeObject&lt;T&gt;(String, JsonSerializerSettings)</a>	Deserializes the JSON to the specified .NET type.
	<a href="#">DeserializeObject(String, Type, JsonSerializer[])</a>	Deserializes the JSON to the specified .NET type.



[DeserializeObject\(String, Type, JsonSerializerSettings\)](#)

Deserializes the JSON to the specified .NET type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**Method (value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►**DeserializeObject<T>(String)**

Deserializes the JSON to the specified .NET type.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public static T DeserializeObject<T>(
    string value
)
```

```
Public Shared Function DeserializeObject(Of T) ( _
    value As String _
) As T
```

```
public:
generic<typename T>
static T DeserializeObject(
    String^ value
)
```

**Generic Template Parameters*****T***

The type of the object to deserialize to.

**Parameters*****value*** ([String](#))

The JSON to deserialize.

**Return Value**

The deserialized object from the Json string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

# Json.NET - Quick Starts & API Documentation **DeserializeObject Method (value)**

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## DeserializeObject(String)

C#

Deserializes the JSON to a .NET object.

### ▣ Declaration Syntax

```
C#           Visual Basic           Visual C++
public static Object DeserializeObject(
    string value
)
```

```
Public Shared Function DeserializeObject ( _
    value As String _
) As Object
```

```
public:
static Object^ DeserializeObject(
    String^ value
)
```

### ▣ Parameters

#### **value (String)**

The JSON to deserialize.

### ▣ Return Value

The deserialized object from the Json string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# DeserializeObject<T> Method (value, converters)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## DeserializeObject<T>(String, JsonConvert[])

Deserializes the JSON to the specified .NET type.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public static T DeserializeObject<T>(
    string value,
    params JsonConvert[] converters
)
```

```
Public Shared Function DeserializeObject(Of T) ( _
    value As String, _
    ParamArray converters As JsonConvert() _
) As T
```

```
public:
generic<typename T>
static T DeserializeObject(
    String^ value,
    ... array<JsonConverter^>^ converters
)
```

### ▣ Generic Template Parameters

**T**

The type of the object to deserialize to.

### ▣ Parameters

**value** ([String](#))

The JSON to deserialize.

***converters*** ([JsonConverter\[\]](#))

Converters to use while deserializing.

▣ **Return Value**

The deserialized object from the JSON string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# DeserializeObject Method (value, settings)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## DeserializeObject(String, JsonSerializerSettings)

Deserializes the JSON to a .NET object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static Object DeserializeObject(  
    string value,  
    JsonSerializerSettings settings  
)
```

```
Public Shared Function DeserializeObject ( _  
    value As String, _  
    settings As JsonSerializerSettings _  
) As Object
```

```
public:  
static Object^ DeserializeObject(  
    String^ value,  
    JsonSerializerSettings^ settings  
)
```

### ▣ Parameters

#### **value** ([String](#))

The JSON to deserialize.

#### **settings** ([JsonSerializerSettings](#))

The [JsonSerializerSettings](#) used to deserialize the object. If this is null, default serialization settings will be is used.

### ▣ Return Value

The deserialized object from the JSON string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

# DeserializeObject<T> Method (value, settings)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## DeserializeObject<T>(String, JsonSerializerSettings)

Deserializes the JSON to the specified .NET type.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static T DeserializeObject<T>(
    string value,
    JsonSerializerSettings settings
)
```

```
Public Shared Function DeserializeObject(Of T) ( _
    value As String, _
    settings As JsonSerializerSettings _
) As T
```

```
public:
generic<typename T>
static T DeserializeObject(
    String^ value,
    JsonSerializerSettings^ settings
)
```

### ▣ Generic Template Parameters

**T**

The type of the object to deserialize to.

### ▣ Parameters

**value** ([String](#))

The object to deserialize.

***settings* ([JsonSerializerSettings](#))**

The [JsonSerializerSettings](#) used to deserialize the object. If this is null, default serialization settings will be is used.

**▣ Return Value**

The deserialized object from the JSON string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# Json.NET - Quick Starts & API Documentation **DeserializeObject Method** **(value, type)**

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## DeserializeObject(String, Type)

C#

Deserializes the JSON to the specified .NET type.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static Object DeserializeObject(  
    string value,  
    Type type  
)
```

```
Public Shared Function DeserializeObject ( _  
    value As String, _  
    type As Type _  
) As Object
```

```
public:  
static Object^ DeserializeObject(  
    String^ value,  
    Type^ type  
)
```

### ▣ Parameters

#### **value** ([String](#))

The JSON to deserialize.

#### **type** ([Type](#))

The [Type](#) of object being deserialized.

### ▣ Return Value

The deserialized object from the Json string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## DeserializeObject Method (value, type, converters)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

**DeserializeObject(String, Type, JsonSerializer[])**

C#

Deserializes the JSON to the specified .NET type.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static Object DeserializeObject(  
    string value,  
    Type type,  
    params JsonConverter[] converters  
)
```

```
Public Shared Function DeserializeObject ( _  
    value As String, _  
    type As Type, _  
    ParamArray converters As JsonConverter() _  
) As Object
```

```
public:  
static Object^ DeserializeObject(  
    String^ value,  
    Type^ type,  
    ... array<JsonConverter^>^ converters  
)
```

### ▣ Parameters

#### **value** ([String](#))

The JSON to deserialize.

#### **type** ([Type](#))

The type of the object to deserialize.

***converters*** ([JsonConverter\[\]](#))

Converters to use while deserializing.

▣ **Return Value**

The deserialized object from the JSON string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## DeserializeObject Method (value, type, settings)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### DeserializeObject(String, Type, JsonSerializerSettings)

Deserializes the JSON to the specified .NET type.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static Object DeserializeObject(  
    string value,  
    Type type,  
    JsonSerializerSettings settings  
)
```

```
Public Shared Function DeserializeObject ( _  
    value As String, _  
    type As Type, _  
    settings As JsonSerializerSettings _  
) As Object
```

```
public:  
static Object^ DeserializeObject(  
    String^ value,  
    Type^ type,  
    JsonSerializerSettings^ settings  
)
```

#### ▣ Parameters

##### **value** ([String](#))

The JSON to deserialize.

##### **type** ([Type](#))

The type of the object to deserialize to.

***settings* ([JsonSerializerSettings](#))**

The [JsonSerializerSettings](#) used to deserialize the object. If this is null, default serialization settings will be is used.

**▣ Return Value**

The deserialized object from the JSON string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### DeserializeObjectAsync()

#### Members

Icon	Member	Description
	<a href="#">DeserializeObjectAsync&lt;T&gt;(Void)</a>	
	<a href="#">DeserializeObjectAsync&lt;T&gt;(Void, T)</a>	
	<a href="#">DeserializeObjectAsync(Void)</a>	
	<a href="#">DeserializeObjectAsync(Void, Object, String)</a>	

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# DeserializeObjectAsync Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## DeserializeObjectAsync(Void)

[Missing <summary> documentation for "M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync(System

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static Task DeserializeObjectAsync(  
    void value  
)
```

```
Public Shared Function DeserializeObjectAsync ( _  
    value As Void _  
) As Task
```

```
public:  
static Task^ DeserializeObjectAsync(  
    void value  
)
```

### ▣ Parameters

*value* ([Void](#))

[Missing <param name="value"/> documentation for "M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync(Sys

### ▣ Return Value

[Missing <returns> documentation for "M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync(System

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)



Documentation **DeserializeObjectAsync<T> Method (value)**

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

**DeserializeObjectAsync<T>(Void)**

[Missing <summary> documentation for "M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync`1(System)"]

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
<pre>public static Task DeserializeObjectAsync&lt;T&gt;(     void value )</pre>		

```
Public Shared Function DeserializeObjectAsync(Of T)
    value As Void _
) As Task
```

```
public:
generic<typename T>
static Task^ DeserializeObjectAsync(
    void value
)
```

▣ **Generic Template Parameters**

*T*

[Missing <typeparam name="T"/> documentation for "M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync`1(System)"]

▣ **Parameters**

*value* ([Void](#))

[Missing <param name="value"/> documentation for  
"M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync`1(\$

## Return Value

[Missing <returns> documentation for  
"M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync`1(Syste

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



# DeserializeObjectAsync Method (value, type, settings)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

DeserializeObjectAsync(Void, Object, String)

C#

[Missing <summary> documentation for  
"M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync(System

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static Task DeserializeObjectAsync(  
    void value,  
    Object type,  
    string settings  
)
```

```
Public Shared Function DeserializeObjectAsync ( _  
    value As Void, _  
    type As Object, _  
    settings As String _  
) As Task
```

```
public:  
static Task^ DeserializeObjectAsync(  
    void value,  
    Object^ type,  
    String^ settings  
)
```

## ▣ Parameters

**value** ([Void](#))

[Missing <param name="value"/> documentation for  
"M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync(Sys

*type* ([Object](#))

[Missing <param name="type"/> documentation for  
"M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync(Sys

*settings* ([String](#))

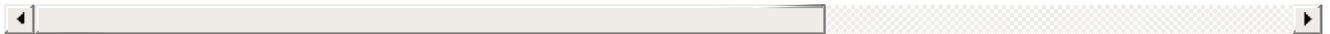
[Missing <param name="settings"/> documentation for  
"M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync(Sys

#### ▣ **Return Value**

[Missing <returns> documentation for  
"M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync(System

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



Documentation **DeserializeObjectAsync<T> Method (value, settings)**

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

**DeserializeObjectAsync<T>(Void, T)**

**[Missing <summary> documentation for "M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync`1(System**

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
<pre>public static Task DeserializeObjectAsync&lt;T&gt;(     void value,     T settings )</pre>		

```
Public Shared Function DeserializeObjectAsync(Of T)
    value As Void, _
    settings As T _
) As Task
```

```
public:
generic<typename T>
static Task^ DeserializeObjectAsync(
    void value,
    T settings
)
```

▣ **Generic Template Parameters**

**T**

**[Missing <typeparam name="T"/> documentation for "M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync`1(\$**

## Parameters

*value* ([Void](#))

[Missing <param name="value"/> documentation for  
"M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync`1(\$

*settings* (T)

[Missing <param name="settings"/> documentation for  
"M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync`1(\$

## Return Value

[Missing <returns> documentation for  
"M:Newtonsoft.Json.JsonConvert.DeserializeObjectAsync`1(Syste

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### DeserializeXmlNode()

#### Members

Icon	Member	Description
	<a href="#">DeserializeXmlNode(String)</a>	Deserializes the XmlNode from a JSON string.
	<a href="#">DeserializeXmlNode(String, String)</a>	Deserializes the XmlNode from a JSON string nested in a root element
	<a href="#">DeserializeXmlNode(String, String, Boolean)</a>	Deserializes the XmlNode from a JSON string nested in a root element

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### DeserializeXmlNode(String)

Deserializes the XmlNode from a JSON string.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static XmlDocument DeserializeXmlNode(  
    string value  
)
```

```
Public Shared Function DeserializeXmlNode ( _  
    value As String _  
) As XmlDocument
```

```
public:  
static XmlDocument^ DeserializeXmlNode(  
    String^ value  
)
```

#### ▣ Parameters

##### **value (String)**

The JSON string.

#### ▣ Return Value

The deserialized XmlNode

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# DeserializeXmlNode Method (value, deserializeRootElementName)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## DeserializeXmlNode(String, String)

Deserializes the XmlNode from a JSON string nested in a root element.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public static XmlDocument DeserializeXmlNode(  
    string value,  
    string deserializeRootElementName  
)
```

```
Public Shared Function DeserializeXmlNode ( _  
    value As String, _  
    deserializeRootElementName As String _  
) As XmlDocument
```

```
public:  
static XmlDocument^ DeserializeXmlNode(  
    String^ value,  
    String^ deserializeRootElementName  
)
```

### Parameters

#### value (String)

The JSON string.

#### deserializeRootElementName (String)

The name of the root element to append when deserializing.

### Return Value

The deserialized XmlNode

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## DeserializeXmlNode Method (value, deserializeRootElementName, writeArrayAttribute)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### DeserializeXmlNode(String, String, Boolean)

Deserializes the XmlNode from a JSON string nested in a root element.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static XmlDocument DeserializeXmlNode(  
    string value,  
    string deserializeRootElementName,  
    bool writeArrayAttribute  
)
```

```
Public Shared Function DeserializeXmlNode ( _  
    value As String, _  
    deserializeRootElementName As String, _  
    writeArrayAttribute As Boolean _  
) As XmlDocument
```

```
public:  
static XmlDocument^ DeserializeXmlNode(  
    String^ value,  
    String^ deserializeRootElementName,  
    bool writeArrayAttribute  
)
```

#### ▣ Parameters

##### **value (String)**

The JSON string.

##### **deserializeRootElementName (String)**

The name of the root element to append when deserializing.

***writeArrayAttribute*** ([Boolean](#))

A flag to indicate whether to write the Json.NET array attribute. This attribute helps preserve arrays when converting the written XML back to JSON.

▣ **Return Value**

The deserialized XmlNode

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## DeserializeXmlNode()

### Members

Icon	Member	Description
	<a href="#">DeserializeXmlNode(String)</a>	Deserializes the <a href="#">XmlNode</a> from a JSON string.
	<a href="#">DeserializeXmlNode(String, String)</a>	Deserializes the <a href="#">XmlNode</a> from a JSON string nested in a root element.
	<a href="#">DeserializeXmlNode(String, String, Boolean)</a>	Deserializes the <a href="#">XmlNode</a> from a JSON string nested in a root element.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# DeserializeXmlNode Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## DeserializeXmlNode(String)

Deserializes the [XmlNode](#) from a JSON string.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static XDocument DeserializeXmlNode(  
    string value  
)
```

```
Public Shared Function DeserializeXmlNode ( _  
    value As String _  
) As XDocument
```

```
public:  
static XDocument^ DeserializeXmlNode(  
    String^ value  
)
```

### ▣ Parameters

#### **value (String)**

The JSON string.

### ▣ Return Value

The deserialized XmlNode

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## DeserializeXmlNode Method (value, deserializeRootElementName)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### DeserializeXmlNode(String, String)

Deserializes the [XmlNode](#) from a JSON string nested in a root element.

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static <a href="#">XDocument</a> DeserializeXmlNode(     <a href="#">string</a> value,     <a href="#">string</a> deserializeRootElementName )</pre>		
	<pre>Public Shared Function DeserializeXmlNode ( _     value As <a href="#">String</a>, _     deserializeRootElementName As <a href="#">String</a> _ ) As <a href="#">XDocument</a></pre>	
		<pre>public: static <a href="#">XDocument</a>^ DeserializeXmlNode(     <a href="#">String</a>^ value,     <a href="#">String</a>^ deserializeRootElementName )</pre>

#### Parameters

##### value ([String](#))

The JSON string.

##### deserializeRootElementName ([String](#))

The name of the root element to append when deserializing.

#### Return Value

The deserialized XmlNode

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## DeserializeXmlNode Method (value, deserializeRootElementName, writeArrayAttribute)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### DeserializeXmlNode(String, String, Boolean)

Deserializes the [XmlNode](#) from a JSON string nested in a root element.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static XDocument DeserializeXmlNode(  
    string value,  
    string deserializeRootElementName,  
    bool writeArrayAttribute  
)
```

```
Public Shared Function DeserializeXmlNode ( _  
    value As String, _  
    deserializeRootElementName As String, _  
    writeArrayAttribute As Boolean _  
) As XDocument
```

```
public:  
static XDocument^ DeserializeXmlNode(  
    String^ value,  
    String^ deserializeRootElementName,  
    bool writeArrayAttribute  
)
```

#### ▣ Parameters

##### **value** ([String](#))

The JSON string.

##### **deserializeRootElementName** ([String](#))

The name of the root element to append when deserializing.

***writeArrayAttribute*** ([Boolean](#))

A flag to indicate whether to write the Json.NET array attribute. This attribute helps preserve arrays when converting the written XML back to JSON.

▣ **Return Value**

The deserialized XNode

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **False**

Represents JavaScript's boolean value false as a string. This field is read-only.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static readonly string False
```

```
Public Shared ReadOnly False As String
```

```
public:  
static initonly String^ False
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **NaN**

Represents JavaScript's NaN as a string. This field is read-only.

C# ▼

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static readonly string NaN
```

```
Public Shared ReadOnly NaN As String
```

```
public:  
static initonly String^ NaN
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **NegativeInfinity**

Represents JavaScript's negative infinity as a string. This field is read-only.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static readonly string NegativeInfinity
```

```
Public Shared ReadOnly NegativeInfinity As String
```

```
public:  
static initonly String^ NegativeInfinity
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **Null**

Represents JavaScript's null as a string. This field is read-only.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static readonly string Null
```

```
Public Shared ReadOnly Null As String
```

```
public:  
static initonly String^ Null
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **PopulateObject()**

C# ▼

▣ **Members**

Icon	Member	Description
	<a href="#">PopulateObject(String, Object)</a>	Populates the object with values from the JSON string.
	<a href="#">PopulateObject(String, Object, JsonSerializerSettings)</a>	Populates the object with values from the JSON string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# PopulateObject Method (value, target)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## PopulateObject(String, Object)

Populates the object with values from the JSON string.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static void PopulateObject(  
    string value,  
    Object target  
)
```

```
Public Shared Sub PopulateObject ( _  
    value As String, _  
    target As Object _  
)
```

```
public:  
static void PopulateObject(  
    String^ value,  
    Object^ target  
)
```

### ▣ Parameters

#### **value** ([String](#))

The JSON to populate values from.

#### **target** ([Object](#))

The target object to populate values onto.

## PopulateObject Method (value, target, settings)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### PopulateObject(String, Object, JsonSerializerSettings)

Populates the object with values from the JSON string.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static void PopulateObject(  
    string value,  
    Object target,  
    JsonSerializerSettings settings  
)
```

```
Public Shared Sub PopulateObject ( _  
    value As String, _  
    target As Object, _  
    settings As JsonSerializerSettings _  
)
```

```
public:  
static void PopulateObject(  
    String^ value,  
    Object^ target,  
    JsonSerializerSettings^ settings  
)
```

#### ▣ Parameters

##### **value** ([String](#))

The JSON to populate values from.

##### **target** ([Object](#))

The target object to populate values onto.

**settings ([JsonSerializerSettings](#))**

The [JsonSerializerSettings](#) used to deserialize the object. If this is null, default serialization settings will be is used.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# PopulateObjectAsync Method (value, target, settings)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## PopulateObjectAsync(String, Object, JsonSerializerSettings)

Asynchronously populates the object with values from the JSON string.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public static Task PopulateObjectAsync(  
    string value,  
    Object target,  
    JsonSerializerSettings settings  
)
```

```
Public Shared Function PopulateObjectAsync ( _  
    value As String, _  
    target As Object, _  
    settings As JsonSerializerSettings _  
) As Task
```

```
public:  
static Task^ PopulateObjectAsync(  
    String^ value,  
    Object^ target,  
    JsonSerializerSettings^ settings  
)
```

### Parameters

#### **value** ([String](#))

The JSON to populate values from.

#### **target** ([Object](#))

The target object to populate values onto.

**settings ([JsonSerializerSettings](#))**

The [JsonSerializerSettings](#) used to deserialize the object. If this is null, default serialization settings will be is used.

▣ **Return Value**

A task that represents the asynchronous populate operation.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **PositiveInfinity**

Represents JavaScript's positive infinity as a string. This field is read-only.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static readonly string PositiveInfinity
```

```
Public Shared ReadOnly PositiveInfinity As String
```

```
public:  
static initonly String^ PositiveInfinity
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# SerializeObject Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **SerializeObject()**

C# ▼

## Members

Icon	Member	Description
	<a href="#">SerializeObject(Object)</a>	Serializes the specified object to a JSON string.
	<a href="#">SerializeObject(Object, Formatting)</a>	Serializes the specified object to a JSON string.
	<a href="#">SerializeObject(Object, JsonConverters[])</a>	Serializes the specified object to a JSON string using a collection of <a href="#">JsonConverter</a> .
	<a href="#">SerializeObject(Object, Formatting, JsonConverters[])</a>	Serializes the specified object to a JSON string using a collection of <a href="#">JsonConverter</a> .
	<a href="#">SerializeObject(Object, JsonSerializerSettings)</a>	Serializes the specified object to a JSON string using a collection of <a href="#">JsonConverter</a> .
	<a href="#">SerializeObject(Object, Formatting, JsonSerializerSettings)</a>	Serializes the specified object to a JSON string using a collection of <a href="#">JsonConverter</a> .

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

# SerializeObject Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## SerializeObject(Object)

C#

Serializes the specified object to a JSON string.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string SerializeObject(  
    Object value  
)
```

```
Public Shared Function SerializeObject ( _  
    value As Object _  
) As String
```

```
public:  
static String^ SerializeObject(  
    Object^ value  
)
```

### Parameters

#### *value* ([Object](#))

The object to serialize.

### Return Value

A JSON string representation of the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# SerializeObject Method (value, formatting)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## SerializeObject(Object, Formatting)

Serializes the specified object to a JSON string.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string SerializeObject(  
    Object value,  
    Formatting formatting  
)
```

```
Public Shared Function SerializeObject ( _  
    value As Object, _  
    formatting As Formatting _  
) As String
```

```
public:  
static String^ SerializeObject(  
    Object^ value,  
    Formatting formatting  
)
```

### ▣ Parameters

#### **value** ([Object](#))

The object to serialize.

#### **formatting** ([Formatting](#))

Indicates how the output is formatted.

### ▣ Return Value

A JSON string representation of the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## SerializeObject Method (value, formatting, converters)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### SerializeObject(Object, Formatting, JsonSerializer[])

Serializes the specified object to a JSON string using a collection of [JsonConverter](#).

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string SerializeObject(  
    Object value,  
    Formatting formatting,  
    params JsonSerializer[] converters  
)
```

```
Public Shared Function SerializeObject ( _  
    value As Object, _  
    formatting As Formatting, _  
    ParamArray converters As JsonSerializer() _  
) As String
```

```
public:  
static String^ SerializeObject(  
    Object^ value,  
    Formatting formatting,  
    ... array<JsonConverter^>^ converters  
)
```

#### ▣ Parameters

##### **value** ([Object](#))

The object to serialize.

##### **formatting** ([Formatting](#))

Indicates how the output is formatted.

***converters*** ([JsonConverter\[\]](#))

A collection converters used while serializing.

▣ **Return Value**

A JSON string representation of the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## SerializeObject Method (value, formatting, settings)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### SerializeObject(Object, Formatting, JsonSerializerSettings)

Serializes the specified object to a JSON string using a collection of [JsonConverter](#).

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string SerializeObject(  
    Object value,  
    Formatting formatting,  
    JsonSerializerSettings settings  
)
```

```
Public Shared Function SerializeObject ( _  
    value As Object, _  
    formatting As Formatting, _  
    settings As JsonSerializerSettings _  
) As String
```

```
public:  
static String^ SerializeObject(  
    Object^ value,  
    Formatting formatting,  
    JsonSerializerSettings^ settings  
)
```

#### ▣ Parameters

##### **value** ([Object](#))

The object to serialize.

##### **formatting** ([Formatting](#))

Indicates how the output is formatted.

***settings* ([JsonSerializerSettings](#))**

The [JsonSerializerSettings](#) used to serialize the object. If this is null default serialization settings will be used.

▣ **Return Value**

A JSON string representation of the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## SerializeObject Method (value, converters)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### SerializeObject(Object, JsonSerializer[])

Serializes the specified object to a JSON string using a collection of [JsonConverter](#).

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string SerializeObject(  
    Object value,  
    params JsonSerializer[] converters  
)
```

```
Public Shared Function SerializeObject ( _  
    value As Object, _  
    ParamArray converters As JsonSerializer() _  
) As String
```

```
public:  
static String^ SerializeObject(  
    Object^ value,  
    ... array<JsonConverter^>^ converters  
)
```

#### Parameters

##### **value (Object)**

The object to serialize.

##### **converters (JsonSerializer[])**

A collection converters used while serializing.

#### Return Value

A JSON string representation of the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

# SerializeObject Method (value, settings)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## SerializeObject(Object, JsonSerializerSettings)

Serializes the specified object to a JSON string using a collection of [JsonConverter](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string SerializeObject(  
    Object value,  
    JsonSerializerSettings settings  
)
```

```
Public Shared Function SerializeObject ( _  
    value As Object, _  
    settings As JsonSerializerSettings _  
) As String
```

```
public:  
static String^ SerializeObject(  
    Object^ value,  
    JsonSerializerSettings^ settings  
)
```

### Parameters

#### **value (Object)**

The object to serialize.

#### **settings (JsonSerializerSettings)**

The [JsonSerializerSettings](#) used to serialize the object. If this is null default serialization settings will be used.

### Return Value

A JSON string representation of the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### SerializeObjectAsync()

#### Members

Icon	Member	Description
	<a href="#">SerializeObjectAsync(Void)</a>	
	<a href="#">SerializeObjectAsync(Void, String)</a>	
	<a href="#">SerializeObjectAsync(Void, String, Object)</a>	

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### SerializeObjectAsync(Void)

[Missing <summary> documentation for  
"M:Newtonsoft.Json.JsonConvert.SerializeObjectAsync(System.Vo

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static Task SerializeObjectAsync(  
    void value  
)
```

```
Public Shared Function SerializeObjectAsync ( _  
    value As Void _  
) As Task
```

```
public:  
static Task^ SerializeObjectAsync(  
    void value  
)
```

#### ▣ Parameters

*value* ([Void](#))

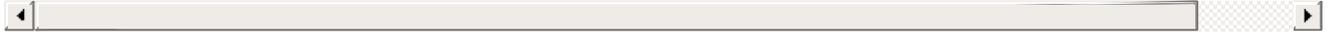
[Missing <param name="value"/> documentation for  
"M:Newtonsoft.Json.JsonConvert.SerializeObjectAsync(Syste

#### ▣ Return Value

[Missing <returns> documentation for  
"M:Newtonsoft.Json.JsonConvert.SerializeObjectAsync(System.Vo

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)



# SerializeObjectAsync Method (value, formatting)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

SerializeObjectAsync(Void, String)

C# ▼

[Missing <summary> documentation for  
"M:Newtonsoft.Json.JsonConvert.SerializeObjectAsync(System.Vo

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static Task SerializeObjectAsync(  
    void value,  
    string formatting  
)
```

```
Public Shared Function SerializeObjectAsync ( _  
    value As Void, _  
    formatting As String _  
) As Task
```

```
public:  
static Task^ SerializeObjectAsync(  
    void value,  
    String^ formatting  
)
```

## ▣ Parameters

**value** ([Void](#))

[Missing <param name="value"/> documentation for  
"M:Newtonsoft.Json.JsonConvert.SerializeObjectAsync(Syste

**formatting** ([String](#))

[Missing <param name="formatting"/> documentation for  
"M:Newtonsoft.Json.JsonConvert.SerializeObjectAsync(Syste

## Return Value

[Missing <returns> documentation for  
"M:Newtonsoft.Json.JsonConvert.SerializeObjectAsync(System.Vo

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



# SerializeObjectAsync Method (value, formatting, settings)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

SerializeObjectAsync(Void, String, Object)

C#

[Missing <summary> documentation for  
"M:Newtonsoft.Json.JsonConvert.SerializeObjectAsync(System.Vo

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static Task SerializeObjectAsync(  
    void value,  
    string formatting,  
    Object settings  
)
```

```
Public Shared Function SerializeObjectAsync ( _  
    value As Void, _  
    formatting As String, _  
    settings As Object _  
) As Task
```

```
public:  
static Task^ SerializeObjectAsync(  
    void value,  
    String^ formatting,  
    Object^ settings  
)
```

## ▣ Parameters

**value** ([Void](#))

[Missing <param name="value"/> documentation for  
"M:Newtonsoft.Json.JsonConvert.SerializeObjectAsync(Syste

*formatting* ([String](#))

[Missing <param name="formatting"/> documentation for  
"M:Newtonsoft.Json.JsonConvert.SerializeObjectAsync(Syste

*settings* ([Object](#))

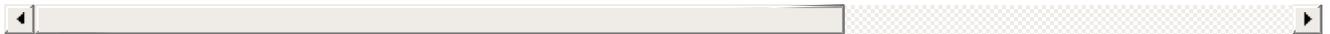
[Missing <param name="settings"/> documentation for  
"M:Newtonsoft.Json.JsonConvert.SerializeObjectAsync(Syste

#### ▣ Return Value

[Missing <returns> documentation for  
"M:Newtonsoft.Json.JsonConvert.SerializeObjectAsync(System.Vo

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## SerializeXmlNode()

### Members

Icon	Member	Description
	<a href="#">SerializeXmlNode(XmlNode)</a>	Serializes the XML node to a JSON string.
	<a href="#">SerializeXmlNode(XmlNode, Formatting)</a>	Serializes the XML node to a JSON string.
	<a href="#">SerializeXmlNode(XmlNode, Formatting, Boolean)</a>	Serializes the XML node to a JSON string.

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# SerializeXmlNode Method (node)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## SerializeXmlNode(XmlNode)

Serializes the XML node to a JSON string.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string SerializeXmlNode(  
    XmlNode node  
)
```

```
Public Shared Function SerializeXmlNode ( _  
    node As XmlNode _  
) As String
```

```
public:  
static String^ SerializeXmlNode(  
    XmlNode^ node  
)
```

### ▣ Parameters

#### ***node*** ([XmlNode](#))

The node to serialize.

### ▣ Return Value

A JSON string of the XmlNode.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# SerializeXmlNode Method (node, formatting)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## SerializeXmlNode(XmlNode, Formatting)

C#

Serializes the XML node to a JSON string.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string SerializeXmlNode(  
    XmlNode node,  
    Formatting formatting  
)
```

```
Public Shared Function SerializeXmlNode ( _  
    node As XmlNode, _  
    formatting As Formatting _  
) As String
```

```
public:  
static String^ SerializeXmlNode(  
    XmlNode^ node,  
    Formatting formatting  
)
```

### Parameters

#### **node** ([XmlNode](#))

The node to serialize.

#### **formatting** ([Formatting](#))

Indicates how the output is formatted.

### Return Value

A JSON string of the XmlNode.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

# SerializeXmlNode Method (node, formatting, omitRootObject)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## SerializeXmlNode(XmlNode, Formatting, Boolean)

Serializes the XML node to a JSON string.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string SerializeXmlNode(  
    XmlNode node,  
    Formatting formatting,  
    bool omitRootObject  
)
```

```
Public Shared Function SerializeXmlNode ( _  
    node As XmlNode, _  
    formatting As Formatting, _  
    omitRootObject As Boolean _  
) As String
```

```
public:  
static String^ SerializeXmlNode(  
    XmlNode^ node,  
    Formatting formatting,  
    bool omitRootObject  
)
```

### ▣ Parameters

#### **node** ([XmlNode](#))

The node to serialize.

#### **formatting** ([Formatting](#))

Indicates how the output is formatted.

***omitRootObject*** ([Boolean](#))

Omits writing the root object.

▣ **Return Value**

A JSON string of the XmlNode.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **SerializeXmlNode()**

## Members

Icon	Member	Description
	<a href="#">SerializeXmlNode(XObject)</a>	Serializes the <a href="#">XmlNode</a> to a JSON string.
	<a href="#">SerializeXmlNode(XObject, Formatting)</a>	Serializes the <a href="#">XmlNode</a> to a JSON string.
	<a href="#">SerializeXmlNode(XObject, Formatting, Boolean)</a>	Serializes the <a href="#">XmlNode</a> to a JSON string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (node)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### SerializeXmlNode(XObject)

Serializes the [XmlNode](#) to a JSON string.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string SerializeXmlNode(  
    XmlNode node  
)
```

```
Public Shared Function SerializeXmlNode ( _  
    node As XmlNode _  
) As String
```

```
public:  
static String^ SerializeXmlNode(  
    XmlNode^ node  
)
```

#### ▣ Parameters

##### ***node*** ([XObject](#))

The node to convert to JSON.

#### ▣ Return Value

A JSON string of the XmlNode.

## (node, formatting)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### SerializeXNode(XObject, Formatting)

Serializes the [XNode](#) to a JSON string.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string SerializeXNode(     XObject node,     Formatting formatting )</pre>		
	<pre>Public Shared Function SerializeXNode ( _     node As XObject, _     formatting As Formatting _ ) As String</pre>	
		<pre>public: static String^ SerializeXNode(     XObject^ node,     Formatting formatting )</pre>

#### ▣ Parameters

##### **node** ([XObject](#))

The node to convert to JSON.

##### **formatting** ([Formatting](#))

Indicates how the output is formatted.

#### ▣ Return Value

A JSON string of the XNode.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## SerializeXmlNode Method (node, formatting, omitRootObject)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### SerializeXmlNode(XObject, Formatting, Boolean)

Serializes the [XmlNode](#) to a JSON string.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string SerializeXmlNode(  
    XmlNode node,  
    Formatting formatting,  
    bool omitRootObject  
)
```

```
Public Shared Function SerializeXmlNode ( _  
    node As XmlNode, _  
    formatting As Formatting, _  
    omitRootObject As Boolean _  
) As String
```

```
public:  
static String^ SerializeXmlNode(  
    XmlNode^ node,  
    Formatting formatting,  
    bool omitRootObject  
)
```

#### ▣ Parameters

##### **node (XmlNode)**

The node to serialize.

##### **formatting (Formatting)**

Indicates how the output is formatted.

***omitRootObject*** ([Boolean](#))

Omits writing the root object.

▣ **Return Value**

A JSON string of the XNode.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Members

Icon	Member	Description
	<a href="#">ToString(DateTime)</a>	Converts the <a href="#">DateTime</a> to its JSON string representation.
	<a href="#">ToString(DateTime, DateFormatHandling, DateTimeZoneHandling)</a>	Converts the <a href="#">DateTime</a> to its JSON string representation using the <a href="#">DateFormatHandling</a> specified.
	<a href="#">ToString(DateTimeOffset)</a>	Converts the <a href="#">DateTimeOffset</a> to its JSON string representation.
	<a href="#">ToString(DateTimeOffset, DateFormatHandling)</a>	Converts the <a href="#">DateTimeOffset</a> to its JSON string representation using the <a href="#">DateFormatHandling</a> specified.
	<a href="#">ToString(Boolean)</a>	Converts the <a href="#">Boolean</a> to its JSON string representation.
	<a href="#">ToString(Char)</a>	Converts the <a href="#">Char</a> to its JSON string representation.
	<a href="#">ToString(Enum)</a>	Converts the <a href="#">Enum</a> to its JSON string representation.

 <a href="#">ToString(Int32)</a>	Converts the <a href="#">Int32</a> to its JSON string representation.
 <a href="#">ToString(Int16)</a>	Converts the <a href="#">Int16</a> to its JSON string representation.
 <a href="#">ToString(UInt16)</a>	Converts the <a href="#">UInt16</a> to its JSON string representation.
 <a href="#">ToString(UInt32)</a>	Converts the <a href="#">UInt32</a> to its JSON string representation.
 <a href="#">ToString(Int64)</a>	Converts the <a href="#">Int64</a> to its JSON string representation.
 <a href="#">ToString(UInt64)</a>	Converts the <a href="#">UInt64</a> to its JSON string representation.
 <a href="#">ToString(Single)</a>	Converts the <a href="#">Single</a> to its JSON string representation.
 <a href="#">ToString(Double)</a>	Converts the <a href="#">Double</a> to its JSON string representation.
 <a href="#">ToString(Byte)</a>	Converts the <a href="#">Byte</a> to its JSON string representation.

		representation.
	<a href="#">ToString(SByte)</a>	Converts the <a href="#">SByte</a> to its JSON string representation.
	<a href="#">ToString(Decimal)</a>	Converts the <a href="#">Decimal</a> to its JSON string representation.
	<a href="#">ToString(Guid)</a>	Converts the <a href="#">Guid</a> to its JSON string representation.
	<a href="#">ToString(TimeSpan)</a>	Converts the <a href="#">TimeSpan</a> to its JSON string representation.
	<a href="#">ToString(Uri)</a>	Converts the <a href="#">Uri</a> to its JSON string representation.
	<a href="#">ToString(String)</a>	Converts the <a href="#">String</a> to its JSON string representation.
	<a href="#">ToString(String, Char)</a>	Converts the <a href="#">String</a> to its JSON string representation.
	<a href="#">ToString(Object)</a>	Converts the <a href="#">Object</a> to its JSON string representation.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## ToString(Boolean)

Converts the [Boolean](#) to its JSON string representation.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     bool value )</pre>		
	<pre>Public Shared Function ToString ( _     value As Boolean _ ) As String</pre>	
		<pre>public: static String^ ToString(     bool value )</pre>

### ▣ Parameters

#### **value** ([Boolean](#))

The value to convert.

### ▣ Return Value

A JSON string representation of the [Boolean](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Converts the [Byte](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     byte value )</pre>		
	<pre>Public Shared Function ToString ( _     value As Byte _ ) As String</pre>	
		<pre>public: static String^ ToString(     unsigned char value )</pre>

## ▣ Parameters

### **value (Byte)**

The value to convert.

## ▣ Return Value

A JSON string representation of the [Byte](#).

Converts the [Char](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     char value )</pre>	<pre>Public Shared Function ToString ( _     value As Char _ ) As String</pre>	<pre>public: static String^ ToString(     wchar_t value )</pre>

## ▣ Parameters

### **value (Char)**

The value to convert.

## ▣ Return Value

A JSON string representation of the [Char](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## Tostring(DateTime)

Converts the [DateTime](#) to its JSON string representation.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     DateTime value )</pre>		
	<pre>Public Shared Function ToString ( _     value As DateTime _ ) As String</pre>	
		<pre>public: static String^ ToString(     DateTime value )</pre>

### ▣ Parameters

#### **value** ([DateTime](#))

The value to convert.

### ▣ Return Value

A JSON string representation of the [DateTime](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## ToString Method (value, format, timeZoneHandling)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

### ToString(DateTime, DateFormatHandling, DateTimeZoneHandling)

Converts the [DateTime](#) to its JSON string representation using the [DateFormatHandling](#) specified.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    DateTime value,  
    DateFormatHandling format,  
    DateTimeZoneHandling timeZoneHandling  
)
```

```
Public Shared Function ToString ( _  
    value As DateTime, _  
    format As DateFormatHandling, _  
    timeZoneHandling As DateTimeZoneHandling _  
) As String
```

```
public:  
static String^ ToString(  
    DateTime value,  
    DateFormatHandling format,  
    DateTimeZoneHandling timeZoneHandling  
)
```

#### Parameters

##### **value** ([DateTime](#))

The value to convert.

##### **format** ([DateFormatHandling](#))

The format the date will be converted to.

***timeZoneHandling*** ([DateTimeZoneHandling](#))

The time zone handling when the date is converted to a string.

▣ **Return Value**

A JSON string representation of the [DateTime](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## Tostring(DateTimeOffset)

C#

Converts the [DateTimeOffset](#) to its JSON string representation.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    DateTimeOffset value  
)
```

```
Public Shared Function ToString ( _  
    value As DateTimeOffset _  
) As String
```

```
public:  
static String^ ToString(  
    DateTimeOffset value  
)
```

### ▣ Parameters

**value** ([DateTimeOffset](#))

The value to convert.

### ▣ Return Value

A JSON string representation of the [DateTimeOffset](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# Tostring Method (value, format)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## Tostring(DateTimeOffset, DateFormatHandling)

Converts the [DateTimeOffset](#) to its JSON string representation using the [DateFormatHandling](#) specified.

### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     DateTimeOffset value,     DateFormatHandling format )</pre>		
	<pre>Public Shared Function ToString ( _     value As DateTimeOffset, _     format As DateFormatHandling _ ) As String</pre>	
		<pre>public: static String^ ToString(     DateTimeOffset value,     DateFormatHandling format )</pre>

### Parameters

**value** ([DateTimeOffset](#))

The value to convert.

**format** ([DateFormatHandling](#))

The format the date will be converted to.

### Return Value

A JSON string representation of the [DateTimeOffset](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## Tostring(Decimal)

C#

Converts the [Decimal](#) to its JSON string representation.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    decimal value  
)
```

```
Public Shared Function ToString ( _  
    value As Decimal _  
) As String
```

```
public:  
static String^ ToString(  
    Decimal value  
)
```

### ▣ Parameters

#### **value** ([Decimal](#))

The value to convert.

### ▣ Return Value

A JSON string representation of the [SByte](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Converts the [Double](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     double value )</pre>	<pre>Public Shared Function ToString ( _     value As Double _ ) As String</pre>	<pre>public: static String^ ToString(     double value )</pre>

## ▣ Parameters

### **value (Double)**

The value to convert.

## ▣ Return Value

A JSON string representation of the [Double](#).

Converts the [Enum](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     Enum value )</pre>	<pre>Public Shared Function ToString ( _     value As Enum _ ) As String</pre>	<pre>public: static String^ ToString(     Enum^ value )</pre>

## ▣ Parameters

### **value (Enum)**

The value to convert.

## ▣ Return Value

A JSON string representation of the [Enum](#).

Converts the [Guid](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     Guid value )</pre>	<pre>Public Shared Function ToString ( _     value As Guid _ ) As String</pre>	<pre>public: static String^ ToString(     Guid value )</pre>

## ▣ Parameters

### **value (Guid)**

The value to convert.

## ▣ Return Value

A JSON string representation of the [Guid](#).

Converts the [Int16](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     short value )</pre>	<pre>Public Shared Function ToString ( _     value As Short _ ) As String</pre>	
<pre>public: static String^ ToString(     short value )</pre>		

## ▣ Parameters

### **value** ([Int16](#))

The value to convert.

## ▣ Return Value

A JSON string representation of the [Int16](#).

Converts the [Int32](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     int value )</pre>	<pre>Public Shared Function ToString ( _     value As Integer _ ) As String</pre>	<pre>public: static String^ ToString(     int value )</pre>

## ▣ Parameters

### **value** ([Int32](#))

The value to convert.

## ▣ Return Value

A JSON string representation of the [Int32](#).

Converts the [Int64](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     long value )</pre>		
	<pre>Public Shared Function ToString ( _     value As Long _ ) As String</pre>	
		<pre>public: static String^ ToString(     long long value )</pre>

## ▣ Parameters

### **value (Int64)**

The value to convert.

## ▣ Return Value

A JSON string representation of the [Int64](#).

Converts the [Object](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     Object value )</pre>		
	<pre>Public Shared Function ToString ( _     value As Object _ ) As String</pre>	
		<pre>public: static String^ ToString(     Object^ value )</pre>

## ▣ Parameters

### **value (Object)**

The value to convert.

## ▣ Return Value

A JSON string representation of the [Object](#).

Converts the [SByte](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     sbyte value )</pre>		
	<pre>Public Shared Function ToString ( _     value As SByte _ ) As String</pre>	
		<pre>public: static String^ ToString(     signed char value )</pre>

## ▣ Parameters

### **value (SByte)**

The value to convert.

## ▣ Return Value

A JSON string representation of the [SByte](#).

Converts the [Single](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     float value )</pre>	<pre>Public Shared Function ToString ( _     value As Single _ ) As String</pre>	<pre>public: static String^ ToString(     float value )</pre>

## ▣ Parameters

### **value (Single)**

The value to convert.

## ▣ Return Value

A JSON string representation of the [Single](#).

Converts the [String](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     string value )</pre>	<pre>Public Shared Function ToString ( _     value As String _ ) As String</pre>	<pre>public: static String^ ToString(     String^ value )</pre>

## ▣ Parameters

### **value (String)**

The value to convert.

## ▣ Return Value

A JSON string representation of the [String](#).

## ToString Method (value, delimiter)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► [ToString\(String, Char\)](#)

Converts the [String](#) to its JSON string representation.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    string value,  
    char delimiter  
)
```

```
Public Shared Function ToString ( _  
    value As String, _  
    delimiter As Char _  
) As String
```

```
public:  
static String^ ToString(  
    String^ value,  
    wchar_t delimiter  
)
```

### ▣ Parameters

#### **value** ([String](#))

The value to convert.

#### **delimiter** ([Char](#))

The string delimiter character.

### ▣ Return Value

A JSON string representation of the [String](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ►

## Tostring(TimeSpan)

Converts the [TimeSpan](#) to its JSON string representation.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     TimeSpan value )</pre>		
	<pre>Public Shared Function ToString ( _     value As TimeSpan _ ) As String</pre>	
		<pre>public: static String^ ToString(     TimeSpan value )</pre>

### ▣ Parameters

#### **value** ([TimeSpan](#))

The value to convert.

### ▣ Return Value

A JSON string representation of the [TimeSpan](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Converts the [UInt16](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     ushort value )</pre>	<pre>Public Shared Function ToString ( _     value As UShort _ ) As String</pre>	<pre>public: static String^ ToString(     unsigned short value )</pre>

## ▣ Parameters

### ***value*** ([UInt16](#))

The value to convert.

## ▣ Return Value

A JSON string representation of the [UInt16](#).

Converts the [UInt32](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     uint value )</pre>		
	<pre>Public Shared Function ToString ( _     value As UInteger _ ) As String</pre>	
		<pre>public: static String^ ToString(     unsigned int value )</pre>

## ▣ Parameters

### ***value*** ([UInt32](#))

The value to convert.

## ▣ Return Value

A JSON string representation of the [UInt32](#).

Converts the [UInt64](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     ulong value )</pre>	<pre>Public Shared Function ToString ( _     value As ULong _ ) As String</pre>	<pre>public: static String^ ToString(     unsigned long long value )</pre>

## ▣ Parameters

### ***value*** ([UInt64](#))

The value to convert.

## ▣ Return Value

A JSON string representation of the [UInt64](#).

Converts the [Uri](#) to its JSON string representation.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static string ToString(     Uri value )</pre>		
	<pre>Public Shared Function ToString ( _     value As Uri _ ) As String</pre>	
		<pre>public: static String^ ToString(     Uri^ value )</pre>

## ▣ Parameters

### **value (Uri)**

The value to convert.

## ▣ Return Value

A JSON string representation of the [Uri](#).

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **True**

Represents JavaScript's boolean value true as a string. This field is read-only.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static readonly string True
```

```
Public Shared ReadOnly True As String
```

```
public:  
static initonly String^ True
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **Undefined**

Represents JavaScript's undefined as a string. This field is read-only. C# ▼

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static readonly string Undefined
```

```
Public Shared ReadOnly Undefined As String
```

```
public:  
static initonly String^ Undefined
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonConverter Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonConverter**

Converts an object to and from JSON.

## [-] Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract class JsonConverter
```

```
Public MustInherit Class JsonConverter
```

```
public ref class JsonConverter abstract
```

## [-] Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declare <input checked="" type="checkbox"/> Inherit

Icon	Member	Description
	<a href="#">JsonConverter()</a>	Initializes a new instance of the <b>JsonConverter</b> class
	<a href="#">CanConvert(Type)</a>	Determines whether this instance can convert the specified object type.
	<a href="#">CanRead</a>	Gets a value indicating whether this <b>JsonConverter</b> can read JSON.
	<a href="#">CanWrite</a>	Gets a value indicating whether this

		<b>JsonConverter</b> can write JSON.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetSchema()</a>	Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a> .
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ReadJson(JsonReader, Type, Object, JsonSerializer)</a>	Reads the JSON representation of the object.
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .

		(Inherited from <a href="#">Object</a> .)
	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of the object.

## ▣ Inheritance Hierarchy

### [Object](#)

#### └─ **JsonConverter**

- ├─ [BinaryConverter](#)
- ├─ [BsonObjectIdConverter](#)
- ├─ [CustomCreationConverter<T>](#)
- ├─ [DataSetConverter](#)
- ├─ [DataTableConverter](#)
- ├─ [DateTimeConverterBase](#)
- ├─ [EntityKeyMemberConverter](#)
- ├─ [ExpandoObjectConverter](#)
- ├─ [KeyValuePairConverter](#)
- ├─ [RegexConverter](#)
- ├─ [StringEnumConverter](#)
- ├─ [VersionConverter](#)
- ├─ [XmlNodeConverter](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConverter](#) ►

### JsonConverter()

Initializes a new instance of the [JsonConverter](#) class

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<code>protected</code> JsonConverter()		
<code>Protected Sub</code> New		
<code>protected:</code> JsonConverter()		

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (objectType)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConverter](#) ►

### CanConvert(Type)

Determines whether this instance can convert the specified object type. C#

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract bool CanConvert(  
    Type objectType  
)
```

```
Public MustOverride Function CanConvert ( _  
    objectType As Type _  
) As Boolean
```

```
public:  
virtual bool CanConvert(  
    Type^ objectType  
) abstract
```

#### ▣ Parameters

##### ***objectType*** ([Type](#))

Type of the object.

#### ▣ Return Value

**true** if this instance can convert the specified object type; otherwise, **false**.

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConverter](#) ► **CanRead**

Gets a value indicating whether this [JsonConverter](#) can read JSON.

C# ▼

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual bool CanRead { get; }
```

```
Public Overridable ReadOnly Property CanRead As Bool  
    Get
```

```
public:  
virtual property bool CanRead {  
    bool get ();  
}
```

## ▣ Value

true if this [JsonConverter](#) can read JSON; otherwise, false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConverter](#) ► **CanWrite**

Gets a value indicating whether this [JsonConverter](#) can write JSON.

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual bool CanWrite { get; }
```

```
Public Overridable ReadOnly Property CanWrite As Boolean  
    Get
```

```
public:  
virtual property bool CanWrite {  
    bool get ();  
}
```

## Value

true if this [JsonConverter](#) can write JSON; otherwise, false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConverter](#) ► **GetSchema()**

Gets the [JsonSchema](#) of the JSON produced by the `JsonConverter`.

C#

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual JsonSchema GetSchema()
```

```
Public Overridable Function GetSchema As JsonSchema
```

```
public:  
virtual JsonSchema^ GetSchema()
```

## ▣ Return Value

The [JsonSchema](#) of the JSON produced by the `JsonConverter`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## ReadJson Method (reader, objectType, existingValue, serializer)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConverter](#) ►

**ReadJson(JsonReader, Type, Object, JsonSerializer)**

C#

Reads the JSON representation of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract Object ReadJson(  
    JsonReader reader,  
    Type objectType,  
    Object existingValue,  
    JsonSerializer serializer  
)
```

```
Public MustOverride Function ReadJson ( _  
    reader As JsonReader, _  
    objectType As Type, _  
    existingValue As Object, _  
    serializer As JsonSerializer _  
) As Object
```

```
public:  
virtual Object^ ReadJson(  
    JsonReader^ reader,  
    Type^ objectType,  
    Object^ existingValue,  
    JsonSerializer^ serializer  
) abstract
```

### ▣ Parameters

**reader** ([JsonReader](#))

The [JsonReader](#) to read from.

***objectType*** ([Type](#))

Type of the object.

***existingValue*** ([Object](#))

The existing value of object being read.

***serializer*** ([JsonSerializer](#))

The calling serializer.

#### ▣ **Return Value**

The object value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConverter](#) ►

### WriteJson(JsonWriter, Object, JsonSerializer)

C#

Writes the JSON representation of the object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract void WriteJson(  
    JsonWriter writer,  
    Object value,  
    JsonSerializer serializer  
)
```

```
Public MustOverride Sub WriteJson ( _  
    writer As JsonWriter, _  
    value As Object, _  
    serializer As JsonSerializer _  
)
```

```
public:  
virtual void WriteJson(  
    JsonWriter^ writer,  
    Object^ value,  
    JsonSerializer^ serializer  
) abstract
```

#### ▣ Parameters

##### **writer** ([JsonWriter](#))

The [JsonWriter](#) to write to.

##### **value** ([Object](#))

The value.

***serializer* ([JsonSerializer](#))**

The calling serializer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonConverterAttribute**

Instructs the [JsonSerializer](#) to use the specified [JsonConverter](#) when serializing the member or class.

### Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public sealed class JsonConverterAttribute : Attribute
```

```
Public NotInheritable Class JsonConverterAttribute _
    Inherits Attribute
```

```
public ref class JsonConverterAttribute sealed : public
```

### Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
			<input checked="" type="checkbox"/> Declared
			<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonConverterAttribute(Type)</a>	Initializes a new instance of the <b>JsonConverterAttribute</b> class.
	<a href="#">ConverterType</a>	Gets the type of the converter.
	<a href="#">Equals(Object)</a>	Returns a value that indicates whether this instance is equal to a

		specified object.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Returns the hash code for this instance.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">IsDefaultAttribute()</a>	When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">Match(Object)</a>	When overridden in a derived class, returns a value that indicate whether this instance equals a specified object.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents

		the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">TypeId</a>	When implemented in a derived class, gets a unique identifier for this <a href="#">Attribute</a> .  (Inherited from <a href="#">Attribute</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

└─ [Attribute](#)

└─ **JsonConverterAttribute**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)



# JsonConverterAttribute Constructor (converterType)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConverterAttribute](#) ►

## JsonConverterAttribute(Type)

Initializes a new instance of the [JsonConverterAttribute](#) class.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonConverterAttribute(  
    Type converterType  
)
```

```
Public Sub New ( _  
    converterType As Type _  
)
```

```
public:  
JsonConverterAttribute(  
    Type^ converterType  
)
```

### Parameters

#### **converterType** ([Type](#))

Type of the converter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConverterAttribute](#) ►

## ConverterType

Gets the type of the converter.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Type ConverterType { get; }
```

```
Public ReadOnly Property ConverterType As Type  
    Get
```

```
public:  
property Type^ ConverterType {  
    Type^ get ();  
}
```

### ▣ Value

The type of the converter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonConverterCollection**

C# ▼

Represents a collection of [JsonConverter](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonConverterCollection : Collection<Js
```

```
Public Class JsonConverterCollection _  
    Inherits Collection(Of JsonConverter)
```

```
public ref class JsonConverterCollection : public Co
```

### Members

All Members

Constructors

Methods

Properties

Public  
 Protected

Instance  
 Static

Declared  
 Inherited

Icon	Member	Description
	<a href="#">JsonConverterCollection()</a>	Initializes a new instance of the <b>JsonConverterCollection</b> class
	<a href="#">Add(T)</a>	Adds an object to the end of the <a href="#">Collection&lt;T&gt;</a> .  (Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a> .)
	<a href="#">Clear()</a>	Removes all elements from the

		<a href="#">Collection&lt;T&gt;</a> . (Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a> .)
	<a href="#">ClearItems()</a>	Removes all elements from the <a href="#">Collection&lt;T&gt;</a> . (Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a> .)
	<a href="#">Contains(T)</a>	Determines whether an element is in the <a href="#">Collection&lt;T&gt;</a> . (Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a> .)
	<a href="#">CopyTo(T[], Int32)</a>	Copies the entire <a href="#">Collection&lt;T&gt;</a> to a compatible one-dimensional <a href="#">Array</a> , starting at the specified index of the target array. (Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a> .)
	<a href="#">Count</a>	Gets the number of elements actually contained in the <a href="#">Collection&lt;T&gt;</a> . (Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.

		(Inherited from <a href="#">Object</a> .)
	<a href="#">GetEnumerator()</a>	Returns an enumerator that iterates through the <a href="#">Collection&lt;T&gt;</a> .  (Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">IndexOf(T)</a>	Searches for the specified object and returns the zero-based index of the first occurrence within the entire <a href="#">Collection&lt;T&gt;</a> .  (Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a> .)
	<a href="#">Insert(Int32, T)</a>	Inserts an element into the <a href="#">Collection&lt;T&gt;</a> at the specified index.  (Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a> .)
	<a href="#">InsertItem(Int32, T)</a>	Inserts an element into the <a href="#">Collection&lt;T&gt;</a> at the specified index.  (Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a> .)
	<a href="#">Item[Int32]</a>	Gets or sets the element at the specified index.  (Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a> .)

	<a href="#">Items</a>	<p>Gets a <a href="#">IList&lt;T&gt;</a> wrapper around the <a href="#">Collection&lt;T&gt;</a>.</p> <p>(Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a>.)</p>
	<a href="#">MemberwiseClone()</a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">Remove(T)</a>	<p>Removes the first occurrence of a specific object from the <a href="#">Collection&lt;T&gt;</a>.</p> <p>(Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a>.)</p>
	<a href="#">RemoveAt(Int32)</a>	<p>Removes the element at the specified index of the <a href="#">Collection&lt;T&gt;</a>.</p> <p>(Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a>.)</p>
	<a href="#">RemoveItem(Int32)</a>	<p>Removes the element at the specified index of the <a href="#">Collection&lt;T&gt;</a>.</p> <p>(Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a>.)</p>
	<a href="#">SetItem(Int32, T)</a>	<p>Replaces the element at the specific index.</p> <p>(Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a>.)</p>
	<a href="#">ToString()</a>	<p>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>

## ▣ Inheritance Hierarchy

[Object](#)

└─ [Collection](#)<[JsonConverter](#)>

└─ **JsonConverterCollection**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)



# JsonConverterCollection Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConverterCollection](#) ►

## JsonConverterCollection()

Initializes a new instance of the [JsonConverterCollection](#) class

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonConverterCollection()
```

```
Public Sub New
```

```
public:  
JsonConverterCollection()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonDictionaryAttribute**

C#

Instructs the [JsonSerializer](#) how to serialize the collection.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public sealed class JsonDictionaryAttribute : JsonCo
```

```
Public NotInheritable Class JsonDictionaryAttribute
    Inherits JsonContainerAttribute
```

```
public ref class JsonDictionaryAttribute sealed : pu
```

### Members

All Members

Constructors

Methods

Properties

Public  
 Protected

Instance  
 Static

Declare  
 Inherited

Icon	Member	Description
	<a href="#">JsonDictionaryAttribute()</a>	Initializes a new instance of the <b>JsonDictionaryAttribute</b> class.
	<a href="#">JsonDictionaryAttribute(String)</a>	Initializes a new instance of the <b>JsonDictionaryAttribute</b> class with the specified container Id.
	<a href="#">Description</a>	Gets or sets the description.

		(Inherited from <a href="#">JsonContainerAttribute.</a> )
	<a href="#">Equals(Object)</a>	Returns a value that indicates whether this instance is equal to specified object.  (Inherited from <a href="#">Attribute.</a> )
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object.</a> )
	<a href="#">GetHashCode()</a>	Returns the hash code for this instance.  (Inherited from <a href="#">Attribute.</a> )
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object.</a> )
	<a href="#">Id</a>	Gets or sets the id.  (Inherited from <a href="#">JsonContainerAttribute.</a> )
	<a href="#">IsDefaultAttribute()</a>	When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.  (Inherited from <a href="#">Attribute.</a> )
	<a href="#">IsReference</a>	Gets or sets a value that indicate whether to preserve object references.

		(Inherited from <a href="#">JsonContainerAttribute.</a> )
	<a href="#">ItemConverterType</a>	Gets the collection's items converter.  (Inherited from <a href="#">JsonContainerAttribute.</a> )
	<a href="#">ItemsReference</a>	Gets or sets a value that indicate whether to preserve collection's items references.  (Inherited from <a href="#">JsonContainerAttribute.</a> )
	<a href="#">ItemReferenceLoopHandling</a>	Gets or sets the reference loop handling used when serializing the collection's items.  (Inherited from <a href="#">JsonContainerAttribute.</a> )
	<a href="#">ItemTypeNameHandling</a>	Gets or sets the type name handling used when serializing the collection's items.  (Inherited from <a href="#">JsonContainerAttribute.</a> )
	<a href="#">Match(Object)</a>	When overridden in a derived class, returns a value that indicates whether this instance equals a specified object.  (Inherited from <a href="#">Attribute.</a> )
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .

		(Inherited from <a href="#">Object</a> .)
	<a href="#">Title</a>	Gets or sets the title.  (Inherited from <a href="#">JsonContainerAttribute</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">TypeId</a>	When implemented in a derived class, gets a unique identifier for this <a href="#">Attribute</a> .  (Inherited from <a href="#">Attribute</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

└─ [Attribute](#)

└─ [JsonContainerAttribute](#)

└─ **JsonDictionaryAttribute**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonDictionaryAttribute](#) ►

**JsonDictionaryAttribute()**

### Members

Icon	Member	Description
	<a href="#">JsonDictionaryAttribute()</a>	Initializes a new instance of the <a href="#">JsonDictionaryAttribute</a> class.
	<a href="#">JsonDictionaryAttribute(String)</a>	Initializes a new instance of the <a href="#">JsonDictionaryAttribute</a> class with the specified container Id.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonDictionaryAttribute Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonDictionaryAttribute](#) ►

## JsonDictionaryAttribute()

Initializes a new instance of the [JsonDictionaryAttribute](#) class.

class.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonDictionaryAttribute()
```

```
Public Sub New
```

```
public:  
JsonDictionaryAttribute()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# JsonDictionaryAttribute Constructor (id)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonDictionaryAttribute](#) ►

## JsonDictionaryAttribute(String)

Initializes a new instance of the [JsonDictionaryAttribute](#) class with the specified container Id.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonDictionaryAttribute(  
    string id  
)
```

```
Public Sub New ( _  
    id As String _  
)
```

```
public:  
JsonDictionaryAttribute(  
    String^ id  
)
```

### Parameters

***id*** ([String](#))

The container Id.

# JsonException Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonException**

The exception thrown when an error occurs during Json serialization or deserialization.

## Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
[SerializableAttribute]
public class JsonException : Exception
```

```
<SerializableAttribute> _
Public Class JsonException _
    Inherits Exception
```

```
[SerializableAttribute]
public ref class JsonException : public Exception
```

## Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
			<input checked="" type="checkbox"/> Declare
			<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonException()</a>	Initializes a new instance of the <b>JsonException</b> class.
	<a href="#">JsonException(String)</a>	Initializes a new instance of the <b>JsonException</b> class with a specified error message.

	<a href="#">JsonException(String, Exception)</a>	Initializes a new instance of the <b>JsonException</b> class with a specified error message and a reference to the inner exception that is the cause of this exception.
	<a href="#">JsonException(SerializationInfo, StreamingContext)</a>	Initializes a new instance of the <b>JsonException</b> class.
	<a href="#">Data</a>	Gets a collection of key/value pairs that provide additional user-defined information about the exception.  (Inherited from <a href="#">Exception</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetBaseException()</a>	When overridden in a derived class, returns the <a href="#">Exception</a> that is the root cause of one or more subsequent exceptions.

		(Inherited from <a href="#">Exception</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetObjectData(SerializationInfo, StreamingContext)</a>	When overridden in a derived class, sets the <a href="#">SerializationInfo</a> with information about the exception.  (Inherited from <a href="#">Exception</a> .)
	<a href="#">GetType()</a>	Gets the runtime type of the current instance.  (Inherited from <a href="#">Exception</a> .)
	<a href="#">HelpLink</a>	Gets or sets a link to the help file associated with this exception.  (Inherited from <a href="#">Exception</a> .)
	<a href="#">HResult</a>	Gets or sets HRESULT, a code numerical value that is assigned to a specific exception.  (Inherited from <a href="#">Exception</a> .)
	<a href="#">InnerException</a>	Gets the <a href="#">Exception</a> instance that caused the current exception.  (Inherited from <a href="#">Exception</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Message</a>	Gets a message that describes

		the current exception.  (Inherited from <a href="#">Exception</a> .)
	<a href="#">Source</a>	Gets or sets the name of the application or the object that causes the error.  (Inherited from <a href="#">Exception</a> .)
	<a href="#">StackTrace</a>	Gets a string representation of the frames on the call stack at the time the current exception was thrown.  (Inherited from <a href="#">Exception</a> .)
	<a href="#">TargetSite</a>	Gets the method that throws the current exception.  (Inherited from <a href="#">Exception</a> .)
	<a href="#">ToString()</a>	Creates and returns a string representation of the current exception.  (Inherited from <a href="#">Exception</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

└─ [Exception](#)

└─ **JsonException**

└─ [JsonReaderException](#)

└─ [JsonSerializationException](#)

└─ [JsonWriterException](#)

└─ [JsonSchemaException](#)

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonException](#) ►

### JsonException()

#### Members

Icon	Member	Description
	<a href="#">JsonException()</a>	Initializes a new instance of the <a href="#">JsonException</a> class.
	<a href="#">JsonException(String)</a>	Initializes a new instance of the <a href="#">JsonException</a> class with a specified error message.
	<a href="#">JsonException(String, Exception)</a>	Initializes a new instance of the <a href="#">JsonException</a> class with a specified error message and a reference to the inner exception that is the cause of this exception.
	<a href="#">JsonException(SerializationInfo, StreamingContext)</a>	Initializes a new instance of the <a href="#">JsonException</a> class.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonException](#) ►

### JsonException()

Initializes a new instance of the [JsonException](#) class.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonException()
```

```
Public Sub New
```

```
public:  
JsonException()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# JsonException

## Constructor (info, context)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonException](#) ►

### JsonException(SerializationInfo, StreamingContext)

Initializes a new instance of the [JsonException](#) class.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonException(  
    SerializationInfo info,  
    StreamingContext context  
)
```

```
Public Sub New ( _  
    info As SerializationInfo, _  
    context As StreamingContext _  
)
```

```
public:  
JsonException(  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

#### ▣ Parameters

##### **info** ([SerializationInfo](#))

The [SerializationInfo](#) that holds the serialized object data about the exception being thrown.

##### **context** ([StreamingContext](#))

The [StreamingContext](#) that contains contextual information about the source or destination.

#### ▣ Exceptions

Exception	Condition
<a href="#">ArgumentNullException</a>	The <i>info</i> parameter is null.
<a href="#">SerializationException</a>	The class name is null or <a href="#">HResult</a> is zero (0).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor (message)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonException](#) ►

### JsonException(String)

Initializes a new instance of the [JsonException](#) class with a specified error message.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonException(     string message )</pre>	<pre>Public Sub New ( _     message As String _ )</pre>	<pre>public: JsonException(     String^ message )</pre>

#### ▣ Parameters

##### **message** ([String](#))

The error message that explains the reason for the exception.

# JsonException

## Constructor (message, innerException)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonException](#) ►

### JsonException(String, Exception)

Initializes a new instance of the [JsonException](#) class with a specified error message and a reference to the inner exception that is the cause of this exception.

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonException(     string message,     Exception innerException )</pre>	<pre>Public Sub New ( _     message As String, _     innerException As Exception _ )</pre>	<pre>public: JsonException(     String^ message,     Exception^ innerException )</pre>

#### Parameters

##### **message** ([String](#))

The error message that explains the reason for the exception.

##### **innerException** ([Exception](#))

The exception that is the cause of the current exception, or a null reference (Nothing in Visual Basic) if no inner exception is specified.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

# JsonIgnoreAttribute

## Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonIgnoreAttribute**

Instructs the [JsonSerializer](#) not to serialize the public field or public read/write property value.

### Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public sealed class JsonIgnoreAttribute : Attribute
```

```
Public NotInheritable Class JsonIgnoreAttribute _
    Inherits Attribute
```

```
public ref class JsonIgnoreAttribute sealed : public
```

### Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonIgnoreAttribute()</a>	Initializes a new instance of the <b>JsonIgnoreAttribute</b> class
	<a href="#">Equals(Object)</a>	Returns a value that indicates whether this instance is equal to a specified object.  (Inherited from <a href="#">Attribute</a> .)

	<a href="#">Finalize()</a>	<p>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetHashCode()</a>	<p>Returns the hash code for this instance.</p> <p>(Inherited from <a href="#">Attribute</a>.)</p>
	<a href="#">GetType()</a>	<p>Gets the <a href="#">Type</a> of the current instance</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">IsDefaultAttribute()</a>	<p>When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.</p> <p>(Inherited from <a href="#">Attribute</a>.)</p>
	<a href="#">Match(Object)</a>	<p>When overridden in a derived class, returns a value that indicates whether this instance equals a specified object</p> <p>(Inherited from <a href="#">Attribute</a>.)</p>
	<a href="#">MemberwiseClone()</a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">ToString()</a>	<p>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">TypeId</a>	<p>When implemented in a derived class gets a unique identifier for this <a href="#">Attribute</a>.</p>

(Inherited from [Attribute](#).)

## ▣ Inheritance Hierarchy

[Object](#)

└─ [Attribute](#)

└─ **JsonIgnoreAttribute**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonIgnoreAttribute](#) ►

### JsonIgnoreAttribute()

Initializes a new instance of the [JsonIgnoreAttribute](#) class

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonIgnoreAttribute()</pre>		
	<pre>Public Sub New</pre>	
		<pre>public: JsonIgnoreAttribute()</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonObjectAttribute

## Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonObjectAttribute**

Instructs the [JsonSerializer](#) how to serialize the object.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public sealed class JsonObjectAttribute : JsonContainerAttribute
```

```
Public NotInheritable Class JsonObjectAttribute _  
    Inherits JsonContainerAttribute
```

```
public ref class JsonObjectAttribute sealed : public
```

### Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declare <input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonObjectAttribute()</a>	Initializes a new instance of the <b>JsonObjectAttribute</b> class.
	<a href="#">JsonObjectAttribute(MemberSerialization)</a>	Initializes a new instance of the <b>JsonObjectAttribute</b> class with the specified member serialization.

	<a href="#">JsonObjectAttribute(String)</a>	Initializes a new instance of the <b>JsonObjectAttribute</b> class with the specific container Id.
	<a href="#">Description</a>	Gets or sets the description.  (Inherited from <a href="#">JsonContainerAttribut</a>
	<a href="#">Equals(Object)</a>	Returns a value that indicates whether this instance is equal to a specified object.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a>
	<a href="#">GetHashCode()</a>	Returns the hash code for this instance.  (Inherited from <a href="#">Attribute</a> .)
		

	<a href="#">GetType()</a>	<p>Gets the <a href="#">Type</a> of the current instance.</p> <p>(Inherited from <a href="#">Object</a>)</p>
	<a href="#">Id</a>	<p>Gets or sets the id.</p> <p>(Inherited from <a href="#">JsonContainerAttribute</a>)</p>
	<a href="#">IsDefaultAttribute()</a>	<p>When overridden in a derived class, indicate whether the value of this instance is the default value for the derived class.</p> <p>(Inherited from <a href="#">Attribute</a>.)</p>
	<a href="#">IsReference</a>	<p>Gets or sets a value that indicates whether to preserve object references.</p> <p>(Inherited from <a href="#">JsonContainerAttribute</a>)</p>
	<a href="#">ItemConverterType</a>	<p>Gets the collection's items converter.</p> <p>(Inherited from <a href="#">JsonContainerAttribute</a>)</p>
	<a href="#">ItemIsReference</a>	<p>Gets or sets a value that indicates whether to preserve collection's items references.</p> <p>(Inherited from <a href="#">JsonContainerAttribute</a>)</p>

	<a href="#">ItemReferenceLoopHandling</a>	<p>Gets or sets the reference loop handling used when serializing the collection's items.</p> <p>(Inherited from <a href="#">JsonContainerAttribute</a>)</p>
	<a href="#">ItemRequired</a>	<p>Gets or sets a value that indicates whether the object's properties are required.</p>
	<a href="#">ItemTypeNameHandling</a>	<p>Gets or sets the type name handling used when serializing the collection's items.</p> <p>(Inherited from <a href="#">JsonContainerAttribute</a>)</p>
	<a href="#">Match(Object)</a>	<p>When overridden in a derived class, returns value that indicates whether this instance equals a specified object.</p> <p>(Inherited from <a href="#">Attribute</a>.)</p>
	<a href="#">MemberSerialization</a>	<p>Gets or sets the member serialization.</p>
	<a href="#">MemberwiseClone()</a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p>

		(Inherited from <a href="#">Object</a> )
	<a href="#">Title</a>	Gets or sets the title.  (Inherited from <a href="#">JsonContainerAttribut</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> )
	<a href="#">TypeId</a>	When implemented in derived class, gets a unique identifier for th <a href="#">Attribute</a> .  (Inherited from <a href="#">Attribute</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

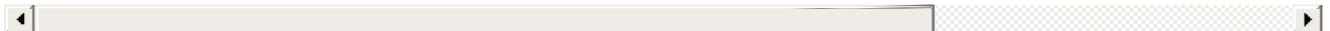
└─ [Attribute](#)

└─ [JsonContainerAttribute](#)

└─ **JsonObjectAttribute**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# JsonObjectAttribute

## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonObjectAttribute](#) ►

### JsonObjectAttribute()

#### Members

Icon	Member	Description
	<a href="#">JsonObjectAttribute()</a>	Initializes a new instance of the <a href="#">JsonObjectAttribute</a> class.
	<a href="#">JsonObjectAttribute(MemberSerialization)</a>	Initializes a new instance of the <a href="#">JsonObjectAttribute</a> class with the specific member serialization.
	<a href="#">JsonObjectAttribute(String)</a>	Initializes a new instance of the <a href="#">JsonObjectAttribute</a> class with the specific container Id.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonObjectAttribute](#) ►

### JsonObjectAttribute()

Initializes a new instance of the [JsonObjectAttribute](#) class.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonObjectAttribute()</pre>		
	<pre>Public Sub New</pre>	
		<pre>public: JsonObjectAttribute()</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonObjectAttribute Constructor (memberSerialization)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonObjectAttribute](#) ►

## JsonObjectAttribute(MemberSerialization)

Initializes a new instance of the [JsonObjectAttribute](#) class with the specified member serialization.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonObjectAttribute(  
    MemberSerialization memberSerialization  
)
```

```
Public Sub New ( _  
    memberSerialization As MemberSerialization _  
)
```

```
public:  
JsonObjectAttribute(  
    MemberSerialization memberSerialization  
)
```

### Parameters

***memberSerialization*** ([MemberSerialization](#))

The member serialization.

## Constructor (id)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonObjectAttribute](#) ►

### JsonObjectAttribute(String)

Initializes a new instance of the [JsonObjectAttribute](#) class with the specified container Id.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonObjectAttribute(     string id )</pre>		
	<pre>Public Sub New ( _     id As String _ )</pre>	
		<pre>public: JsonObjectAttribute(     String^ id )</pre>

#### ▣ Parameters

***id*** ([String](#))

The container Id.

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonObjectAttribute](#) ►

## ItemRequired

Gets or sets a value that indicates whether the object's properties are required.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Required ItemRequired { get; set; }
```

```
Public Property ItemRequired As Required  
    Get  
    Set
```

```
public:  
property Required ItemRequired {  
    Required get ();  
    void set (Required value);  
}
```

### Value

A value indicating whether the object's properties are required.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**Property**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonObjectAttribute](#) ►**MemberSerialization**

Gets or sets the member serialization.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public MemberSerialization MemberSerialization { get
```

```
    Public Property MemberSerialization As MemberSeriali  
        Get  
        Set
```

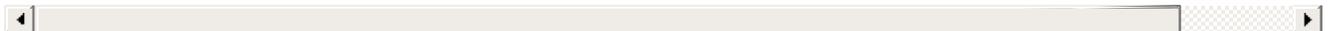
```
public:  
property MemberSerialization MemberSerialization {  
    MemberSerialization get ();  
    void set (MemberSerialization value);  
}
```

▣ **Value**

The member serialization.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# JsonPropertyAttribute

## Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonPropertyAttribute**

Instructs the [JsonSerializer](#) to always serialize the member with the specified name.

### Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public sealed class JsonPropertyAttribute : Attribute
```

```
Public NotInheritable Class JsonPropertyAttribute _
    Inherits Attribute
```

```
public ref class JsonPropertyAttribute sealed : publ
```

### Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
		<input checked="" type="checkbox"/> Declared	
		<input checked="" type="checkbox"/> Inherited	

Icon	Member	Description
	<a href="#">JsonPropertyAttribute()</a>	Initializes a new instance of the <b>JsonPropertyAttribute</b> class.
	<a href="#">JsonPropertyAttribute(String)</a>	Initializes a new instance of the <b>JsonPropertyAttribute</b> class with the specified name.

	<a href="#">DefaultValueHandling</a>	Gets or sets the default value handling used when serializing this property.
	<a href="#">Equals(Object)</a>	Returns a value that indicates whether this instance is equal to a specified object.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Returns the hash code for this instance.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">IsDefaultAttribute()</a>	When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">IsReference</a>	Gets or sets whether this property's value is serialized as a reference.

	<a href="#">ItemConverterType</a>	Gets or sets the converter used when serializing the property's collection items.
	<a href="#">ItemsReference</a>	Gets or sets whether this property's collection items are serialized as a reference.
	<a href="#">ItemReferenceLoopHandling</a>	Gets or sets the the reference loop handling used when serializing the property's collection items.
	<a href="#">ItemTypeNameHandling</a>	Gets or sets the the type name handling used when serializing the property's collection items.
	<a href="#">Match(Object)</a>	When overridden in a derived class, returns a value that indicates whether this instance equals a specified object.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">NullValueHandling</a>	Gets or sets the null value handling used when serializing this property
	<a href="#">ObjectCreationHandling</a>	Gets or sets the object creation

		handling used when deserializing this property.
	<a href="#">Order</a>	Gets or sets the order of serialization and deserialization of member.
	<a href="#">PropertyName</a>	Gets or sets the name of the property.
	<a href="#">ReferenceLoopHandling</a>	Gets or sets the reference loop handling used when serializing this property.
	<a href="#">Required</a>	Gets or sets a value indicating whether this property is required.
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">TypeId</a>	When implemented in a derived class, gets a unique identifier for this <a href="#">Attribute</a> .  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">TypeNameHandling</a>	Gets or sets the type name handling used when serializing this property.

## ▣ Inheritance Hierarchy

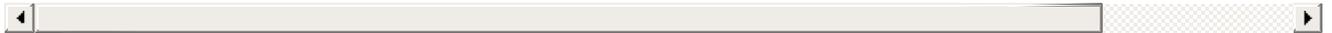
[Object](#)

└─ [Attribute](#)

└─ **JsonPropertyAttribute**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonPropertyAttribute](#) ►

### JsonPropertyAttribute()

#### Members

Icon	Member	Description
	<a href="#">JsonPropertyAttribute()</a>	Initializes a new instance of the <a href="#">JsonPropertyAttribute</a> class.
	<a href="#">JsonPropertyAttribute(String)</a>	Initializes a new instance of the <a href="#">JsonPropertyAttribute</a> class with the specified name.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonPropertyAttribute](#) ►

### JsonPropertyAttribute()

Initializes a new instance of the [JsonPropertyAttribute](#) class.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonPropertyAttribute()</pre>		
	<pre>Public Sub New</pre>	
		<pre>public: JsonPropertyAttribute()</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonPropertyAttribute Constructor (propertyName)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonPropertyAttribute](#) ►

## JsonPropertyAttribute(String)

Initializes a new instance of the [JsonPropertyAttribute](#) class with the specified name.

### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonPropertyAttribute(     string propertyName )</pre>	<pre>Public Sub New ( _     propertyName As String _ )</pre>	<pre>public: JsonPropertyAttribute(     String^ propertyName )</pre>

### Parameters

**propertyName** ([String](#))

Name of the property.

## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonPropertyAttribute](#) ►

### DefaultValueHandling

Gets or sets the default value handling used when serializing this property.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public DefaultValueHandling DefaultValueHandling { g
```

```
Public Property DefaultValueHandling As DefaultValue  
    Get  
    Set
```

```
public:  
property DefaultValueHandling DefaultValueHandling {  
    DefaultValueHandling get ();  
    void set (DefaultValueHandling value);  
}
```

#### Value

The default value handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonPropertyAttribute](#) ►

## IsReference

C#

Gets or sets whether this property's value is serialized as a reference.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool IsReference { get; set; }
```

```
Public Property IsReference As Boolean  
    Get  
    Set
```

```
public:  
property bool IsReference {  
    bool get ();  
    void set (bool value);  
}
```

### ▣ Value

Whether this property's value is serialized as a reference.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonPropertyAttribute](#) ►

### ItemConverterType

Gets or sets the converter used when serializing the property's collection items.

C#

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Type ItemConverterType { get; set; }
```

```
Public Property ItemConverterType As Type
    Get
    Set
```

```
public:
property Type^ ItemConverterType {
    Type^ get ();
    void set (Type^ value);
}
```

#### Value

The collection's items converter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonPropertyAttribute](#) ►

## ItemIsReference

Gets or sets whether this property's collection items are serialized as a reference.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool ItemIsReference { get; set; }
```

```
Public Property ItemIsReference As Boolean  
    Get  
    Set
```

```
public:  
property bool ItemIsReference {  
    bool get ();  
    void set (bool value);  
}
```

### Value

Whether this property's collection items are serialized as a reference.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**ItemReferenceLoopHandling**

C#

Gets or sets the the reference loop handling used when serializing the property's collection items.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public ReferenceLoopHandling ItemReferenceLoopHandli
```

```
Public Property ItemReferenceLoopHandling As Referen  
    Get  
    Set
```

```
public:  
property ReferenceLoopHandling ItemReferenceLoopHand  
    ReferenceLoopHandling get ();  
    void set (ReferenceLoopHandling value);  
}
```

▣ **Value**

The collection's items reference loop handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# ItemTypeNameHandling Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonPropertyAttribute](#) ►

## ItemTypeNameHandling

Gets or sets the the type name handling used when serializing the property's collection items.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public TypeNameHandling ItemTypeNameHandling { get;
```

```
Public Property ItemTypeNameHandling As TypeNameHand  
    Get  
    Set
```

```
public:  
property TypeNameHandling ItemTypeNameHandling {  
    TypeNameHandling get ();  
    void set (TypeNameHandling value);  
}
```

### Value

The collection's items type name handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonPropertyAttribute](#) ►

### NullValueHandling

Gets or sets the null value handling used when serializing this property. C# ▼

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public NullValueHandling NullValueHandling { get; se
```

```
Public Property NullValueHandling As NullValueHandli  
    Get  
    Set
```

```
public:  
property NullValueHandling NullValueHandling {  
    NullValueHandling get ();  
    void set (NullValueHandling value);  
}
```

#### ▣ Value

The null value handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



**Property**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonPropertyAttribute](#) ►**ObjectCreationHandling**

Gets or sets the object creation handling used when deserializing this property.

C#

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public ObjectCreationHandling ObjectCreationHandling
```

```
Public Property ObjectCreationHandling As ObjectCrea  
    Get  
    Set
```

```
public:  
property ObjectCreationHandling ObjectCreationHandli  
    ObjectCreationHandling get ();  
    void set (ObjectCreationHandling value);  
}
```

▣ **Value**

The object creation handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



C#

Gets or sets the order of serialization and deserialization of a member.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public int Order { get; set; }
```

```
Public Property Order As Integer  
    Get  
    Set
```

```
public:  
property int Order {  
    int get ();  
    void set (int value);  
}
```

## Value

The numeric order of serialization or deserialization.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonPropertyAttribute](#) ►**PropertyName**

Gets or sets the name of the property.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public string PropertyName { get; set; }
```

```
Public Property PropertyName As String  
    Get  
    Set
```

```
public:  
property String^ PropertyName {  
    String^ get ();  
    void set (String^ value);  
}
```

▣ **Value**

The name of the property.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonPropertyAttribute](#) ►

### ReferenceLoopHandling

Gets or sets the reference loop handling used when serializing this property.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public ReferenceLoopHandling ReferenceLoopHandling {
```

```
    Public Property ReferenceLoopHandling As ReferenceLo  
        Get  
        Set
```

```
public:  
property ReferenceLoopHandling ReferenceLoopHandling  
    ReferenceLoopHandling get ();  
    void set (ReferenceLoopHandling value);  
}
```

#### Value

The reference loop handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonPropertyAttribute](#) ► **Required**

C#

Gets or sets a value indicating whether this property is required.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Required Required { get; set; }
```

```
Public Property Required As Required  
    Get  
    Set
```

```
public:  
property Required Required {  
    Required get ();  
    void set (Required value);  
}
```

## ▣ Value

A value indicating whether this property is required.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonPropertyAttribute](#) ►

### TypeNameHandling

C#

Gets or sets the type name handling used when serializing this property

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public TypeNameHandling TypeNameHandling { get; set;
```

```
Public Property TypeNameHandling As TypeNameHandling  
    Get  
    Set
```

```
public:  
property TypeNameHandling TypeNameHandling {  
    TypeNameHandling get ();  
    void set (TypeNameHandling value);  
}
```

#### Value

The type name handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonReader Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonReader**

C#

Represents a reader that provides fast, non-cached, forward-only access to serialized Json data.

## Declaration Syntax

C#      Visual Basic      Visual C++

```
public abstract class JsonReader : IDisposable
```

```
Public MustInherit Class JsonReader _  
    Implements IDisposable
```

```
public ref class JsonReader abstract : IDisposable
```

## Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declare <input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonReader()</a>	Initializes a new instance of the <b>JsonReader</b> class with the specified <a href="#">TextReader</a> .
	<a href="#">Close()</a>	Changes the <a href="#">JsonReader.State</a> to Closed.
	<a href="#">CloseInput</a>	Gets or sets a value indicating whether the underlying stream or

		<a href="#">TextReader</a> should be closed when the reader is closed.
	<a href="#">Culture</a>	Gets or sets the culture used when reading JSON. Defaults to <a href="#">InvariantCulture</a> .
	<a href="#">CurrentState</a>	Gets the current reader state.
	<a href="#">DateParseHandling</a>	Get or set how date formatted strings, e.g. "VDate(1198908717056)V" and "2012-03-21T05:40Z", are parsed when reading JSON.
	<a href="#">DateTimeZoneHandling</a>	Get or set how <a href="#">DateTime</a> time zones are handling when reading JSON.
	<a href="#">Depth</a>	Gets the depth of the current token in the JSON document.
	<a href="#">Dispose(Boolean)</a>	Releases unmanaged and - optionally - managed resources
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>  (Inherited from <a href="#">Object</a> .)

	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MaxDepth</a>	Gets or sets the maximum depth allowed when reading JSON. Reading past this depth will throw a <a href="#">JsonReaderException</a> .
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Path</a>	Gets the path of the current JSON token.
	<a href="#">QuoteChar</a>	Gets the quotation mark character used to enclose the value of a string
	<a href="#">Read()</a>	Reads the next JSON token from the stream.

	<a href="#">ReadAsBytes()</a>	Reads the next JSON token from the stream as a <b>Byte[]</b> .
	<a href="#">ReadAsDateTime()</a>	Reads the next JSON token from the stream as a <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">ReadAsDateTimeOffset()</a>	Reads the next JSON token from the stream as a <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">ReadAsDecimal()</a>	Reads the next JSON token from the stream as a <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">ReadAsInt32()</a>	Reads the next JSON token from the stream as a <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">ReadAsString()</a>	Reads the next JSON token from the stream as a <a href="#">String</a> .
	<a href="#">SetStateBasedOnCurrent()</a>	Sets the state based on current token type.
	<a href="#">SetToken(JsonToken)</a>	Sets the current token.
	<a href="#">SetToken(JsonToken, Object)</a>	Sets the current token and value.

	<a href="#">Skip()</a>	Skips the children of the current token.
	<a href="#">TokenType</a>	Gets the type of the current JSON token.
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Value</a>	Gets the text value of the current JSON token.
	<a href="#">ValueType</a>	Gets The Common Language Runtime (CLR) type for the current JSON token.

## ▣ Inheritance Hierarchy

### [Object](#)

- └─ **JsonReader**
  - └─ [BsonReader](#)
  - └─ [JsonTextReader](#)
  - └─ [JsonValidatingReader](#)
  - └─ [JTokenReader](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **JsonReader()**

Initializes a new instance of the [JsonReader](#) class with the specified [TextReader](#).

C#

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected JsonReader ( )
```

```
Protected Sub New
```

```
protected:  
JsonReader ( )
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **Close()**

Changes the [JsonReader.State](#) to Closed.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void Close()
```

```
Public Overridable Sub Close
```

```
public:  
virtual void Close()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **CloseInput**

C#

Gets or sets a value indicating whether the underlying stream or [TextReader](#) should be closed when the reader is closed.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool CloseInput { get; set; }
```

```
Public Property CloseInput As Boolean  
    Get  
    Set
```

```
public:  
property bool CloseInput {  
    bool get ();  
    void set (bool value);  
}
```

## ▣ Value

true to close the underlying stream or [TextReader](#) when the reader is closed; otherwise false. The default is true.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **Culture**

Gets or sets the culture used when reading JSON. Defaults to [InvariantCulture](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public CultureInfo Culture { get; set; }
```

```
Public Property Culture As CultureInfo  
    Get  
    Set
```

```
public:  
property CultureInfo^ Culture {  
    CultureInfo^ get ();  
    void set (CultureInfo^ value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **CurrentState**

C#

Gets the current reader state.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
protected JsonReader.State CurrentState { get; }
```

```
Protected ReadOnly Property CurrentState As JsonReader.State  
Get
```

```
protected:  
property JsonReader.State CurrentState {  
    JsonReader.State get ();  
}
```

## Value

The current reader state.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ►

### DateParseHandling

Get or set how date formatted strings, e.g. "VDate(1198908717056)V" and "2012-03-21T05:40Z", are parsed when reading JSON.

#### Declaration Syntax

C#

Visual Basic

Visual C++

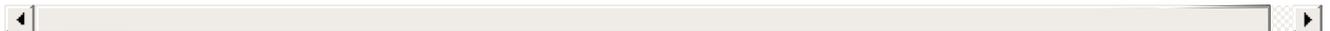
```
public DateParseHandling DateParseHandling { get; se
```

```
Public Property DateParseHandling As DateParseHandli  
    Get  
    Set
```

```
public:  
property DateParseHandling DateParseHandling {  
    DateParseHandling get ();  
    void set (DateParseHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ►

### DateTimeZoneHandling

Get or set how [DateTime](#) time zones are handling when reading JSON.

C#

#### Declaration Syntax

C#

Visual Basic

Visual C++

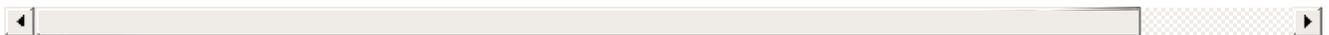
```
public DateTimeZoneHandling DateTimeZoneHandling { g
```

```
Public Property DateTimeZoneHandling As DateTimeZone  
    Get  
    Set
```

```
public:  
property DateTimeZoneHandling DateTimeZoneHandling {  
    DateTimeZoneHandling get ();  
    void set (DateTimeZoneHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **Depth**

Gets the depth of the current token in the JSON document.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual int Depth { get; }
```

```
Public Overridable ReadOnly Property Depth As Intege  
Get
```

```
public:  
virtual property int Depth {  
    int get ();  
}
```

## ▣ Value

The depth of the current token in the JSON document.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# Dispose Method (disposing)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **Dispose(Boolean)**

Releases unmanaged and - optionally - managed resources

C# ▼

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual void Dispose(  
    bool disposing  
)
```

```
Protected Overridable Sub Dispose ( _  
    disposing As Boolean _  
)
```

```
protected:  
virtual void Dispose(  
    bool disposing  
)
```

## ▣ Parameters

### *disposing* ([Boolean](#))

`true` to release both managed and unmanaged resources; `false` to release only unmanaged resources.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **MaxDepth**

Gets or sets the maximum depth allowed when reading JSON. Reading past this depth will throw a [JsonReaderException](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<int> MaxDepth { get; set; }
```

```
Public Property MaxDepth As Nullable(Of Integer)  
    Get  
    Set
```

```
public:  
property Nullable<int> MaxDepth {  
    Nullable<int> get ();  
    void set (Nullable<int> value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets the path of the current JSON token.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual string Path { get; }
```

```
Public Overridable ReadOnly Property Path As String  
    Get
```

```
public:  
virtual property String^ Path {  
    String^ get ();  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **QuoteChar**

Gets the quotation mark character used to enclose the value of a string.

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

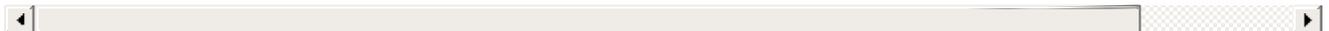
```
public virtual char QuoteChar { get; protected inter
```

```
Public Overridable Property QuoteChar As Char  
    Get  
    Protected Friend Set
```

```
public:  
virtual property wchar_t QuoteChar {  
    wchar_t get ();  
    protected public: void set (wchar_t value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **Read()**

Reads the next JSON token from the stream.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract bool Read()
```

```
Public MustOverride Function Read As Boolean
```

```
public:  
virtual bool Read() abstract
```

## ▣ Return Value

true if the next token was read successfully; false if there are no more tokens to read.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **ReadAsBytes()**

Reads the next JSON token from the stream as a **Byte[]**.

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract byte[] ReadAsBytes()
```

```
Public MustOverride Function ReadAsBytes As Byte()
```

```
public:  
virtual array<unsigned char>^ ReadAsBytes() abstract
```

## Return Value

A **Byte[]** or a null reference if the next JSON token is null. This method will return **null** at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ►

## ReadAsDateTime()

Reads the next JSON token from the stream as a [Nullable<T>](#).

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract Nullable<DateTime> ReadAsDateTime()
```

```
Public MustOverride Function ReadAsDateTime As Nulla
```

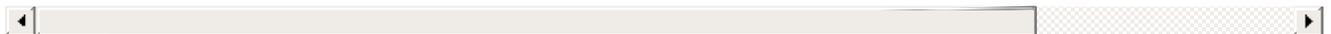
```
public:  
virtual Nullable<DateTime> ReadAsDateTime() abstract
```

### Return Value

A [String](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)



## Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ►

### ReadAsDateTimeOffset()

Reads the next JSON token from the stream as a [Nullable<T>](#).

C#

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract Nullable<DateTimeOffset> ReadAsDateT
```

```
Public MustOverride Function ReadAsDateTimeOffset As
```

```
public:  
virtual Nullable<DateTimeOffset> ReadAsDateTimeOffse
```

#### ▣ Return Value

A [Nullable<T>](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **ReadAsDecimal()**

Reads the next JSON token from the stream as a [Nullable<T>](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract Nullable<decimal> ReadAsDecimal()
```

```
Public MustOverride Function ReadAsDecimal As Nullab
```

```
public:  
virtual Nullable<Decimal> ReadAsDecimal() abstract
```

## ▣ Return Value

A [Nullable<T>](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **ReadAsInt32()**

Reads the next JSON token from the stream as a [Nullable<T>](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract Nullable<int> ReadAsInt32()
```

```
Public MustOverride Function ReadAsInt32 As Nullable
```

```
public:  
virtual Nullable<int> ReadAsInt32() abstract
```

## ▣ Return Value

A [Nullable<T>](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **ReadAsString()**

Reads the next JSON token from the stream as a [String](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract string ReadAsString()
```

```
Public MustOverride Function ReadAsString As String
```

```
public:  
virtual String^ ReadAsString() abstract
```

## ▣ Return Value

A [String](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# SetStateBasedOnCurrent Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ►

## SetStateBasedOnCurrent()

Sets the state based on current token type.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected void SetStateBasedOnCurrent()
```

```
Protected Sub SetStateBasedOnCurrent
```

```
protected:  
void SetStateBasedOnCurrent()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[-] **Members**

Icon	Member	Description
	<a href="#">SetToken(JsonToken)</a>	Sets the current token.
	<a href="#">SetToken(JsonToken, Object)</a>	Sets the current token and value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(newToken)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ►**SetToken(JsonToken)**

Sets the current token.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
protected void SetToken(  
    JsonToken newToken  
)
```

```
Protected Sub SetToken ( _  
    newToken As JsonToken _  
)
```

```
protected:  
void SetToken(  
    JsonToken newToken  
)
```

▣ **Parameters*****newToken* ([JsonToken](#))**

The new token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (newToken, value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ►

### SetToken(JsonToken, Object)

C#

Sets the current token and value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected void SetToken(  
    JsonToken newToken,  
    Object value  
)
```

```
Protected Sub SetToken ( _  
    newToken As JsonToken, _  
    value As Object _  
)
```

```
protected:  
void SetToken(  
    JsonToken newToken,  
    Object^ value  
)
```

#### ▣ Parameters

##### ***newToken*** ([JsonToken](#))

The new token.

##### ***value*** ([Object](#))

The value.

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **Skip()**

C#

Skips the children of the current token.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Skip()
```

```
Public Sub Skip
```

```
public:  
void Skip()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets the type of the current JSON token.

## Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public virtual <a href="#">JsonToken</a> TokenType { get; }</pre>		
	<pre>Public Overridable ReadOnly Property TokenType As <a href="#">Js</a> Get</pre>	
		<pre>public: virtual property <a href="#">JsonToken</a> TokenType {     <a href="#">JsonToken</a> get (); }</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



Gets the text value of the current JSON token.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public virtual <a href="#">Object</a> Value { get; }</pre>		
	<pre>Public Overridable ReadOnly Property Value As <a href="#">Object</a> Get</pre>	
		<pre>public: virtual property <a href="#">Object</a>^ Value {     <a href="#">Object</a>^ get (); }</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **ValueType**

C#

Gets The Common Language Runtime (CLR) type for the current JSON token.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual Type ValueType { get; }
```

```
Public Overridable ReadOnly Property ValueType As Type  
    Get
```

```
public:  
virtual property Type^ ValueType {  
    Type^ get ();  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Enumeration

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **JsonReader.State**

C# ▼

Specifies the state of the reader.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected internal enum State
```

```
Protected Friend Enumeration State
```

```
protected public enum class State
```

### Members

Member	Description
<b>Start</b>	The Read method has not been called.
<b>Complete</b>	The end of the file has been reached successfully.
<b>Property</b>	Reader is at a property.
<b>ObjectStart</b>	Reader is at the start of an object.
<b>Object</b>	Reader is in an object.
<b>ArrayStart</b>	Reader is at the start of an array.
<b>Array</b>	Reader is in an array.
<b>Closed</b>	The Close method has been called.
<b>PostValue</b>	Reader has just read a value.
<b>ConstructorStart</b>	Reader is at the start of a constructor.
<b>Constructor</b>	Reader in a constructor.

<b>Error</b>	An error occurred that prevents the read operation from continuing.
<b>Finished</b>	The end of the file has been reached successfully.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonReaderException

## Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonReaderException**

C#

The exception thrown when an error occurs while reading Json text.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
[SerializableAttribute]  
public class JsonReaderException : JsonException
```

```
<SerializableAttribute> _  
Public Class JsonReaderException _  
    Inherits JsonException
```

```
[SerializableAttribute]  
public ref class JsonReaderException : public JsonEx
```

### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherite

Icon	Member	Description
	<a href="#">JsonReaderException()</a>	Initializes a new instance of the <b>JsonReaderException</b> class.
	<a href="#">JsonReaderException(String)</a>	Initializes a new instance of the

		<b>JsonReaderException</b> class with a specified error message.
	<a href="#">JsonReaderException(String, Exception)</a>	Initializes a new instance of the <b>JsonReaderException</b> class with a specified error message and a reference to the inner exception that is the cause of this exception.
	<a href="#">JsonReaderException(SerializationInfo, StreamingContext)</a>	Initializes a new instance of the <b>JsonReaderException</b> class.
	<a href="#">Data</a>	Gets a collection of key/value pairs that provide additional user-defined information about the exception.  (Inherited from <a href="#">Exception</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resource

		<p>and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetBaseException()</a>	<p>When overridden in a derived class, returns the <a href="#">Exception</a> that is the root cause of one or more subsequent exceptions.</p> <p>(Inherited from <a href="#">Exception</a>.)</p>
	<a href="#">GetHashCode()</a>	<p>Serves as a hash function for a particular type.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetObjectData(SerializationInfo, StreamingContext)</a>	<p>When overridden in a derived class, sets the <a href="#">SerializationInfo</a> with information about the exception.</p> <p>(Inherited from <a href="#">Exception</a>.)</p>
	<a href="#">GetType()</a>	<p>Gets the runtime type of the current instance.</p> <p>(Inherited from <a href="#">Exception</a>.)</p>
	<a href="#">HelpLink</a>	<p>Gets or sets a link to the help file associated with</p>

		<p>this exception.</p> <p>(Inherited from <a href="#">Exception</a>.)</p>
	<a href="#">HResult</a>	<p>Gets or sets HRESULT, coded numerical value that is assigned to a specific exception.</p> <p>(Inherited from <a href="#">Exception</a>.)</p>
	<a href="#">InnerException</a>	<p>Gets the <a href="#">Exception</a> instance that caused the current exception.</p> <p>(Inherited from <a href="#">Exception</a>.)</p>
	<a href="#">LineNumber</a>	<p>Gets the line number indicating where the error occurred.</p>
	<a href="#">LinePosition</a>	<p>Gets the line position indicating where the error occurred.</p>
	<a href="#">MemberwiseClone()</a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">Message</a>	<p>Gets a message that describes the current exception.</p>

		(Inherited from <a href="#">Exception.</a> )
	<a href="#">Path</a>	Gets the path to the JSON where the error occurred.
	<a href="#">Source</a>	Gets or sets the name of the application or the object that causes the error.  (Inherited from <a href="#">Exception.</a> )
	<a href="#">StackTrace</a>	Gets a string representation of the frames on the call stack at the time the current exception was thrown.  (Inherited from <a href="#">Exception.</a> )
	<a href="#">TargetSite</a>	Gets the method that throws the current exception.  (Inherited from <a href="#">Exception.</a> )
	<a href="#">ToString()</a>	Creates and returns a string representation of the current exception.  (Inherited from <a href="#">Exception.</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [Exception](#)

└─ [JsonException](#)

└─ **JsonReaderException**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReaderException](#) ►

### JsonReaderException()

#### Members

Icon	Member	Description
	<a href="#">JsonReaderException()</a>	Initializes a new instance of the <a href="#">JsonReaderException</a> class.
	<a href="#">JsonReaderException(String)</a>	Initializes a new instance of the <a href="#">JsonReaderException</a> class with a specified error message.
	<a href="#">JsonReaderException(String, Exception)</a>	Initializes a new instance of the <a href="#">JsonReaderException</a> class with a specified error message and a reference to the inner exception that is the cause of this exception.
	<a href="#">JsonReaderException(SerializationInfo, StreamingContext)</a>	Initializes a new instance of the <a href="#">JsonReaderException</a> class.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReaderException](#) ►

### JsonReaderException()

Initializes a new instance of the [JsonReaderException](#) class.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonReaderException()</pre>		
	<pre>Public Sub New</pre>	
		<pre>public: JsonReaderException()</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonReaderException

## Constructor (info, context)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReaderException](#) ►

### JsonReaderException(SerializationInfo, StreamingContext)

Initializes a new instance of the [JsonReaderException](#) class.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public JsonReaderException(  
    SerializationInfo info,  
    StreamingContext context  
)
```

```
Public Sub New ( _  
    info As SerializationInfo, _  
    context As StreamingContext _  
)
```

```
public:  
JsonReaderException(  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

#### ▣ Parameters

##### **info** ([SerializationInfo](#))

The [SerializationInfo](#) that holds the serialized object data about the exception being thrown.

##### **context** ([StreamingContext](#))

The [StreamingContext](#) that contains contextual information about the source or destination.

#### ▣ Exceptions

Exception	Condition
<a href="#">ArgumentNullException</a>	The <i>info</i> parameter is null.
<a href="#">SerializationException</a>	The class name is null or <a href="#">HResult</a> is zero (0).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonReaderException

## Constructor (message)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReaderException](#) ►

### JsonReaderException(String)

Initializes a new instance of the [JsonReaderException](#) class with a specified error message.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonReaderException(     <a href="#">string</a> message )</pre>		
	<pre>Public Sub New (      <a href="#">message</a> As <a href="#">String</a>  )</pre>	
		<pre>public: JsonReaderException(     <a href="#">String</a>^ message )</pre>

#### ▣ Parameters

##### **message** ([String](#))

The error message that explains the reason for the exception.

# JsonReaderException Constructor (message, innerException)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReaderException](#) ►

## JsonReaderException(String, Exception)

Initializes a new instance of the [JsonReaderException](#) class with a specified error message and a reference to the inner exception that is the cause of this exception.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonReaderException(  
    string message,  
    Exception innerException  
)
```

```
Public Sub New ( _  
    message As String, _  
    innerException As Exception _  
)
```

```
public:  
JsonReaderException(  
    String^ message,  
    Exception^ innerException  
)
```

### Parameters

#### **message** ([String](#))

The error message that explains the reason for the exception.

#### **innerException** ([Exception](#))

The exception that is the cause of the current exception, or a null reference (Nothing in Visual Basic) if no inner exception is specified.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReaderException](#) ►

## LineNumber

Gets the line number indicating where the error occurred.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public int LineNumber { get; private set; }
```

```
Public Property LineNumber As Integer  
    Get  
    Private Set
```

```
public:  
property int LineNumber {  
    int get ();  
    private: void set (int value);  
}
```

### ▣ Value

The line number indicating where the error occurred.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReaderException](#) ►

## LinePosition

Gets the line position indicating where the error occurred.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public int LinePosition { get; private set; }
```

```
Public Property LinePosition As Integer  
    Get  
    Private Set
```

```
public:  
property int LinePosition {  
    int get ();  
    private: void set (int value);  
}
```

### ▣ Value

The line position indicating where the error occurred.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets the path to the JSON where the error occurred.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public string Path { get; private set; }</pre>		
<pre>Public Property Path As String     Get     Private Set</pre>		
<pre>public: property String^ Path {     String^ get ();     private: void set (String^ value); }</pre>		

## ▣ Value

The path to the JSON where the error occurred.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Documentation **JsonSerializationException Class**

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonSerializationException**

C#

The exception thrown when an error occurs during Json serialization or deserialization.

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
----	--------------	------------

```
[SerializableAttribute]
public class JsonSerializationException : JsonExcept
```

```
<SerializableAttribute> _
Public Class JsonSerializationException _
    Inherits JsonException
```

```
[SerializableAttribute]
public ref class JsonSerializationException : public
```

▣ **Members**

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
			<input checked="" type="checkbox"/> Declare
			<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonSerializationException()</a>	Initializes a new instance of the <b>JsonSerializationException</b> class.
	<a href="#">JsonSerializationException(String)</a>	Initializes a new instance of the <b>JsonSerializationException</b> class.

		the <b>JsonSerializationI</b> class with a specific message.
	<a href="#">JsonSerializationException(String, Exception)</a>	Initializes a new instance of the <b>JsonSerializationI</b> class with a specific message and a reference to the inner exception that is the cause of this exception.
	<a href="#">JsonSerializationException(SerializationInfo, StreamingContext)</a>	Initializes a new instance of the <b>JsonSerializationI</b> class.
	<a href="#">Data</a>	Gets a collection of key-value pairs that provide a user-defined information about the exception.  (Inherited from <a href="#">Exc</a> )
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is the current <a href="#">Object</a> .  (Inherited from <a href="#">Obj</a> )
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to free resources and perform other cleanup operations before the garbage collector reclaims the object.

		is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> )
	<a href="#">GetBaseException()</a>	When overridden in a derived class, returns the <a href="#">Exception</a> that is the cause of one or more subsequent exceptions.  (Inherited from <a href="#">Exception</a> )
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> )
	<a href="#">GetObjectData(SerializationInfo, StreamingContext)</a>	When overridden in a derived class, sets the <a href="#">SerializationInfo</a> with information about the exception.  (Inherited from <a href="#">Exception</a> )
	<a href="#">GetType()</a>	Gets the runtime type of the current instance.  (Inherited from <a href="#">Exception</a> )
	<a href="#">HelpLink</a>	Gets or sets a link to a file associated with the exception.  (Inherited from <a href="#">Exception</a> )
	<a href="#">HResult</a>	Gets or sets the HRESULT-coded numerical value assigned to a specific exception.

		(Inherited from <a href="#">Exc</a>
	<a href="#">InnerException</a>	Gets the <a href="#">Exception</a> that caused the current exception.  (Inherited from <a href="#">Exc</a>
	<a href="#">MemberwiseClone()</a>	Creates a shallow clone of the current <a href="#">Object</a> .  (Inherited from <a href="#">Obj</a>
	<a href="#">Message</a>	Gets a message that describes the current exception.  (Inherited from <a href="#">Exc</a>
	<a href="#">Source</a>	Gets or sets the name of the application or the object that causes the error.  (Inherited from <a href="#">Exc</a>
	<a href="#">StackTrace</a>	Gets a string representation of the frames on the stack at the time the exception was thrown.  (Inherited from <a href="#">Exc</a>
	<a href="#">TargetSite</a>	Gets the method that caused the current exception.  (Inherited from <a href="#">Exc</a>
	<a href="#">ToString()</a>	Creates and returns a string representation of the exception.  (Inherited from <a href="#">Exc</a>

## ▣ Inheritance Hierarchy

[Object](#)

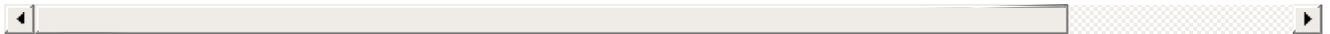
└─ [Exception](#)

└─ [JsonException](#)

└─ **JsonSerializationException**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



Documentation **JsonSerializationException Constructor**

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializationException](#) ►

**JsonSerializationException()**

**Members**

Icon	Member	Description
	<a href="#">JsonSerializationException()</a>	Initializes a new instance of the <a href="#">JsonSerializationException</a> class.
	<a href="#">JsonSerializationException(String)</a>	Initializes a new instance of the <a href="#">JsonSerializationException</a> class with a specific message.
	<a href="#">JsonSerializationException(String, Exception)</a>	Initializes a new instance of the <a href="#">JsonSerializationException</a> class with a specific message and a reference to the inner exception that is the cause of this exception.
	<a href="#">JsonSerializationException(SerializationInfo, StreamingContext)</a>	Initializes a new instance of the <a href="#">JsonSerializationException</a> class.

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)



## JsonSerializationException()

Initializes a new instance of the [JsonSerializationException](#) class.

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSerializationException()
```

```
Public Sub New
```

```
public:  
JsonSerializationException()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Documentation **JsonSerializationException Constructor (info, context)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializationException](#) ►**JsonSerializationException(SerializationInfo, StreamingContext)**Initializes a new instance of the [JsonSerializationException](#) class. ▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public JsonSerializationException(  
    SerializationInfo info,  
    StreamingContext context  
)
```

```
Public Sub New ( _  
    info As SerializationInfo, _  
    context As StreamingContext _  
)
```

```
public:  
JsonSerializationException(  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

▣ **Parameters****info** ([SerializationInfo](#))

The [SerializationInfo](#) that holds the serialized object data about the exception being thrown.

**context** ([StreamingContext](#))

The [StreamingContext](#) that contains contextual information about the source or destination.

## ▣ Exceptions

Exception	Condition
<a href="#">ArgumentNullException</a>	The <i>info</i> parameter is null.
<a href="#">SerializationException</a>	The class name is null or <a href="#">HResult</a> is zero (0).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Documentation **JsonSerializationException Constructor (message)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializationException](#) ►**JsonSerializationException(String)**

C#

Initializes a new instance of the [JsonSerializationException](#) class with a specified error message.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public JsonSerializationException(  
    string message  
)
```

```
Public Sub New ( _  
    message As String _  
)
```

```
public:  
JsonSerializationException(  
    String^ message  
)
```

▣ **Parameters****message (String)**

The error message that explains the reason for the exception.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Documentation **JsonSerializationException Constructor (message, innerException)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializationException](#) ►**JsonSerializationException(String, Exception)**

C#

Initializes a new instance of the [JsonSerializationException](#) class with a specified error message and a reference to the inner exception that is the cause of this exception.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public JsonSerializationException(  
    string message,  
    Exception innerException  
)
```

```
Public Sub New ( _  
    message As String, _  
    innerException As Exception _  
)
```

```
public:  
JsonSerializationException(  
    String^ message,  
    Exception^ innerException  
)
```

▣ **Parameters****message (String)**

The error message that explains the reason for the exception.

**innerException (Exception)**

The exception that is the cause of the current exception, or a null reference (Nothing in Visual Basic) if no inner exception is specified.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

# JsonSerializer Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonSerializer**

Serializes and deserializes objects into and from the JSON format. The **JsonSerializer** enables you to control how objects are encoded into JSON.

## Declaration Syntax

C#	Visual Basic	Visual C++
<code>public class JsonSerializer</code>		

<code>Public Class JsonSerializer</code>
--

<code>public ref class JsonSerializer</code>
--

## Members

All Members	Constructors	Methods	Properties	Events
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declared
				<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonSerializer()</a>	Initializes a new instance of the <b>JsonSerializer</b> class.
	<a href="#">Binder</a>	Gets or sets the <a href="#">SerializationBinder</a> used by the serializer when resolving type names.
	<a href="#">CheckAdditionalContent</a>	Gets a value indicating whether there will be a check for additional content.

		JSON content after deserializing an object.
	<a href="#">ConstructorHandling</a>	Gets or sets how constructors are used during deserialization.
	<a href="#">Context</a>	Gets or sets the <a href="#">StreamingContext</a> used by the serializer when invoking serialization callback methods.
	<a href="#">ContractResolver</a>	Gets or sets the contract resolver used by the serializer when serializing .NET objects to JSON and vice versa.
	<a href="#">Converters</a>	Gets a collection <a href="#">JsonConverter</a> that will be used during serialization.
	<a href="#">Create(JsonSerializerSettings)</a>	Creates a new <b>JsonSerializer</b> instance using the specified <a href="#">JsonSerializerSettings</a> .
	<a href="#">Culture</a>	Gets or sets the culture used when reading JSON. Defaults to <a href="#">InvariantCulture</a> .

	<a href="#">DateFormatHandling</a>	Get or set how dates are written to JSON text.
	<a href="#">DateParseHandling</a>	Get or set how date formatted strings, e.g. "VDate(1198908717056)V" and "2012-03-21T05:40Z", are parsed when reading JSON.
	<a href="#">DateTimeZoneHandling</a>	Get or set how <a href="#">DateTime</a> time zones are handling during serialization and deserialization.
	<a href="#">DefaultValueHandling</a>	Get or set how null default are handled during serialization and deserialization.
	<a href="#">Deserialize(JsonReader)</a>	Deserializes the Json structure contained by the specified <a href="#">JsonReader</a> .
	<a href="#">Deserialize(TextReader, Type)</a>	Deserializes the Json structure contained by the specified <a href="#">StringReader</a> into an instance of the specified type.
	<a href="#">Deserialize&lt;T&gt;(JsonReader)</a>	Deserializes the Json structure contained by the specified <a href="#">JsonReader</a> into an instance of

		the specified type.
	<a href="#">Deserialize(JsonReader, Type)</a>	Deserializes the Json structure contained by the specified <a href="#">JsonReader</a> into an instance of the specified type.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Error</a>	Occurs when the <b>JsonSerializer</b> errors during serialization and deserialization.
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">Formatting</a>	Indicates how JSON text output is formatted.
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current

		instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MaxDepth</a>	Gets or sets the maximum depth allowed when reading JSON. Reading past this depth will throw a <a href="#">JsonReaderException</a> .
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">MissingMemberHandling</a>	Get or set how missing members (e.g. JSON contains a property that isn't a member on the object) are handled during deserialization.
	<a href="#">NullValueHandling</a>	Get or set how null values are handled during serialization and deserialization.
	<a href="#">ObjectCreationHandling</a>	Gets or sets how objects are created during deserialization.
	<a href="#">Populate(TextReader, Object)</a>	Populates the JSON values onto the target object.
	<a href="#">Populate(JsonReader, Object)</a>	Populates the JSON values onto the target object.

	<a href="#">PreserveReferencesHandling</a>	Gets or sets how object references are preserved by the serializer.
	<a href="#">ReferenceLoopHandling</a>	Get or set how reference loops (e.g. a class referencing itself) is handled.
	<a href="#">ReferenceResolver</a>	Gets or sets the <a href="#">IReferenceResolver</a> used by the serializer when resolving references.
	<a href="#">Serialize(TextWriter, Object)</a>	Serializes the specified <a href="#">Object</a> and writes the Json structure to a <a href="#">Stream</a> using the specified <a href="#">TextWriter</a> .
	<a href="#">Serialize(JsonWriter, Object)</a>	Serializes the specified <a href="#">Object</a> and writes the Json structure to a <a href="#">Stream</a> using the specified <a href="#">JsonWriter</a> .
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
		

	<a href="#">TypeNameAssemblyFormat</a>	Gets or sets how a type name assembly is written and resolved by the serializer.
	<a href="#">TypeNameHandling</a>	Gets or sets how type name writing and reading is handled by the serializer.

## ▣ Inheritance Hierarchy

[Object](#)

└─ **JsonSerializer**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► **JsonSerializer()**

Initializes a new instance of the [JsonSerializer](#) class.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSerializer()
```

```
Public Sub New
```

```
public:  
JsonSerializer()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► **Binder**

C#

Gets or sets the [SerializationBinder](#) used by the serializer when resolving type names.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual SerializationBinder Binder { get; set;
```

```
Public Overridable Property Binder As SerializationB  
    Get  
    Set
```

```
public:  
virtual property SerializationBinder^ Binder {  
    SerializationBinder^ get ();  
    void set (SerializationBinder^ value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### CheckAdditionalContent

Gets a value indicating whether there will be a check for additional JSON content after deserializing an object.

C#

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual bool CheckAdditionalContent { get; set;
```

```
Public Overridable Property CheckAdditionalContent As  
    Get  
    Set
```

```
public:  
virtual property bool CheckAdditionalContent {  
    bool get ();  
    void set (bool value);  
}
```

#### Value

**true** if there will be a check for additional JSON content after deserializing an object; otherwise, **false**.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### ConstructorHandling

C#

Gets or sets how constructors are used during deserialization.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual ConstructorHandling ConstructorHandli
```

```
Public Overridable Property ConstructorHandling As C  
    Get  
    Set
```

```
public:  
virtual property ConstructorHandling ConstructorHand  
    ConstructorHandling get ();  
    void set (ConstructorHandling value);  
}
```

#### ▣ Value

The constructor handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► **Context**

Gets or sets the [StreamingContext](#) used by the serializer when invoking serialization callback methods.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual StreamingContext Context { get; set; }
```

```
Public Overridable Property Context As StreamingCont  
    Get  
    Set
```

```
public:  
virtual property StreamingContext Context {  
    StreamingContext get ();  
    void set (StreamingContext value);  
}
```

## ▣ Value

The context.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### ContractResolver

Gets or sets the contract resolver used by the serializer when serializing .NET objects to JSON and vice versa.

#### Declaration Syntax

C#

Visual Basic

Visual C++

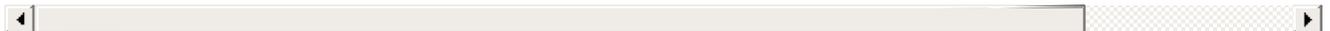
```
public virtual IContractResolver ContractResolver {
```

```
    Public Overridable Property ContractResolver As ICon  
        Get  
        Set
```

```
public:  
virtual property IContractResolver^ ContractResolver  
    IContractResolver^ get ();  
    void set (IContractResolver^ value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► **Converters**

Gets a collection [JsonConverter](#) that will be used during serialization.

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual JsonConverterCollection Converters {
```

```
    Public Overridable ReadOnly Property Converters As J  
        Get
```

```
public:  
virtual property JsonConverterCollection^ Converters  
    JsonConverterCollection^ get ();  
}
```

## Value

Collection [JsonConverter](#) that will be used during serialization.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

## Create(JsonSerializerSettings)

Creates a new [JsonSerializer](#) instance using the specified [JsonSerializerSettings](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JsonSerializer Create(  
    JsonSerializerSettings settings  
)
```

```
Public Shared Function Create ( _  
    settings As JsonSerializerSettings _  
) As JsonSerializer
```

```
public:  
static JsonSerializer^ Create(  
    JsonSerializerSettings^ settings  
)
```

### ▣ Parameters

#### **settings** ([JsonSerializerSettings](#))

The settings to be applied to the [JsonSerializer](#).

### ▣ Return Value

A new [JsonSerializer](#) instance using the specified [JsonSerializerSetting](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► **Culture**

Gets or sets the culture used when reading JSON. Defaults to [InvariantCulture](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual CultureInfo Culture { get; set; }
```

```
Public Overridable Property Culture As CultureInfo  
    Get  
    Set
```

```
public:  
virtual property CultureInfo^ Culture {  
    CultureInfo^ get ();  
    void set (CultureInfo^ value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### DateFormatHandling

Get or set how dates are written to JSON text.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual DateFormatHandling DateFormatHandling
```

```
Public Overridable Property DateFormatHandling As Da  
    Get  
    Set
```

```
public:  
virtual property DateFormatHandling DateFormatHandli  
    DateFormatHandling get ();  
    void set (DateFormatHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### DateParseHandling

Get or set how date formatted strings, e.g. "VDate(1198908717056)V" and "2012-03-21T05:40Z", are parsed when reading JSON.

#### Declaration Syntax

C#

Visual Basic

Visual C++

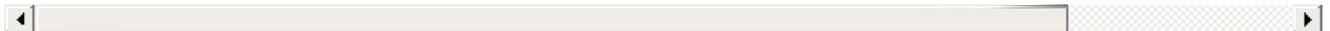
```
public virtual DateParseHandling DateParseHandling {
```

```
    Public Overridable Property DateParseHandling As Dat  
        Get  
        Set
```

```
public:  
virtual property DateParseHandling DateParseHandling  
    DateParseHandling get ();  
    void set (DateParseHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### DateTimeZoneHandling

Get or set how [DateTime](#) time zones are handling during serialization and deserialization.

#### Declaration Syntax

C#

Visual Basic

Visual C++

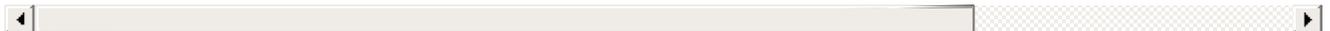
```
public virtual DateTimeZoneHandling DateTimeZoneHand
```

```
Public Overridable Property DateTimeZoneHandling As  
    Get  
    Set
```

```
public:  
virtual property DateTimeZoneHandling DateTimeZoneHa  
    DateTimeZoneHandling get ();  
    void set (DateTimeZoneHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### DefaultValueHandling

Get or set how null default are handled during serialization and deserialization.

#### Declaration Syntax

C#

Visual Basic

Visual C++

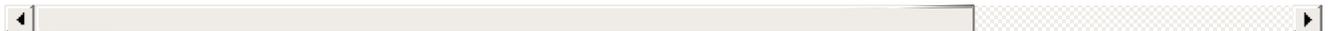
```
public virtual DefaultValueHandling DefaultValueHand
```

```
Public Overridable Property DefaultValueHandling As  
    Get  
    Set
```

```
public:  
virtual property DefaultValueHandling DefaultValueHa  
    DefaultValueHandling get ();  
    void set (DefaultValueHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



C# ▼

[-] **Members**

Icon	Member	Description
	<a href="#">Deserialize(JsonReader)</a>	Deserializes the Json structure contained by the specified <a href="#">JsonReader</a> .
	<a href="#">Deserialize(TextReader, Type)</a>	Deserializes the Json structure contained by the specified <a href="#">StringReader</a> into an instance of the specified type.
	<a href="#">Deserialize&lt;T&gt;(JsonReader)</a>	Deserializes the Json structure contained by the specified <a href="#">JsonReader</a> into an instance of the specified type.
	<a href="#">Deserialize(JsonReader, Type)</a>	Deserializes the Json structure contained by the specified <a href="#">JsonReader</a> into an instance of the specified type.

## (reader)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### Deserialize(JsonReader)

Deserializes the Json structure contained by the specified [JsonReader](#).

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">Object</a> Deserialize(     <a href="#">JsonReader</a> reader )</pre>		
	<pre>Public Function Deserialize ( _     reader As <a href="#">JsonReader</a> _ ) As <a href="#">Object</a></pre>	
		<pre>public: <a href="#">Object</a>^ Deserialize(     <a href="#">JsonReader</a>^ reader )</pre>

#### ▣ Parameters

##### *reader* ([JsonReader](#))

The [JsonReader](#) that contains the JSON structure to deserialize.

#### ▣ Return Value

The [Object](#) being deserialized.

# Deserialize<T> Method (reader)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► **Deserialize<T> (JsonReader)**

Deserializes the Json structure contained by the specified [JsonReader](#) into an instance of the specified type.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public T Deserialize<T>(
    JsonReader reader
)
```

```
Public Function Deserialize(Of T) ( _
    reader As JsonReader _
) As T
```

```
public:
generic<typename T>
T Deserialize(
    JsonReader^ reader
)
```

## Generic Template Parameters

***T***

The type of the object to deserialize.

## Parameters

***reader*** ([JsonReader](#))

The [JsonReader](#) containing the object.

## Return Value

The instance of *T* being deserialized.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## (reader, objectType)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### Deserialize(JsonReader, Type)

Deserializes the Json structure contained by the specified [JsonReader](#) into an instance of the specified type.

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">Object</a> Deserialize(     <a href="#">JsonReader</a> reader,     <a href="#">Type</a> objectType )</pre>	<pre>Public Function Deserialize ( _     reader As <a href="#">JsonReader</a>, _     objectType As <a href="#">Type</a> _ ) As <a href="#">Object</a></pre>	

#### Parameters

##### *reader* ([JsonReader](#))

The [JsonReader](#) containing the object.

##### *objectType* ([Type](#))

The [Type](#) of object being deserialized.

#### Return Value

The instance of *objectType* being deserialized.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## (reader, objectType)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### Deserialize(TextReader, Type)

Deserializes the Json structure contained by the specified [StringReader](#) into an instance of the specified type.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Object Deserialize(  
    TextReader reader,  
    Type objectType  
)
```

```
Public Function Deserialize ( _  
    reader As TextReader, _  
    objectType As Type _  
) As Object
```

```
public:  
Object^ Deserialize(  
    TextReader^ reader,  
    Type^ objectType  
)
```

#### ▣ Parameters

##### **reader** ([TextReader](#))

The [TextReader](#) containing the object.

##### **objectType** ([Type](#))

The [Type](#) of object being deserialized.

#### ▣ Return Value

The instance of *objectType* being deserialized.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► **Error**

C#

Occurs when the [JsonSerializer](#) errors during serialization and deserialization.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual event EventHandler<ErrorEventArgs> Er
```

```
Public Overridable Event Error As EventHandler(Of Er
```

```
public:  
virtual event EventHandler<ErrorEventArgs^>^ Error  
void add (EventHandler<ErrorEventArgs^>^ val  
void remove (EventHandler<ErrorEventArgs^>^  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



Indicates how JSON text output is formatted.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public virtual <a href="#">Formatting</a> Formatting { <a href="#">get</a>; <a href="#">set</a>; }</pre>		
	<pre>Public Overridable Property Formatting As <a href="#">Formatting</a>     Get     Set</pre>	
		<pre>public: virtual property <a href="#">Formatting</a> Formatting {     <a href="#">Formatting</a> <a href="#">get</a> ();     void <a href="#">set</a> (<a href="#">Formatting</a> <i>value</i>); }</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► **MaxDepth**

Gets or sets the maximum depth allowed when reading JSON. Reading past this depth will throw a [JsonReaderException](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual Nullable<int> MaxDepth { get; set; }
```

```
Public Overridable Property MaxDepth As Nullable(Of  
    Get  
    Set
```

```
public:  
virtual property Nullable<int> MaxDepth {  
    Nullable<int> get ();  
    void set (Nullable<int> value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# MissingMemberHandling Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

## MissingMemberHandling

Get or set how missing members (e.g. JSON contains a property that isn't a member on the object) are handled during deserialization.

### Declaration Syntax

C#

Visual Basic

Visual C++

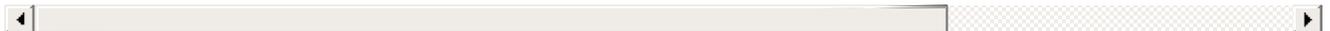
```
public virtual MissingMemberHandling MissingMemberHa
```

```
Public Overridable Property MissingMemberHandling As  
    Get  
    Set
```

```
public:  
virtual property MissingMemberHandling MissingMember  
    MissingMemberHandling get ();  
    void set (MissingMemberHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### NullValueHandling

Get or set how null values are handled during serialization and deserialization.

#### Declaration Syntax

C#

Visual Basic

Visual C++

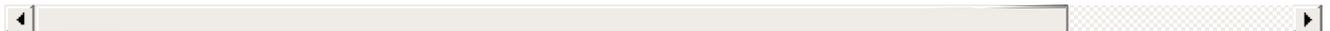
```
public virtual NullValueHandling NullValueHandling {
```

```
    Public Overridable Property NullValueHandling As NullValueHandling  
        Get  
        Set
```

```
public:  
virtual property NullValueHandling NullValueHandling  
    NullValueHandling get ();  
    void set (NullValueHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



**Property**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►**ObjectCreationHandling**

C#

Gets or sets how objects are created during deserialization.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual ObjectCreationHandling ObjectCreation
```

```
Public Overridable Property ObjectCreationHandling A  
    Get  
    Set
```

```
public:  
virtual property ObjectCreationHandling ObjectCreati  
    ObjectCreationHandling get ();  
    void set (ObjectCreationHandling value);  
}
```

▣ **Value**

The object creation handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



C#

## Members

Icon	Member	Description
	<a href="#">Populate(TextReader, Object)</a>	Populates the JSON values onto the target object.
	<a href="#">Populate(JsonReader, Object)</a>	Populates the JSON values onto the target object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# Populate Method (reader, target)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

## Populate(JsonReader, Object)

C#

Populates the JSON values onto the target object.

### Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public void Populate(  
    JsonReader reader,  
    Object target  
)
```

```
Public Sub Populate ( _  
    reader As JsonReader, _  
    target As Object _  
)
```

```
public:  
void Populate(  
    JsonReader^ reader,  
    Object^ target  
)
```

### Parameters

#### **reader** ([JsonReader](#))

The [JsonReader](#) that contains the JSON structure to reader values from.

#### **target** ([Object](#))

The target object to populate values onto.

(4.5.6.14930)

## Populate Method (reader, target)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### Populate(TextReader, Object)

Populates the JSON values onto the target object.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public void Populate(     <a href="#">TextReader</a> reader,     <a href="#">Object</a> target )</pre>		
	<pre>Public Sub Populate (      reader As <a href="#">TextReader</a>,      target As <a href="#">Object</a>  )</pre>	
		<pre>public: void Populate(     <a href="#">TextReader</a>^ reader,     <a href="#">Object</a>^ target )</pre>

#### ▣ Parameters

##### **reader** ([TextReader](#))

The [TextReader](#) that contains the JSON structure to reader values from.

##### **target** ([Object](#))

The target object to populate values onto.

(4.5.6.14930)

## PreserveReferencesHandling

Gets or sets how object references are preserved by the serializer. C# ▼

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

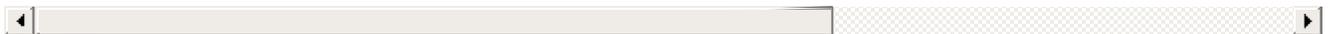
```
public virtual PreserveReferencesHandling PreserveRe
```

```
Public Overridable Property PreserveReferencesHandli  
    Get  
    Set
```

```
public:  
virtual property PreserveReferencesHandling Preserve  
    PreserveReferencesHandling get ();  
    void set (PreserveReferencesHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# ReferenceLoopHandling Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

## ReferenceLoopHandling

Get or set how reference loops (e.g. a class referencing itself) is handle

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual ReferenceLoopHandling ReferenceLoopHa
```

```
Public Overridable Property ReferenceLoopHandling As  
    Get  
    Set
```

```
public:  
virtual property ReferenceLoopHandling ReferenceLoop  
    ReferenceLoopHandling get ();  
    void set (ReferenceLoopHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### ReferenceResolver

Gets or sets the [IReferenceResolver](#) used by the serializer when resolving references.

#### Declaration Syntax

C#

Visual Basic

Visual C++

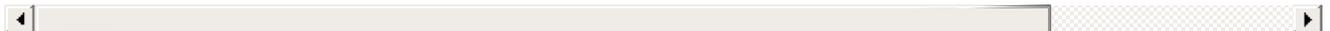
```
public virtual IReferenceResolver ReferenceResolver
```

```
Public Overridable Property ReferenceResolver As IRe  
    Get  
    Set
```

```
public:  
virtual property IReferenceResolver^ ReferenceResolv  
    IReferenceResolver^ get ();  
    void set (IReferenceResolver^ value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



C#

## Members

Icon	Member	Description
	<a href="#">Serialize(TextWriter, Object)</a>	Serializes the specified <a href="#">Object</a> and writes the Json structure to a <a href="#">Stream</a> using the specified <a href="#">TextWriter</a> .
	<a href="#">Serialize(JsonWriter, Object)</a>	Serializes the specified <a href="#">Object</a> and writes the Json structure to a <a href="#">Stream</a> using the specified <a href="#">JsonWriter</a> .

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (jsonWriter, value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### Serialize(JsonWriter, Object)

Serializes the specified [Object](#) and writes the Json structure to a [Stream](#) using the specified [JsonWriter](#).

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public void Serialize(     <a href="#">JsonWriter</a> jsonWriter,     <a href="#">Object</a> value )</pre>	<pre>Public Sub Serialize ( _     jsonWriter As <a href="#">JsonWriter</a>, _     value As <a href="#">Object</a> _ )</pre>	<pre>public: void Serialize(     <a href="#">JsonWriter</a>^ jsonWriter,     <a href="#">Object</a>^ value )</pre>

#### Parameters

##### *jsonWriter* ([JsonWriter](#))

The [JsonWriter](#) used to write the Json structure.

##### *value* ([Object](#))

The [Object](#) to serialize.

(4.5.6.14930)

## (TextWriter, value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### Serialize(TextWriter, Object)

Serializes the specified [Object](#) and writes the Json structure to a [Stream](#) using the specified [TextWriter](#).

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public void Serialize(     TextWriter textWriter,     Object value )</pre>		
	<pre>Public Sub Serialize ( _     textWriter As TextWriter, _     value As Object _ )</pre>	
		<pre>public: void Serialize(     TextWriter^ textWriter,     Object^ value )</pre>

#### Parameters

##### **textWriter** ([TextWriter](#))

The [TextWriter](#) used to write the Json structure.

##### **value** ([Object](#))

The [Object](#) to serialize.

(4.5.6.14930)

**TypeNameAssemblyFormat**

Gets or sets how a type name assembly is written and resolved by the serializer. C# ▼

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual FormatterAssemblyStyle TypeNameAssemb
```

```
Public Overridable Property TypeNameAssemblyFormat A  
    Get  
    Set
```

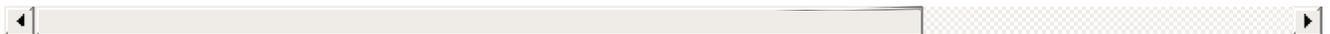
```
public:  
virtual property FormatterAssemblyStyle TypeNameAsse  
    FormatterAssemblyStyle get ();  
    void set (FormatterAssemblyStyle value);  
}
```

▣ **Value**

The type name assembly format.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ►

### TypeNameHandling

Gets or sets how type name writing and reading is handled by the serializer.

#### Declaration Syntax

C#

Visual Basic

Visual C++

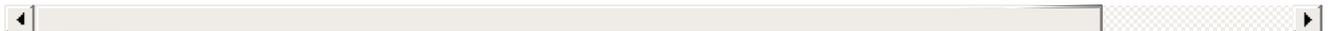
```
public virtual TypeNameHandling TypeNameHandling { g
```

```
Public Overridable Property TypeNameHandling As Type  
    Get  
    Set
```

```
public:  
virtual property TypeNameHandling TypeNameHandling {  
    TypeNameHandling get ();  
    void set (TypeNameHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# JsonSerializerSettings

## Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonSerializerSettings**

Specifies the settings on a [JsonSerializer](#) object.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonSerializerSettings
```

```
Public Class JsonSerializerSettings
```

```
public ref class JsonSerializerSettings
```

### Members

All Members

Constructors

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declare
- Inherited

Icon	Member	Description
	<a href="#">JsonSerializerSettings()</a>	Initializes a new instance of the <b>JsonSerializerSettings</b> class.
	<a href="#">Binder</a>	Gets or sets the <a href="#">SerializationBinder</a> used by the serializer when resolving type names.
	<a href="#">CheckAdditionalContent</a>	Gets a value indicating whether there will be a check for additional

		content after deserializing an object.
	<a href="#">ConstructorHandling</a>	Gets or sets how constructors are used during deserialization.
	<a href="#">Context</a>	Gets or sets the <a href="#">StreamingContext</a> used by the serializer when invoking serialization callback methods.
	<a href="#">ContractResolver</a>	Gets or sets the contract resolver used by the serializer when serializing .NET objects to JSON and vice versa.
	<a href="#">Converters</a>	Gets or sets a collection <a href="#">JsonConverter</a> that will be used during serialization.
	<a href="#">Culture</a>	Gets or sets the culture used when reading JSON. Defaults to <a href="#">InvariantCulture</a> .
	<a href="#">DateFormatHandling</a>	Get or set how dates are written to JSON text.
	<a href="#">DateParseHandling</a>	Get or set how date formatted

		strings, e.g. "VDate(1198908717056)V" and "2012-03-21T05:40Z", are parsed when reading JSON.
	<a href="#">DateTimeZoneHandling</a>	Get or set how <a href="#">DateTime</a> time zones are handling during serialization and deserialization.
	<a href="#">DefaultValueHandling</a>	Gets or sets how null default are handled during serialization and deserialization.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Error</a>	Gets or sets the error handler called during serialization and deserialization.
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">Formatting</a>	Indicates how JSON text output is formatted.

	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MaxDepth</a>	Gets or sets the maximum depth allowed when reading JSON. Reading past this depth will throw a <a href="#">JsonReaderException</a> .
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">MissingMemberHandling</a>	Gets or sets how missing member (e.g. JSON contains a property that isn't a member on the object) are handled during deserialization.
	<a href="#">NullValueHandling</a>	Gets or sets how null values are handled during serialization and deserialization.
	<a href="#">ObjectCreationHandling</a>	Gets or sets how objects are created during deserialization.
		

	<a href="#">PreserveReferencesHandling</a>	Gets or sets how object reference are preserved by the serializer.
	<a href="#">ReferenceLoopHandling</a>	Gets or sets how reference loops (e.g. a class referencing itself) is handled.
	<a href="#">ReferenceResolver</a>	Gets or sets the <a href="#">IReferenceResolver</a> used by the serializer when resolving references.
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">TypeNameAssemblyFormat</a>	Gets or sets how a type name assembly is written and resolved by the serializer.
	<a href="#">TypeNameHandling</a>	Gets or sets how type name writin and reading is handled by the serializer.

## ▣ Inheritance Hierarchy

[Object](#)

└─ **JsonSerializerSettings**

(4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

### JsonSerializerSettings()

Initializes a new instance of the [JsonSerializerSettings](#) class.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSerializerSettings()
```

```
Public Sub New
```

```
public:  
JsonSerializerSettings()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ► **Binder**

C#

Gets or sets the [SerializationBinder](#) used by the serializer when resolving type names.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public SerializationBinder Binder { get; set; }
```

```
Public Property Binder As SerializationBinder  
    Get  
    Set
```

```
public:  
property SerializationBinder^ Binder {  
    SerializationBinder^ get ();  
    void set (SerializationBinder^ value);  
}
```

## ▣ Value

The binder.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

### CheckAdditionalContent

Gets a value indicating whether there will be a check for additional content after deserializing an object.

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public bool CheckAdditionalContent { get; set; }</pre>		
	<pre>Public Property CheckAdditionalContent As Boolean     Get     Set</pre>	
		<pre>public: property bool CheckAdditionalContent {     bool get ();     void set (bool value); }</pre>

#### Value

`true` if there will be a check for additional content after deserializing an object; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

### ConstructorHandling

Gets or sets how constructors are used during deserialization.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public ConstructorHandling ConstructorHandling { get
```

```
Public Property ConstructorHandling As ConstructorHa  
    Get  
    Set
```

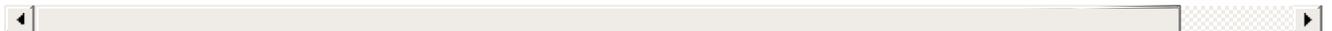
```
public:  
property ConstructorHandling ConstructorHandling {  
    ConstructorHandling get ();  
    void set (ConstructorHandling value);  
}
```

#### ▣ Value

The constructor handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ► **Context**

C#

Gets or sets the [StreamingContext](#) used by the serializer when invoking serialization callback methods.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public StreamingContext Context { get; set; }
```

```
Public Property Context As StreamingContext  
    Get  
    Set
```

```
public:  
property StreamingContext Context {  
    StreamingContext get ();  
    void set (StreamingContext value);  
}
```

## ▣ Value

The context.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

### ContractResolver

Gets or sets the contract resolver used by the serializer when serializing .NET objects to JSON and vice versa.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public IContractResolver ContractResolver { get; set;
```

```
Public Property ContractResolver As IContractResolve  
    Get  
    Set
```

```
public:  
property IContractResolver^ ContractResolver {  
    IContractResolver^ get ();  
    void set (IContractResolver^ value);  
}
```

#### Value

The contract resolver.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

## Converters

Gets or sets a collection [JsonConverter](#) that will be used during serialization.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public IList<JsonConverter> Converters { get; set; }
```

```
Public Property Converters As IList(Of JsonConverter)  
    Get  
    Set
```

```
public:  
property IList<JsonConverter>^ Converters {  
    IList<JsonConverter>^ get ();  
    void set (IList<JsonConverter>^ value);  
}
```

### Value

The converters.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ► **Culture**

Gets or sets the culture used when reading JSON. Defaults to [InvariantCulture](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public CultureInfo Culture { get; set; }
```

```
Public Property Culture As CultureInfo  
    Get  
    Set
```

```
public:  
property CultureInfo^ Culture {  
    CultureInfo^ get ();  
    void set (CultureInfo^ value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

### DateFormatHandling

C#

Get or set how dates are written to JSON text.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public DateFormatHandling DateFormatHandling { get;
```

```
Public Property DateFormatHandling As DateFormatHand  
    Get  
    Set
```

```
public:  
property DateFormatHandling DateFormatHandling {  
    DateFormatHandling get ();  
    void set (DateFormatHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

### DateParseHandling

Get or set how date formatted strings, e.g. "VDate(1198908717056)V" and "2012-03-21T05:40Z", are parsed when reading JSON.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

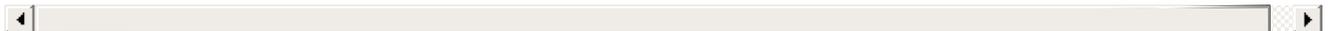
```
public DateParseHandling DateParseHandling { get; se
```

```
Public Property DateParseHandling As DateParseHandli  
    Get  
    Set
```

```
public:  
property DateParseHandling DateParseHandling {  
    DateParseHandling get ();  
    void set (DateParseHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

### DateTimeZoneHandling

Get or set how [DateTime](#) time zones are handling during serialization and deserialization.

#### Declaration Syntax

C#

Visual Basic

Visual C++

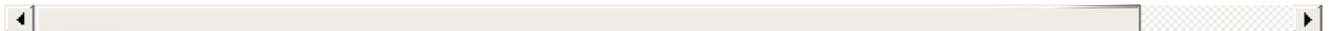
```
public DateTimeZoneHandling DateTimeZoneHandling { g
```

```
Public Property DateTimeZoneHandling As DateTimeZone  
    Get  
    Set
```

```
public:  
property DateTimeZoneHandling DateTimeZoneHandling {  
    DateTimeZoneHandling get ();  
    void set (DateTimeZoneHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

### DefaultValueHandling

Gets or sets how null default are handled during serialization and deserialization.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public DefaultValueHandling DefaultValueHandling { g
```

```
Public Property DefaultValueHandling As DefaultValue  
    Get  
    Set
```

```
public:  
property DefaultValueHandling DefaultValueHandling {  
    DefaultValueHandling get ();  
    void set (DefaultValueHandling value);  
}
```

#### Value

The default value handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ► **Error**

C#

Gets or sets the error handler called during serialization and deserialization.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public EventHandler<ErrorEventArgs> Error { get; set
```

```
Public Property Error As EventHandler(Of ErrorEventA  
    Get  
    Set
```

```
public:  
property EventHandler<ErrorEventArgs^>^ Error {  
    EventHandler<ErrorEventArgs^>^ get ();  
    void set (EventHandler<ErrorEventArgs^>^ val  
}
```

## Value

The error handler called during serialization and deserialization.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

## Formatting

Indicates how JSON text output is formatted.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Formatting Formatting { get; set; }
```

```
Public Property Formatting As Formatting  
    Get  
    Set
```

```
public:  
property Formatting Formatting {  
    Formatting get ();  
    void set (Formatting value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

## MaxDepth

Gets or sets the maximum depth allowed when reading JSON. Reading past this depth will throw a [JsonReaderException](#).

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<int> MaxDepth { get; set; }
```

```
Public Property MaxDepth As Nullable(Of Integer)  
    Get  
    Set
```

```
public:  
property Nullable<int> MaxDepth {  
    Nullable<int> get ();  
    void set (Nullable<int> value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

### MissingMemberHandling

Gets or sets how missing members (e.g. JSON contains a property that isn't a member on the object) are handled during deserialization.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public MissingMemberHandling MissingMemberHandling {
```

```
    Public Property MissingMemberHandling As MissingMemberHandling  
        Get  
        Set
```

```
public:  
property MissingMemberHandling MissingMemberHandling  
    MissingMemberHandling get ();  
    void set (MissingMemberHandling value);  
}
```

#### Value

Missing member handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

### NullValueHandling

Gets or sets how null values are handled during serialization and deserialization.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public NullValueHandling NullValueHandling { get; se
```

```
Public Property NullValueHandling As NullValueHandli  
    Get  
    Set
```

```
public:  
property NullValueHandling NullValueHandling {  
    NullValueHandling get ();  
    void set (NullValueHandling value);  
}
```

#### Value

Null value handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



**Property**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►**ObjectCreationHandling**

Gets or sets how objects are created during deserialization.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public ObjectCreationHandling ObjectCreationHandling
```

```
Public Property ObjectCreationHandling As ObjectCrea  
    Get  
    Set
```

```
public:  
property ObjectCreationHandling ObjectCreationHandli  
    ObjectCreationHandling get ();  
    void set (ObjectCreationHandling value);  
}
```

▣ **Value**

The object creation handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## PreserveReferencesHandling

Gets or sets how object references are preserved by the serializer. C# ▼

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public PreserveReferencesHandling PreserveReferences
```

```
Public Property PreserveReferencesHandling As Preser  
    Get  
    Set
```

```
public:  
property PreserveReferencesHandling PreserveReferenc  
    PreserveReferencesHandling get ();  
    void set (PreserveReferencesHandling value);  
}
```

### ▣ Value

The preserve references handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

### ReferenceLoopHandling

Gets or sets how reference loops (e.g. a class referencing itself) is handled.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public ReferenceLoopHandling ReferenceLoopHandling {
```

```
    Public Property ReferenceLoopHandling As ReferenceLo  
        Get  
        Set
```

```
public:  
property ReferenceLoopHandling ReferenceLoopHandling  
    ReferenceLoopHandling get ();  
    void set (ReferenceLoopHandling value);  
}
```

#### Value

Reference loop handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

### ReferenceResolver

Gets or sets the [IReferenceResolver](#) used by the serializer when resolving references.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public IReferenceResolver ReferenceResolver { get; s
```

```
Public Property ReferenceResolver As IReferenceResol  
    Get  
    Set
```

```
public:  
property IReferenceResolver^ ReferenceResolver {  
    IReferenceResolver^ get ();  
    void set (IReferenceResolver^ value);  
}
```

#### Value

The reference resolver.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



**TypeNameAssemblyFormat**

Gets or sets how a type name assembly is written and resolved by the serializer.

C#

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public FormatterAssemblyStyle TypeNameAssemblyFormat
```

```
Public Property TypeNameAssemblyFormat As FormatterA  
    Get  
    Set
```

```
public:  
property FormatterAssemblyStyle TypeNameAssemblyForm  
    FormatterAssemblyStyle get ();  
    void set (FormatterAssemblyStyle value);  
}
```

▣ **Value**

The type name assembly format.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ►

### TypeNameHandling

Gets or sets how type name writing and reading is handled by the serializer.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public TypeNameHandling TypeNameHandling { get; set;
```

```
Public Property TypeNameHandling As TypeNameHandling  
    Get  
    Set
```

```
public:  
property TypeNameHandling TypeNameHandling {  
    TypeNameHandling get ();  
    void set (TypeNameHandling value);  
}
```

#### Value

The type name handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonTextReader Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextReader](#) C#

Represents a reader that provides fast, non-cached, forward-only access to JSON text data.

## Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class JsonTextReader : JsonReader,
    IJsonLineInfo
```

```
Public Class JsonTextReader _
    Inherits JsonReader _
    Implements IJsonLineInfo
```

```
public ref class JsonTextReader : public JsonReader,
    IJsonLineInfo
```

## Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
			<input checked="" type="checkbox"/> Declare
			<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonTextReader(TextReader)</a>	Initializes a new instance of the <a href="#">JsonReader</a> class with the specific <a href="#">TextReader</a> .
	<a href="#">Close()</a>	Changes the state to closed.  (Overrides <a href="#">JsonReader.Close()</a> .)

	<a href="#">CloseInput</a>	Gets or sets a value indicating whether the underlying stream or <a href="#">TextReader</a> should be closed when the reader is closed.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">Culture</a>	Gets or sets the culture used when reading JSON. Defaults to <a href="#">InvariantCulture</a> .  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">CurrentState</a>	Gets the current reader state.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">DateParseHandling</a>	Get or set how date formatted strings, e.g. "VDate(1198908717056)V" and "2012-03-21T05:40Z", are parsed when reading JSON.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">DateTimeZoneHandling</a>	Get or set how <a href="#">DateTime</a> time zones are handled when reading JSON.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">Depth</a>	Gets the depth of the current token in the JSON document.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">Dispose(Boolean)</a>	Releases unmanaged and - optionally - managed resources.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .

		(Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other clear operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">HasLineInfo()</a>	Gets a value indicating whether the class can return line information.
	<a href="#">LineNumber</a>	Gets the current line number.
	<a href="#">LinePosition</a>	Gets the current line position.
	<a href="#">MaxDepth</a>	Gets or sets the maximum depth allowed when reading JSON. Reads past this depth will throw a <a href="#">JsonReaderException</a> .  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
		

	<a href="#">Path</a>	Gets the path of the current JSON token.  (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">QuoteChar</a>	Gets the quotation mark character used to enclose the value of a string.  (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">Read()</a>	Reads the next JSON token from the stream.  (Overrides <a href="#">JsonReader.Read().</a> )
	<a href="#">ReadAsBytes()</a>	Reads the next JSON token from the stream as a <b>Byte[]</b> .  (Overrides <a href="#">JsonReader.ReadAsBytes().</a> )
	<a href="#">ReadAsDateTime()</a>	Reads the next JSON token from the stream as a <a href="#">Nullable&lt;T&gt;</a> .  (Overrides <a href="#">JsonReader.ReadAsDateTime().</a> )
	<a href="#">ReadAsDateTimeOffset()</a>	Reads the next JSON token from the stream as a <a href="#">Nullable&lt;T&gt;</a> .  (Overrides <a href="#">JsonReader.ReadAsDateTimeOffset().</a> )
	<a href="#">ReadAsDecimal()</a>	Reads the next JSON token from the stream as a <a href="#">Nullable&lt;T&gt;</a> .  (Overrides <a href="#">JsonReader.ReadAsDecimal().</a> )
	<a href="#">ReadAsInt32()</a>	Reads the next JSON token from the stream as a <a href="#">Nullable&lt;T&gt;</a> .

		(Overrides <a href="#">JsonReader.ReadAsInt32().</a> )
	<a href="#">ReadAsString()</a>	Reads the next JSON token from the stream as a <a href="#">String</a> .  (Overrides <a href="#">JsonReader.ReadAsString().</a> )
	<a href="#">SetStateBasedOnCurrent()</a>	Sets the state based on current token type.  (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">SetToken(JsonToken)</a>	Sets the current token.  (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">SetToken(JsonToken, Object)</a>	Sets the current token and value.  (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">Skip()</a>	Skips the children of the current token.  (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">TokenType</a>	Gets the type of the current JSON token.  (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">Value</a>	Gets the text value of the current JSON token.  (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">ValueType</a>	Gets The Common Language Runtime (CLR) type for the current JSON token.

(Inherited from [JsonReader.](#))

## ▣ Inheritance Hierarchy

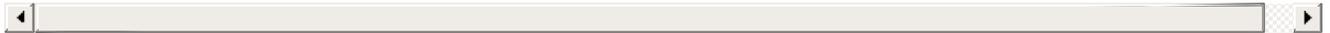
[Object](#)

└─ [JsonReader](#)

└─ **JsonTextReader**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Constructor (reader)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextReader](#) ►

### JsonTextReader(TextReader)

Initializes a new instance of the [JsonReader](#) class with the specified [TextReader](#).

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonTextReader(     <a href="#">TextReader</a> reader )</pre>		
	<pre>Public Sub New ( _     reader As <a href="#">TextReader</a> _ )</pre>	
		<pre>public: JsonTextReader(     <a href="#">TextReader</a>^ reader )</pre>

#### ▣ Parameters

**reader** ([TextReader](#))

The [TextReader](#) containing the XML data to read.

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextReader](#) ► **Close()**

C#

Changes the state to closed.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void Close()
```

```
Public Overrides Sub Close
```

```
public:  
virtual void Close() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextReader](#) ► **HasLineInfo()**

Gets a value indicating whether the class can return line information.

C#

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool HasLineInfo()
```

```
Public Function HasLineInfo As Boolean
```

```
public:  
virtual bool HasLineInfo() sealed
```

## ▣ Return Value

**true** if LineNumber and LinePosition can be provided; otherwise, **false**.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets the current line number.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">int</a> LineNumber { <a href="#">get</a>; }</pre>		
	<pre>Public ReadOnly Property LineNumber As <a href="#">Integer</a>     Get</pre>	
		<pre>public: virtual property <a href="#">int</a> LineNumber {     <a href="#">int</a> get () sealed; }</pre>

## ▣ Value

The current line number or 0 if no line information is available (for example, HasLineInfo returns false).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets the current line position.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">int</a> LinePosition { <a href="#">get</a>; }</pre>		
	<pre>Public ReadOnly Property LinePosition As <a href="#">Integer</a>     Get</pre>	
		<pre>public: virtual property <a href="#">int</a> LinePosition {     <a href="#">int</a> get () sealed; }</pre>

## ▣ Value

The current line position or 0 if no line information is available (for example, HasLineInfo returns false).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextReader](#) ► **Read()**

C#

Reads the next JSON token from the stream.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override bool Read()
```

```
Public Overrides Function Read As Boolean
```

```
public:  
virtual bool Read() override
```

## ▣ Return Value

true if the next token was read successfully; false if there are no more tokens to read.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextReader](#) ►

## ReadAsBytes()

Reads the next JSON token from the stream as a **Byte**.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override byte[] ReadAsBytes()
```

```
Public Overrides Function ReadAsBytes As Byte()
```

```
public:  
virtual array<unsigned char>^ ReadAsBytes() override
```

### ▣ Return Value

A **Byte** or a null reference if the next JSON token is null. This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextReader](#) ►

## ReadAsDateTime()

Reads the next JSON token from the stream as a [Nullable<T>](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<DateTime> ReadAsDateTime()
```

```
Public Overrides Function ReadAsDateTime As Nullable
```

```
public:  
virtual Nullable<DateTime> ReadAsDateTime() override
```

### ▣ Return Value

A [String](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextReader](#) ►

### ReadAsDateTimeOffset()

Reads the next JSON token from the stream as a [Nullable<T>](#).

C#

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<DateTimeOffset> ReadAsDateT
```

```
Public Overrides Function ReadAsDateTimeOffset As Nu
```

```
public:  
virtual Nullable<DateTimeOffset> ReadAsDateTimeOffse
```

#### ▣ Return Value

A [DateTimeOffset](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextReader](#) ►

## ReadAsDecimal()

Reads the next JSON token from the stream as a [Nullable<T>](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<decimal> ReadAsDecimal()
```

```
Public Overrides Function ReadAsDecimal As Nullable(
```

```
public:  
virtual Nullable<Decimal> ReadAsDecimal() override
```

### ▣ Return Value

A [Nullable<T>](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextReader](#) ► **ReadAsInt32()**

Reads the next JSON token from the stream as a [Nullable<T>](#).

C#

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<int> ReadAsInt32()
```

```
Public Overrides Function ReadAsInt32 As Nullable\(Of
```

```
public:  
virtual Nullable<int> ReadAsInt32() override
```

## ▣ Return Value

A [Nullable<T>](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextReader](#) ►

## ReadAsString()

Reads the next JSON token from the stream as a [String](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override string ReadAsString()
```

```
Public Overrides Function ReadAsString As String
```

```
public:  
virtual String^ ReadAsString() override
```

### ▣ Return Value

A [String](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonTextWriter Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonTextWriter**

Represents a writer that provides a fast, non-cached, forward-only way generating Json data.

## Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class JsonTextWriter : JsonWriter
```

```
Public Class JsonTextWriter _
    Inherits JsonWriter
```

```
public ref class JsonTextWriter : public JsonWriter
```

## Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
			<input checked="" type="checkbox"/> Declared
			<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonTextWriter(TextWriter)</a>	Creates an instance of the <a href="#">JsonWriter</a> class using <a href="#">TextWriter</a> .
	<a href="#">Close()</a>	Closes this stream and the underlying stream.  (Overrides <a href="#">JsonWriter.Close()</a> )
	<a href="#">CloseOutput</a>	Gets or sets a value indicating whether to close the underlying stream or not.

		<p>should be closed when the stream is closed.</p> <p>(Inherited from <a href="#">JsonWriter</a>.)</p>
	<a href="#">DateFormatHandling</a>	<p>Get or set how dates are formatted as text.</p> <p>(Inherited from <a href="#">JsonWriter</a>.)</p>
	<a href="#">DateTimeZoneHandling</a>	<p>Get or set how <a href="#">DateTime</a> values are handled when writing JSON.</p> <p>(Inherited from <a href="#">JsonWriter</a>.)</p>
	<a href="#">Equals(Object)</a>	<p>Determines whether the current <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">Finalize()</a>	<p>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is garbage collected.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">Flush()</a>	<p>Flushes whatever is in the buffer to the underlying streams and closes the underlying stream.</p> <p>(Overrides <a href="#">JsonWriter.Flush()</a>.)</p>
	<a href="#">Formatting</a>	<p>Indicates how JSON text is formatted.</p> <p>(Inherited from <a href="#">JsonWriter</a>.)</p>
	<a href="#">GetHashCode()</a>	<p>Serves as a hash function for a particular type.</p> <p>(Inherited from <a href="#">Object</a>.)</p>

	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Indentation</a>	Gets or sets how many lines to write for each level in the <a href="#">Formatting</a> is set to <a href="#">Formatting.Indentation</a> .
	<a href="#">IndentChar</a>	Gets or sets which character to use for indenting when <a href="#">Formatting</a> is set to <a href="#">Formatting.Indentation</a> .
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">Path</a>	Gets the path of the writer. (Inherited from <a href="#">JsonWriter</a> .)
	<a href="#">QuoteChar</a>	Gets or sets which character to use for quote attribute values.
	<a href="#">QuoteName</a>	Gets or sets a value indicating whether object names will be surrounded by quotes.
	<a href="#">Top</a>	Gets the top. (Inherited from <a href="#">JsonWriter</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current object.

		current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteComment(String)</a>	Writes out a comment  <b>C#</b> <hr/> <pre>/* . . . */</pre> containing the specified t  (Overrides <a href="#">JsonWriter.WriteComme</a>
	<a href="#">WriteEnd(JsonToken)</a>	Writes the specified end  (Overrides <a href="#">JsonWriter.WriteEnd(Jso</a>
	<a href="#">WriteEnd()</a>	Writes the end of the cur or array.  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteEndArray()</a>	Writes the end of an arra  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteEndConstructor()</a>	Writes the end construct  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteEndObject()</a>	Writes the end of a Json  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteIndent()</a>	Writes indent characters.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteIndentSpace()</a>	Writes an indent space.

		(Overrides <a href="#">JsonWriter.WriteIndentS</a>
	<a href="#">WriteNull()</a>	Writes a null value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WritePropertyName(String)</a>	Writes the property name name/value pair on a Jsc  (Overrides <a href="#">JsonWriter.WriteProperty</a>
	<a href="#">WriteRaw(String)</a>	Writes raw JSON.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteRawValue(String)</a>	Writes raw JSON where expected and updates th  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteStartArray()</a>	Writes the beginning of a  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteStartConstructor(String)</a>	Writes the start of a cons given name.  (Overrides <a href="#">JsonWriter.WriteStartCon</a>
	<a href="#">WriteStartObject()</a>	Writes the beginning of a  (Overrides <a href="#">JsonWriter.WriteStartObj</a>
	<a href="#">WriteState</a>	Gets the state of the writ  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteToken(JsonReader)</a>	Writes the current <a href="#">JsonR</a>

		(Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteUndefined()</a>	Writes an undefined value. (Overrides <a href="#">JsonWriter.WriteUndefined()</a> )
	<a href="#">WriteValue(String)</a>	Writes a <a href="#">String</a> value. (Overrides <a href="#">JsonWriter.WriteValue(String)</a> )
	<a href="#">WriteValue(Int32)</a>	Writes a <a href="#">Int32</a> value. (Overrides <a href="#">JsonWriter.WriteValue(Int32)</a> )
	<a href="#">WriteValue(UInt32)</a>	Writes a <a href="#">UInt32</a> value. (Overrides <a href="#">JsonWriter.WriteValue(UInt32)</a> )
	<a href="#">WriteValue(Int64)</a>	Writes a <a href="#">Int64</a> value. (Overrides <a href="#">JsonWriter.WriteValue(Int64)</a> )
	<a href="#">WriteValue(UInt64)</a>	Writes a <a href="#">UInt64</a> value. (Overrides <a href="#">JsonWriter.WriteValue(UInt64)</a> )
	<a href="#">WriteValue(Single)</a>	Writes a <a href="#">Single</a> value. (Overrides <a href="#">JsonWriter.WriteValue(Single)</a> )
	<a href="#">WriteValue(Double)</a>	Writes a <a href="#">Double</a> value. (Overrides <a href="#">JsonWriter.WriteValue(Double)</a> )
	<a href="#">WriteValue(Boolean)</a>	Writes a <a href="#">Boolean</a> value. (Overrides <a href="#">JsonWriter.WriteValue(Boolean)</a> )

 <a href="#">WriteValue(Int16)</a>	Writes a <a href="#">Int16</a> value.  (Overrides <a href="#">JsonWriter.W</a>
 <a href="#">WriteValue(UInt16)</a>	Writes a <a href="#">UInt16</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(U</a>
 <a href="#">WriteValue(Char)</a>	Writes a <a href="#">Char</a> value.  (Overrides <a href="#">JsonWriter.W</a>
 <a href="#">WriteValue(Byte)</a>	Writes a <a href="#">Byte</a> value.  (Overrides <a href="#">JsonWriter.W</a>
 <a href="#">WriteValue(SByte)</a>	Writes a <a href="#">SByte</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(S</a>
 <a href="#">WriteValue(Decimal)</a>	Writes a <a href="#">Decimal</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
 <a href="#">WriteValue(DateTime)</a>	Writes a <a href="#">DateTime</a> value  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
 <a href="#">WriteValue(Byte[])</a>	Writes a <b>Byte[]</b> value.  (Overrides <a href="#">JsonWriter.WriteValue(B</a>
 <a href="#">WriteValue(DateTimeOffset)</a>	Writes a <a href="#">DateTimeOffset</a>  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
 <a href="#">WriteValue(Guid)</a>	Writes a <a href="#">Guid</a> value.

		(Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(TimeSpan)</a>	Writes a <a href="#">TimeSpan</a> value  (Overrides <a href="#">JsonWriter.WriteValue(Ti</a>
	<a href="#">WriteValue(Uri)</a>	Writes a <a href="#">Uri</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(Nullable&lt;Int32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;UInt32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Int64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;UInt64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Single&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Double&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Boolean&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Int16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;UInt16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val

		(Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Char&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Byte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;SByte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Decimal&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;DateTime&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;DateTimeOffset&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Guid&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;TimeSpan&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Object)</a>	Writes a <a href="#">Object</a> value. An if the value cannot be wr JSON token.  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValueDelimiter()</a>	Writes the JSON value d  (Overrides <a href="#">JsonWriter.WriteValueDe</a>



[WriteWhitespace\(String\)](#)

Writes out the given whit

(Overrides

[JsonWriter.WriteWhitesp](#)

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonWriter](#)

└─ **JsonTextWriter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Constructor (TextWriter)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

### JsonTextWriter(TextWriter)

Creates an instance of the `JsonWriter` class using the specified [TextWriter](#).

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonTextWriter(     <a href="#">TextWriter</a> <i>textWriter</i> )</pre>		
	<pre>Public Sub New (      <i>textWriter</i> As <a href="#">TextWriter</a>  )</pre>	
		<pre>public: JsonTextWriter(     <a href="#">TextWriter</a>^ <i>textWriter</i> )</pre>

#### Parameters

##### *textWriter* ([TextWriter](#))

The `TextWriter` to write to.

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **Close()**

C#

Closes this stream and the underlying stream.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void Close()
```

```
Public Overrides Sub Close
```

```
public:  
virtual void Close() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **Flush()**

C#

Flushes whatever is in the buffer to the underlying streams and also flushes the underlying stream.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void Flush()
```

```
Public Overrides Sub Flush
```

```
public:  
virtual void Flush() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

C#

Gets or sets how many IndentChars to write for each level in the hierarchy when [Formatting](#) is set to `Formatting.Indented`.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public int Indentation { get; set; }
```

```
Public Property Indentation As Integer  
    Get  
    Set
```

```
public:  
property int Indentation {  
    int get ();  
    void set (int value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **IndentChar**

Gets or sets which character to use for indenting when [Formatting](#) is set to [Formatting.Indented](#).

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public char IndentChar { get; set; }
```

```
Public Property IndentChar As Char  
    Get  
    Set
```

```
public:  
property wchar_t IndentChar {  
    wchar_t get ();  
    void set (wchar_t value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **QuoteChar**

Gets or sets which character to use to quote attribute values.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public char QuoteChar { get; set; }
```

```
Public Property QuoteChar As Char  
    Get  
    Set
```

```
public:  
property wchar_t QuoteChar {  
    wchar_t get ();  
    void set (wchar_t value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **QuoteName**

C#

Gets or sets a value indicating whether object names will be surrounded with quotes.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool QuoteName { get; set; }
```

```
Public Property QuoteName As Boolean  
    Get  
    Set
```

```
public:  
property bool QuoteName {  
    bool get ();  
    void set (bool value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (text)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

### WriteComment(String)

C#

Writes out a comment

C#

 Cop

```
/* ... */
```

containing the specified text.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteComment(  
    string text  
)
```

```
Public Overrides Sub WriteComment ( _  
    text As String _  
)
```

```
public:  
virtual void WriteComment(  
    String^ text  
) override
```

#### ▣ Parameters

**text** ([String](#))

Text to place inside the comment.

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

## WriteEnd(JsonToken)

Writes the specified end token.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected override void WriteEnd(  
    JsonToken token  
)
```

```
Protected Overrides Sub WriteEnd ( _  
    token As JsonToken _  
)
```

```
protected:  
virtual void WriteEnd(  
    JsonToken token  
) override
```

### ▣ Parameters

***token*** ([JsonToken](#))

The end token to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **WriteIndent()**

C#

Writes indent characters.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected override void WriteIndent()
```

```
Protected Overrides Sub WriteIndent
```

```
protected:  
virtual void WriteIndent() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

### WriteIndentSpace()

Writes an indent space.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected override void WriteIndentSpace()
```

```
Protected Overrides Sub WriteIndentSpace
```

```
protected:  
virtual void WriteIndentSpace() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **WriteNull()**

C#

Writes a null value.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteNull()
```

```
Public Overrides Sub WriteNull
```

```
public:  
virtual void WriteNull() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Method (name)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

### WritePropertyName(String)

Writes the property name of a name/value pair on a Json object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WritePropertyName(  
    string name  
)
```

```
Public Overrides Sub WritePropertyName ( _  
    name As String _  
)
```

```
public:  
virtual void WritePropertyName(  
    String^ name  
) override
```

#### ▣ Parameters

**name** ([String](#))

The name of the property.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

## WriteRaw(String)

Writes raw JSON.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteRaw(  
    string json  
)
```

```
Public Overrides Sub WriteRaw ( _  
    json As String _  
)
```

```
public:  
virtual void WriteRaw(  
    String^ json  
) override
```

### ▣ Parameters

*json* (**String**)

The raw JSON to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

## WriteStartArray()

Writes the beginning of a Json array.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteStartArray()
```

```
Public Overrides Sub WriteStartArray
```

```
public:  
virtual void WriteStartArray() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**Method (name)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►**WriteStartConstructor(String)**

Writes the start of a constructor with the given name.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteStartConstructor(  
    string name  
)
```

```
Public Overrides Sub WriteStartConstructor ( _  
    name As String _  
)
```

```
public:  
virtual void WriteStartConstructor(  
    String^ name  
) override
```

**Parameters*****name* (String)**

The name of the constructor.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

## WriteStartObject()

Writes the beginning of a Json object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteStartObject()
```

```
Public Overrides Sub WriteStartObject
```

```
public:  
virtual void WriteStartObject() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►**WriteUndefined()**

Writes an undefined value.

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteUndefined()
```

```
Public Overrides Sub WriteUndefined
```

```
public:  
virtual void WriteUndefined() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

C#

## Members

Icon	Member	Description
	<a href="#">WriteValue(String)</a>	Writes a <a href="#">String</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(S</a>
	<a href="#">WriteValue(Int32)</a>	Writes a <a href="#">Int32</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(In</a>
	<a href="#">WriteValue(UInt32)</a>	Writes a <a href="#">UInt32</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(U</a>
	<a href="#">WriteValue(Int64)</a>	Writes a <a href="#">Int64</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(In</a>
	<a href="#">WriteValue(UInt64)</a>	Writes a <a href="#">UInt64</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(U</a>
	<a href="#">WriteValue(Single)</a>	Writes a <a href="#">Single</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(S</a>
	<a href="#">WriteValue(Double)</a>	Writes a <a href="#">Double</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(Boolean)</a>	Writes a <a href="#">Boolean</a> value.

		(Overrides <a href="#">JsonWriter.WriteValue(B</a>
	<a href="#">WriteValue(Int16)</a>	Writes a <a href="#">Int16</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(In</a>
	<a href="#">WriteValue(UInt16)</a>	Writes a <a href="#">UInt16</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(U</a>
	<a href="#">WriteValue(Char)</a>	Writes a <a href="#">Char</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(C</a>
	<a href="#">WriteValue(Byte)</a>	Writes a <a href="#">Byte</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(SByte)</a>	Writes a <a href="#">SByte</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(S</a>
	<a href="#">WriteValue(Decimal)</a>	Writes a <a href="#">Decimal</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(DateTime)</a>	Writes a <a href="#">DateTime</a> value  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(Byte[])</a>	Writes a <b>Byte[]</b> value.  (Overrides <a href="#">JsonWriter.WriteValue(B</a>
		

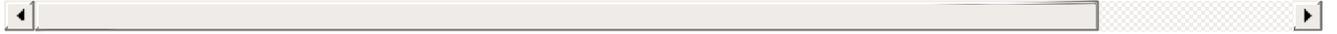
	<a href="#">WriteValue(DateTimeOffset)</a>	Writes a <a href="#">DateTimeOffset</a>  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(Guid)</a>	Writes a <a href="#">Guid</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(G</a>
	<a href="#">WriteValue(TimeSpan)</a>	Writes a <a href="#">TimeSpan</a> value  (Overrides <a href="#">JsonWriter.WriteValue(Ti</a>
	<a href="#">WriteValue(Uri)</a>	Writes a <a href="#">Uri</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(Nullable&lt;Int32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;UInt32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Int64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;UInt64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Single&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Double&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Boolean&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val

	<a href="#">WriteValue(Nullable&lt;Int16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;UInt16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Char&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Byte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;SByte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Decimal&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;DateTime&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;DateTimeOffset&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Guid&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;TimeSpan&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Object)</a>	Writes a <a href="#">Object</a> value. An exception is raised if the value cannot be written.

single JSON token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)



## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

### WriteValue(Boolean)

Writes a [Boolean](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    bool value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Boolean _  
)
```

```
public:  
virtual void WriteValue(  
    bool value  
) override
```

#### ▣ Parameters

##### **value** ([Boolean](#))

The [Boolean](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►**WriteValue(Byte)**Writes a [Byte](#) value.**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    byte value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Byte _  
)
```

```
public:  
virtual void WriteValue(  
    unsigned char value  
) override
```

**▣ Parameters****value ([Byte](#))**The [Byte](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►**WriteValue(Byte[])**

C#

Writes a **Byte[]** value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    byte[] value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Byte() _  
)
```

```
public:  
virtual void WriteValue(  
    array<unsigned char>^ value  
) override
```

▣ **Parameters****value** ([Byte\[\]](#))The **Byte[]** value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

### WriteValue(Char)

Writes a [Char](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    char value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Char _  
)
```

```
public:  
virtual void WriteValue(  
    wchar\_t value  
) override
```

#### ▣ Parameters

##### **value** ([Char](#))

The [Char](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►**WriteValue(DateTime)**

Writes a [DateTime](#) value.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    DateTime value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As DateTime _  
)
```

```
public:  
virtual void WriteValue(  
    DateTime value  
) override
```

**Parameters****value ([DateTime](#))**

The [DateTime](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►**WriteValue(DateTimeOffset)**Writes a [DateTimeOffset](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    DateTimeOffset value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As DateTimeOffset _  
)
```

```
public:  
virtual void WriteValue(  
    DateTimeOffset value  
) override
```

▣ **Parameters****value ([DateTimeOffset](#))**The [DateTimeOffset](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

### WriteValue(Decimal)

Writes a [Decimal](#) value.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override void WriteValue(     <a href="#">decimal</a> value )</pre>		
	<pre>Public Overrides Sub WriteValue ( _     value As <a href="#">Decimal</a> _ )</pre>	
		<pre>public: virtual void WriteValue(     <a href="#">Decimal</a> value ) override</pre>

#### ▣ Parameters

##### **value** ([Decimal](#))

The [Decimal](#) value to write.

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

### WriteValue(Double)

Writes a [Double](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    double value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Double _  
)
```

```
public:  
virtual void WriteValue(  
    double value  
) override
```

#### ▣ Parameters

##### **value** ([Double](#))

The [Double](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►**WriteValue(Guid)**Writes a [Guid](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    Guid value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Guid _  
)
```

```
public:  
virtual void WriteValue(  
    Guid value  
) override
```

▣ **Parameters****value ([Guid](#))**The [Guid](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►**WriteValue(Int16)**

Writes a [Int16](#) value.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    short value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Short _  
)
```

```
public:  
virtual void WriteValue(  
    short value  
) override
```

**Parameters****value ([Int16](#))**

The [Int16](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►**WriteValue(Int32)**

Writes a [Int32](#) value.

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    int value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Integer _  
)
```

```
public:  
virtual void WriteValue(  
    int value  
) override
```

**▣ Parameters****value ([Int32](#))**

The [Int32](#) value to write.

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►**WriteValue(Int64)**

Writes a [Int64](#) value.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    long value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Long _  
)
```

```
public:  
virtual void WriteValue(  
    long long value  
) override
```

**Parameters****value ([Int64](#))**

The [Int64](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►**WriteValue(SByte)**Writes a [SByte](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    sbyte value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As SByte _  
)
```

```
public:  
virtual void WriteValue(  
    signed char value  
) override
```

▣ **Parameters****value (SByte)**The [SByte](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

### WriteValue(Single)

Writes a [Single](#) value.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override void WriteValue(     float value )</pre>		
	<pre>Public Overrides Sub WriteValue ( _     value As Single _ )</pre>	
		<pre>public: virtual void WriteValue(     float value ) override</pre>

#### ▣ Parameters

##### **value** ([Single](#))

The [Single](#) value to write.

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►**WriteValue(String)**Writes a [String](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    string value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As String _  
)
```

```
public:  
virtual void WriteValue(  
    String^ value  
) override
```

▣ **Parameters****value ([String](#))**The [String](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►**WriteValue(TimeSpan)**Writes a [TimeSpan](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    TimeSpan value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As TimeSpan _  
)
```

```
public:  
virtual void WriteValue(  
    TimeSpan value  
) override
```

▣ **Parameters****value ([TimeSpan](#))**The [TimeSpan](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

### WriteValue(UInt16)

Writes a [UInt16](#) value.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override void WriteValue(     ushort value )</pre>	<pre>Public Overrides Sub WriteValue ( _     value As UShort _ )</pre>	<pre>public: virtual void WriteValue(     unsigned short value ) override</pre>

#### ▣ Parameters

##### **value** ([UInt16](#))

The [UInt16](#) value to write.

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►**WriteValue(UInt32)**

Writes a [UInt32](#) value.

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    uint value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As UInteger _  
)
```

```
public:  
virtual void WriteValue(  
    unsigned int value  
) override
```

**▣ Parameters****value ([UInt32](#))**

The [UInt32](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

### WriteValue(UInt64)

C#

Writes a [UInt64](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    ulong value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As ULong _  
)
```

```
public:  
virtual void WriteValue(  
    unsigned long long value  
) override
```

#### ▣ Parameters

**value** ([UInt64](#))

The [UInt64](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **WriteValue(Uri)**

C#

Writes a [Uri](#) value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    Uri value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Uri _  
)
```

```
public:  
virtual void WriteValue(  
    Uri^ value  
) override
```

### ▣ Parameters

#### **value (Uri)**

The [Uri](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

### WriteValueDelimiter()

Writes the JSON value delimiter.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected override void WriteValueDelimiter()
```

```
Protected Overrides Sub WriteValueDelimiter
```

```
protected:  
virtual void WriteValueDelimiter() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# WriteWhitespace Method (ws)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►

## WriteWhitespace(String)

Writes out the given white space.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteWhitespace(  
    string ws  
)
```

```
Public Overrides Sub WriteWhitespace ( _  
    ws As String _  
)
```

```
public:  
virtual void WriteWhitespace(  
    String^ ws  
) override
```

### ▣ Parameters

**ws** ([String](#))

The string of white space characters.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Specifies the type of Json token.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum JsonToken
```

```
Public Enumeration JsonToken
```

```
public enum class JsonToken
```

## ▣ Members

Member	Description
<b>None</b>	This is returned by the <a href="#">JsonReader</a> if a <a href="#">Read()</a> method has not been called.
<b>StartObject</b>	An object start token.
<b>StartArray</b>	An array start token.
<b>StartConstructor</b>	A constructor start token.
<b>PropertyName</b>	An object property name.
<b>Comment</b>	A comment.
<b>Raw</b>	Raw JSON.
<b>Integer</b>	An integer.
<b>Float</b>	A float.
<b>String</b>	A string.
<b>Boolean</b>	A boolean.
<b>Null</b>	A null token.

<b>Undefined</b>	An undefined token.
<b>EndObject</b>	An object end token.
<b>EndArray</b>	An array end token.
<b>EndConstructor</b>	A constructor end token.
<b>Date</b>	A Date.
<b>Bytes</b>	Byte data.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# JsonValidatingReader

## Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonValidatingReader**

C#

Represents a reader that provides [JsonSchema](#) validation.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonValidatingReader : JsonReader,  
    IJsonLineInfo
```

```
Public Class JsonValidatingReader _  
    Inherits JsonReader _  
    Implements IJsonLineInfo
```

```
public ref class JsonValidatingReader : public JsonR  
    IJsonLineInfo
```

### Members

All Members

Constructors

Methods

Properties

Events

Public

Protected

Instance

Static

Declared

Inherited

Icon	Member	Description
	<a href="#">JsonValidatingReader(JsonReader)</a>	Initializes a new instance of <b>JsonValidatingReader</b> class and validates the content returned by the given <a href="#">JsonReader</a> .
	<a href="#">Close()</a>	Changes the <a href="#">JsonReader.S</a> property to Closed.  (Inherited from <a href="#">JsonReader</a> .)

	<a href="#">CloseInput</a>	Gets or sets a value indicating whether the underlying stream or <a href="#">TextReader</a> should be closed when the reader is closed.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">Culture</a>	Gets or sets the culture used when reading JSON. Defaults to <a href="#">InvariantCulture</a> .  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">CurrentState</a>	Gets the current reader state.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">DateParseHandling</a>	Get or set how date formats, such as "VDate(11989087170562012-03-21T05:40Z)", are parsed when reading JSON.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">DateTimeZoneHandling</a>	Get or set how <a href="#">DateTime</a> values are handled when reading JSON.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">Depth</a>	Gets the depth of the current token in the JSON document.  (Overrides <a href="#">JsonReader.Depth</a> .)
	<a href="#">Dispose(Boolean)</a>	Releases unmanaged and - optionally - managed resources.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current object.

		(Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt resources and perform other operations before the <a href="#">Object</a> reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current object.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MaxDepth</a>	Gets or sets the maximum depth allowed when reading JSON. If the depth past this depth will throw a <a href="#">JsonReaderException</a> .  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Path</a>	Gets the path of the current token.  (Overrides <a href="#">JsonReader.Path</a> .)
	<a href="#">QuoteChar</a>	Gets the quotation mark character used to enclose the value of the current token.  (Overrides <a href="#">JsonReader.QuoteChar</a> .)
	<a href="#">Read()</a>	Reads the next JSON token from the stream.

		(Overrides <a href="#">JsonReader.Read</a>
	<a href="#">ReadAsBytes()</a>	<p>Reads the next JSON token stream as a <b>Byte[]</b>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsBytes()</a>)</p>
	<a href="#">ReadAsDateTime()</a>	<p>Reads the next JSON token stream as a <a href="#">Nullable&lt;T&gt;</a>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsDateTir</a></p>
	<a href="#">ReadAsDateTimeOffset()</a>	<p>Reads the next JSON token stream as a <a href="#">Nullable&lt;T&gt;</a>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsDateTir</a></p>
	<a href="#">ReadAsDecimal()</a>	<p>Reads the next JSON token stream as a <a href="#">Nullable&lt;T&gt;</a>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsDecima</a></p>
	<a href="#">ReadAsInt32()</a>	<p>Reads the next JSON token stream as a <a href="#">Nullable&lt;T&gt;</a>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsInt32()</a>).</p>
	<a href="#">ReadAsString()</a>	<p>Reads the next JSON token stream as a <a href="#">String</a>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsString()</a></p>
	<a href="#">Reader</a>	Gets the <a href="#">JsonReader</a> used t this <b>JsonValidatingReader</b>

	<a href="#">Schema</a>	Gets or sets the schema.
	<a href="#">SetStateBasedOnCurrent()</a>	Sets the state based on current type.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">SetToken(JsonToken)</a>	Sets the current token.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">SetToken(JsonToken, Object)</a>	Sets the current token and value.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">Skip()</a>	Skips the children of the current token.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">TokenType</a>	Gets the type of the current token.  (Overrides <a href="#">JsonReader.TokenType</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ValidationEventHandler</a>	Sets an event handler for receiving schema validation errors.
	<a href="#">Value</a>	Gets the text value of the current token.  (Overrides <a href="#">JsonReader.Value</a> .)
	<a href="#">ValueType</a>	Gets the Common Language Runtime (CLR) type for the current <a href="#">JsonToken</a> .

(Overrides [JsonReader.Valu](#)

## ▣ Inheritance Hierarchy

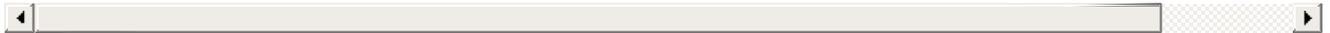
[Object](#)

└─ [JsonReader](#)

└─ **JsonValidatingReader**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



# JsonValidatingReader

## Constructor (reader)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ►

### JsonValidatingReader(JsonReader)

Initializes a new instance of the [JsonValidatingReader](#) class that validates the content returned from the given [JsonReader](#).

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonValidatingReader(     <a href="#">JsonReader</a> reader )</pre>		
	<pre>Public Sub New (      reader As <a href="#">JsonReader</a>  )</pre>	
		<pre>public: JsonValidatingReader(     <a href="#">JsonReader</a>^ reader )</pre>

#### Parameters

**reader** ([JsonReader](#))

The [JsonReader](#) to read from while validating.

C#

Gets the depth of the current token in the JSON document.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override int Depth { get; }
```

```
Public Overrides ReadOnly Property Depth As Integer  
    Get
```

```
public:  
virtual property int Depth {  
    int get () override;  
}
```

## ▣ Value

The depth of the current token in the JSON document.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets the path of the current JSON token.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override string Path { get; }</pre>		
	<pre>Public Overrides ReadOnly Property Path As String     Get</pre>	
		<pre>public: virtual property String^ Path {     String^ get () override; }</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ►

## QuoteChar

C#

Gets the quotation mark character used to enclose the value of a string.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public override char QuoteChar { get; protected internal set;
```

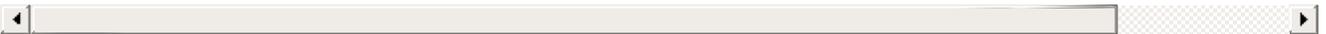
```
Public Overrides Property QuoteChar As Char  
    Get  
    Protected Friend Set
```

```
public:  
virtual property wchar\_t QuoteChar {  
    wchar\_t get () override;  
    protected public: void set (wchar\_t value) override;  
}
```

### Value

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ► **Read()**

C# ▼

Reads the next JSON token from the stream.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override bool Read()
```

```
Public Overrides Function Read As Boolean
```

```
public:  
virtual bool Read() override
```

## ▣ Return Value

true if the next token was read successfully; false if there are no more tokens to read.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ►

## ReadAsBytes()

Reads the next JSON token from the stream as a **Byte[]**.

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override byte[] ReadAsBytes()
```

```
Public Overrides Function ReadAsBytes As Byte()
```

```
public:  
virtual array<unsigned char>^ ReadAsBytes() override
```

### ▣ Return Value

A **Byte[]** or a null reference if the next JSON token is null.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ►

## ReadAsDateTime()

Reads the next JSON token from the stream as a [Nullable<T>](#).

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<DateTime> ReadAsDateTime()
```

```
Public Overrides Function ReadAsDateTime As Nullable
```

```
public:  
virtual Nullable<DateTime> ReadAsDateTime() override
```

### ▣ Return Value

A [String](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ►

### ReadAsDateTimeOffset()

Reads the next JSON token from the stream as a [Nullable<T>](#).

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<DateTimeOffset> ReadAsDateT
```

```
Public Overrides Function ReadAsDateTimeOffset As Nu
```

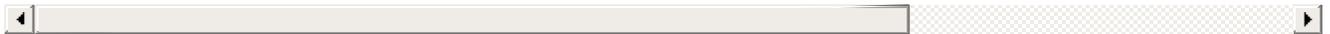
```
public:  
virtual Nullable<DateTimeOffset> ReadAsDateTimeOffse
```

#### ▣ Return Value

A [Nullable<T>](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ►

## ReadAsDecimal()

Reads the next JSON token from the stream as a [Nullable<T>](#).

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<decimal> ReadAsDecimal()
```

```
Public Overrides Function ReadAsDecimal As Nullable(
```

```
public:  
virtual Nullable<Decimal> ReadAsDecimal() override
```

### Return Value

A [Nullable<T>](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ►

## ReadAsInt32()

Reads the next JSON token from the stream as a [Nullable<T>](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<int> ReadAsInt32()
```

```
Public Overrides Function ReadAsInt32 As Nullable\(Of
```

```
public:  
virtual Nullable<int> ReadAsInt32() override
```

### ▣ Return Value

A [Nullable<T>](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ►

## ReadAsString()

Reads the next JSON token from the stream as a [String](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override string ReadAsString()
```

```
Public Overrides Function ReadAsString As String
```

```
public:  
virtual String^ ReadAsString() override
```

### ▣ Return Value

A [String](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ► **Reader**

C#

Gets the [JsonReader](#) used to construct this [JsonValidatingReader](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonReader Reader { get; }
```

```
Public ReadOnly Property Reader As JsonReader  
    Get
```

```
public:  
property JsonReader^ Reader {  
    JsonReader^ get ();  
}
```

## ▣ Value

The [JsonReader](#) specified in the constructor.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets the schema.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">JsonSchema</a> Schema { get; set; }</pre>		
<pre>Public Property Schema As <a href="#">JsonSchema</a>     Get     Set</pre>		
<pre>public: property <a href="#">JsonSchema</a>^ Schema {     <a href="#">JsonSchema</a>^ get ();     void set (<a href="#">JsonSchema</a>^ <i>value</i>); }</pre>		

## ▣ Value

The schema.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ►

## TokenType

C#

Gets the type of the current Json token.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public override JsonToken TokenType { get; }
```

```
Public Overrides ReadOnly Property TokenType As JsonToken  
Get
```

```
public:  
virtual property JsonToken TokenType {  
    JsonToken get () override;  
}
```

### Value

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Event

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ►

### ValidationEventHandler

Sets an event handler for receiving schema validation errors.

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public event <a href="#">ValidationEventHandler</a> ValidationEventH</pre>		
	<pre>Public Event ValidationEventHandler As <a href="#">ValidationEve</a></pre>	
		<pre>public:     event <a href="#">ValidationEventHandler</a>^ ValidationEventHandle         void add (<a href="#">ValidationEventHandler</a>^ value);         void remove (<a href="#">ValidationEventHandler</a>^ value); }</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)



Gets the text value of the current Json token.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override <a href="#">Object</a> Value { get; }</pre>		
	<pre>Public Overrides ReadOnly Property Value As <a href="#">Object</a> Get</pre>	
		<pre>public: virtual property <a href="#">Object</a>^ Value {     <a href="#">Object</a>^ get () override; }</pre>

## ▣ Value

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ►

## ValueType

Gets the Common Language Runtime (CLR) type for the current Json token.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Type ValueType { get; }
```

```
Public Overrides ReadOnly Property ValueType As Type  
    Get
```

```
public:  
virtual property Type^ ValueType {  
    Type^ get () override;  
}
```

### Value

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Represents a writer that provides a fast, non-cached, forward-only way generating Json data.

## Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public abstract class JsonWriter : IDisposable
```

```
Public MustInherit Class JsonWriter _
    Implements IDisposable
```

```
public ref class JsonWriter abstract : IDisposable
```

## Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
			<input checked="" type="checkbox"/> Declare
			<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonWriter()</a>	Creates an instance of the <b>JsonWriter</b> class
	<a href="#">Close()</a>	Closes this stream and the underlying stream.
	<a href="#">CloseOutput</a>	Gets or sets a value indicating whether the underlying stream or

		<a href="#">TextWriter</a> should be closed when the writer is closed.
	<a href="#">DateFormatHandling</a>	Get or set how dates are written to JSON text.
	<a href="#">DateTimeZoneHandling</a>	Get or set how <a href="#">DateTime</a> time zones are handling when writing JSON.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">Flush()</a>	Flushes whatever is in the buffer to the underlying streams and also flushes the underlying stream.
	<a href="#">Formatting</a>	Indicates how JSON text output is formatted.

	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Path</a>	Gets the path of the writer.
	<a href="#">Top</a>	Gets the top.
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteComment(String)</a>	Writes out a comment  <b>C#</b>  Copy <pre>/* . . . */</pre> containing the specified text.

	<a href="#"><u>WriteEnd()</u></a>	Writes the end of the current Json object or array.
	<a href="#"><u>WriteEnd(JsonToken)</u></a>	Writes the specified end token.
	<a href="#"><u>WriteEndArray()</u></a>	Writes the end of an array.
	<a href="#"><u>WriteEndConstructor()</u></a>	Writes the end constructor.
	<a href="#"><u>WriteEndObject()</u></a>	Writes the end of a Json object.
	<a href="#"><u>WriteIndent()</u></a>	Writes indent characters.
	<a href="#"><u>WriteIndentSpace()</u></a>	Writes an indent space.
	<a href="#"><u>WriteNull()</u></a>	Writes a null value.
	<a href="#"><u>WritePropertyName(String)</u></a>	Writes the property name of a name/value pair on a Json object.

 <a href="#">WriteRaw(String)</a>	Writes raw JSON without changing the writer's state.
 <a href="#">WriteRawValue(String)</a>	Writes raw JSON where value is expected and updates the writer's state
 <a href="#">WriteStartArray()</a>	Writes the beginning of a Json array.
 <a href="#">WriteStartConstructor(String)</a>	Writes the start of a constructor with the given name.
 <a href="#">WriteStartObject()</a>	Writes the beginning of a Json object.
 <a href="#">WriteState</a>	Gets the state of the writer.
 <a href="#">WriteToken(JsonReader)</a>	Writes the current <a href="#">JsonReader</a> token.
 <a href="#">WriteUndefined()</a>	Writes an undefined value.

 <a href="#">WriteValue(String)</a>	Writes a <a href="#">String</a> value.
 <a href="#">WriteValue(Int32)</a>	Writes a <a href="#">Int32</a> value.
 <a href="#">WriteValue(UInt32)</a>	Writes a <a href="#">UInt32</a> value.
 <a href="#">WriteValue(Int64)</a>	Writes a <a href="#">Int64</a> value.
 <a href="#">WriteValue(UInt64)</a>	Writes a <a href="#">UInt64</a> value.
 <a href="#">WriteValue(Single)</a>	Writes a <a href="#">Single</a> value.
 <a href="#">WriteValue(Double)</a>	Writes a <a href="#">Double</a> value.
 <a href="#">WriteValue(Boolean)</a>	Writes a <a href="#">Boolean</a> value.
 <a href="#">WriteValue(Int16)</a>	Writes a <a href="#">Int16</a> value.
 <a href="#">WriteValue(UInt16)</a>	Writes a <a href="#">UInt16</a> value.
 <a href="#">WriteValue(Char)</a>	Writes a <a href="#">Char</a> value.
 <a href="#">WriteValue(Byte)</a>	Writes a <a href="#">Byte</a> value.

	<a href="#">WriteValue(SByte)</a>	Writes a <a href="#">SByte</a> value.
	<a href="#">WriteValue(Decimal)</a>	Writes a <a href="#">Decimal</a> value.
	<a href="#">WriteValue(DateTime)</a>	Writes a <a href="#">DateTime</a> value
	<a href="#">WriteValue(DateTimeOffset)</a>	Writes a <a href="#">DateTimeOffset</a> value.
	<a href="#">WriteValue(Guid)</a>	Writes a <a href="#">Guid</a> value.
	<a href="#">WriteValue(TimeSpan)</a>	Writes a <a href="#">TimeSpan</a> value
	<a href="#">WriteValue(Nullable&lt;Int32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;UInt32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;Int64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;UInt64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a>

		value.
	<a href="#">WriteValue(Nullable&lt;Single&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;Double&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;Boolean&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;Int16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;UInt16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;Char&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;Byte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;SByte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.

	<a href="#">WriteValue(Nullable&lt;Decimal&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;DateTime&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;DateTimeOffset&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;Guid&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;TimeSpan&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Byte[])</a>	Writes a <b>Byte[]</b> value.
	<a href="#">WriteValue(Uri)</a>	Writes a <a href="#">Uri</a> value.
	<a href="#">WriteValue(Object)</a>	Writes a <a href="#">Object</a> value. An error will be raised if the value cannot be written as a single JSON token.
		

	<a href="#">WriteValueDelimiter()</a>	Writes the JSON value delimiter.
	<a href="#">WriteWhitespace(String)</a>	Writes out the given white space.

## ▣ Inheritance Hierarchy

[Object](#)

└─ **JsonWriter**

└─ [BsonWriter](#)

└─ [JsonTextWriter](#)

└─ [JTokenWriter](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **JsonWriter()**

C#

Creates an instance of the `JsonWriter` class.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected JsonWriter()
```

```
Protected Sub New
```

```
protected:  
JsonWriter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **Close()**

C#

Closes this stream and the underlying stream.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void Close()
```

```
Public Overridable Sub Close
```

```
public:  
virtual void Close()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **CloseOutput**

C#

Gets or sets a value indicating whether the underlying stream or [TextReader](#) should be closed when the writer is closed.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool CloseOutput { get; set; }
```

```
Public Property CloseOutput As Boolean  
    Get  
    Set
```

```
public:  
property bool CloseOutput {  
    bool get ();  
    void set (bool value);  
}
```

## ▣ Value

true to close the underlying stream or [TextReader](#) when the writer is closed; otherwise false. The default is true.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►

### DateFormatHandling

Get or set how dates are written to JSON text.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public DateFormatHandling DateFormatHandling { get;
```

```
Public Property DateFormatHandling As DateFormatHand  
    Get  
    Set
```

```
public:  
property DateFormatHandling DateFormatHandling {  
    DateFormatHandling get ();  
    void set (DateFormatHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►

### DateTimeZoneHandling

Get or set how [DateTime](#) time zones are handling when writing JSON.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public DateTimeZoneHandling DateTimeZoneHandling { g
```

```
Public Property DateTimeZoneHandling As DateTimeZone  
    Get  
    Set
```

```
public:  
property DateTimeZoneHandling DateTimeZoneHandling {  
    DateTimeZoneHandling get ();  
    void set (DateTimeZoneHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **Flush()**

C#

Flushes whatever is in the buffer to the underlying streams and also flushes the underlying stream.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract void Flush()
```

```
Public MustOverride Sub Flush
```

```
public:  
virtual void Flush() abstract
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Indicates how JSON text output is formatted.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">Formatting</a> Formatting { <a href="#">get</a>; <a href="#">set</a>; }</pre>		
<pre>Public Property Formatting As <a href="#">Formatting</a>     Get     Set</pre>		
<pre>public: property <a href="#">Formatting</a> Formatting {     <a href="#">Formatting</a> <a href="#">get</a> ();     void <a href="#">set</a> (<a href="#">Formatting</a> <i>value</i>); }</pre>		

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets the path of the writer.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Path { get; }
```

```
Public ReadOnly Property Path As String  
    Get
```

```
public:  
property String^ Path {  
    String^ get ();  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets the top.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected internal int Top { get; }
```

```
Protected Friend ReadOnly Property Top As Integer  
    Get
```

```
protected public:  
property int Top {  
    int get ();  
}
```

## ▣ Value

The top.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (text)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►

### WriteComment(String)

C#

Writes out a comment

C#

 Cop

```
/* ... */
```

containing the specified text.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteComment(  
    string text  
)
```

```
Public Overridable Sub WriteComment ( _  
    text As String _  
)
```

```
public:  
virtual void WriteComment(  
    String^ text  
)
```

#### ▣ Parameters

**text** ([String](#))

Text to place inside the comment.

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteEnd()**

C# ▼

## Members

Icon	Member	Description
	<a href="#">WriteEnd()</a>	Writes the end of the current Json object or array.
	<a href="#">WriteEnd(JsonToken)</a>	Writes the specified end token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteEnd()**

Writes the end of the current Json object or array.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public virtual void WriteEnd()</pre>		
	<pre>Public Overridable Sub WriteEnd</pre>	
		<pre>public: virtual void WriteEnd()</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►

## WriteEnd(JsonToken)

Writes the specified end token.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual void WriteEnd(  
    JsonToken token  
)
```

```
Protected Overridable Sub WriteEnd ( _  
    token As JsonToken _  
)
```

```
protected:  
virtual void WriteEnd(  
    JsonToken token  
)
```

### ▣ Parameters

#### ***token*** ([JsonToken](#))

The end token to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteEndArray()**

C#

Writes the end of an array.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteEndArray()
```

```
Public Overridable Sub WriteEndArray
```

```
public:  
virtual void WriteEndArray()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**Method**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteEndConstructor()**

Writes the end constructor.

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteEndConstructor()
```

```
Public Overridable Sub WriteEndConstructor
```

```
public:  
virtual void WriteEndConstructor()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteEndObject()**

C#

Writes the end of a Json object.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteEndObject()
```

```
Public Overridable Sub WriteEndObject
```

```
public:  
virtual void WriteEndObject()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteIndent()**

C#

Writes indent characters.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual void WriteIndent()
```

```
Protected Overridable Sub WriteIndent
```

```
protected:  
virtual void WriteIndent()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteIndentSpace()**

C# ▼

Writes an indent space.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual void WriteIndentSpace()
```

```
Protected Overridable Sub WriteIndentSpace
```

```
protected:  
virtual void WriteIndentSpace()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteNull()**

C#

Writes a null value.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteNull()
```

```
Public Overridable Sub WriteNull
```

```
public:  
virtual void WriteNull()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Method (name)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►

### WritePropertyName(String)

Writes the property name of a name/value pair on a Json object.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public virtual void WritePropertyName(     <u>string</u> name )</pre>	<pre>Public Overridable Sub WritePropertyName (      name As <u>String</u>  )</pre>	<pre>public: virtual void WritePropertyName(     <u>String</u>^ name )</pre>

#### ▣ Parameters

##### *name* ([String](#))

The name of the property.

Writes raw JSON without changing the writer's state.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public virtual void WriteRaw(     <u>string</u> json )</pre>	<pre>Public Overridable Sub WriteRaw ( _     json As <u>String</u> _ )</pre>	<pre>public: virtual void WriteRaw(     <u>String</u>^ json )</pre>

## ▣ Parameters

### *json* (**String**)

The raw JSON to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (json)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►

### WriteRawValue(String)

Writes raw JSON where a value is expected and updates the writer's state. C# ▼

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteRawValue(  
    string json  
)
```

```
Public Overridable Sub WriteRawValue ( _  
    json As String _  
)
```

```
public:  
virtual void WriteRawValue(  
    String^ json  
)
```

#### ▣ Parameters

##### *json* (**String**)

The raw JSON to write.

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteStartArray()**

C#

Writes the beginning of a Json array.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteStartArray()
```

```
Public Overridable Sub WriteStartArray
```

```
public:  
virtual void WriteStartArray()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**Method (name)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteStartConstructor(String)**

Writes the start of a constructor with the given name.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteStartConstructor(  
    string name  
)
```

```
Public Overridable Sub WriteStartConstructor ( _  
    name As String _  
)
```

```
public:  
virtual void WriteStartConstructor(  
    String^ name  
)
```

**Parameters*****name* ([String](#))**

The name of the constructor.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Writes the beginning of a Json object.

## ▣ Declaration Syntax

 C# Visual Basic Visual C++

```
public virtual void WriteStartObject()
```

```
Public Overridable Sub WriteStartObject
```

```
public:  
virtual void WriteStartObject()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets the state of the writer.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public WriteState WriteState { get; }
```

```
Public ReadOnly Property WriteState As WriteState  
    Get
```

```
public:  
property WriteState WriteState {  
    WriteState get ();  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (reader)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►

### WriteToken(JsonReader)

Writes the current [JsonReader](#) token.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void WriteToken(  
    JsonReader reader  
)
```

```
Public Sub WriteToken ( _  
    reader As JsonReader _  
)
```

```
public:  
void WriteToken(  
    JsonReader^ reader  
)
```

#### ▣ Parameters

**reader** ([JsonReader](#))

The [JsonReader](#) to read the token from.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteUndefined()**

Writes an undefined value.

### ▣ Declaration Syntax

 C# Visual Basic Visual C++

```
public virtual void WriteUndefined()
```

```
Public Overridable Sub WriteUndefined
```

```
public:  
virtual void WriteUndefined()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

C#

## Members

Icon	Member	Description
	<a href="#">WriteValue(String)</a>	Writes a <a href="#">String</a> value.
	<a href="#">WriteValue(Int32)</a>	Writes a <a href="#">Int32</a> value.
	<a href="#">WriteValue(UInt32)</a>	Writes a <a href="#">UInt32</a> value.
	<a href="#">WriteValue(Int64)</a>	Writes a <a href="#">Int64</a> value.
	<a href="#">WriteValue(UInt64)</a>	Writes a <a href="#">UInt64</a> value.
	<a href="#">WriteValue(Single)</a>	Writes a <a href="#">Single</a> value.
	<a href="#">WriteValue(Double)</a>	Writes a <a href="#">Double</a> value.
	<a href="#">WriteValue(Boolean)</a>	Writes a <a href="#">Boolean</a> value.
	<a href="#">WriteValue(Int16)</a>	Writes a <a href="#">Int16</a> value.
	<a href="#">WriteValue(UInt16)</a>	Writes a <a href="#">UInt16</a> value.

	<a href="#">WriteValue(Char)</a>	Writes a <a href="#">Char</a> value.
	<a href="#">WriteValue(Byte)</a>	Writes a <a href="#">Byte</a> value.
	<a href="#">WriteValue(SByte)</a>	Writes a <a href="#">SByte</a> value.
	<a href="#">WriteValue(Decimal)</a>	Writes a <a href="#">Decimal</a> value.
	<a href="#">WriteValue(DateTime)</a>	Writes a <a href="#">DateTime</a> value
	<a href="#">WriteValue(DateTimeOffset)</a>	Writes a <a href="#">DateTimeOffset</a> value.
	<a href="#">WriteValue(Guid)</a>	Writes a <a href="#">Guid</a> value.
	<a href="#">WriteValue(TimeSpan)</a>	Writes a <a href="#">TimeSpan</a> value
	<a href="#">WriteValue(Nullable&lt;Int32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;UInt32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.

 <a href="#">WriteValue(Nullable&lt;Int64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
 <a href="#">WriteValue(Nullable&lt;UInt64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
 <a href="#">WriteValue(Nullable&lt;Single&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
 <a href="#">WriteValue(Nullable&lt;Double&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
 <a href="#">WriteValue(Nullable&lt;Boolean&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
 <a href="#">WriteValue(Nullable&lt;Int16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
 <a href="#">WriteValue(Nullable&lt;UInt16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
 <a href="#">WriteValue(Nullable&lt;Char&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
 <a href="#">WriteValue(Nullable&lt;Byte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.

	<a href="#">WriteValue(Nullable&lt;SByte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;Decimal&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;DateTime&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;DateTimeOffset&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;Guid&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Nullable&lt;TimeSpan&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
	<a href="#">WriteValue(Byte[])</a>	Writes a <b>Byte[]</b> value.
	<a href="#">WriteValue(Uri)</a>	Writes a <a href="#">Uri</a> value.
	<a href="#">WriteValue(Object)</a>	Writes a <a href="#">Object</a> value. An error will be raised if the value cannot be written

		as a single JSON token.
--	--	-------------------------

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(Boolean)**Writes a [Boolean](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    bool value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Boolean _  
)
```

```
public:  
virtual void WriteValue(  
    bool value  
)
```

▣ **Parameters****value ([Boolean](#))**The [Boolean](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Byte)**

C#

Writes a [Byte](#) value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    byte value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Byte _  
)
```

```
public:  
virtual void WriteValue(  
    unsigned char value  
)
```

### ▣ Parameters

#### **value (Byte)**

The [Byte](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Byte[])**

C#

Writes a **Byte[]** value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    byte[] value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Byte() _  
)
```

```
public:  
virtual void WriteValue(  
    array<unsigned char>^ value  
)
```

### ▣ Parameters

**value (Byte[])**

The **Byte[]** value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Char)**

C#

Writes a [Char](#) value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    char value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Char _  
)
```

```
public:  
virtual void WriteValue(  
    wchar_t value  
)
```

### ▣ Parameters

#### **value (Char)**

The [Char](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►

### WriteValue(DateTime)

Writes a [DateTime](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    DateTime value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As DateTime _  
)
```

```
public:  
virtual void WriteValue(  
    DateTime value  
)
```

#### ▣ Parameters

##### **value** ([DateTime](#))

The [DateTime](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(DateTimeOffset)**Writes a [DateTimeOffset](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    DateTimeOffset value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As DateTimeOffset _  
)
```

```
public:  
virtual void WriteValue(  
    DateTimeOffset value  
)
```

▣ **Parameters****value ([DateTimeOffset](#))**The [DateTimeOffset](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(Decimal)**Writes a [Decimal](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    decimal value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Decimal _  
)
```

```
public:  
virtual void WriteValue(  
    Decimal value  
)
```

▣ **Parameters****value ([Decimal](#))**The [Decimal](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Double)**

C#

Writes a [Double](#) value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    double value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Double _  
)
```

```
public:  
virtual void WriteValue(  
    double value  
)
```

### ▣ Parameters

#### **value (Double)**

The [Double](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Guid)**

C#

Writes a [Guid](#) value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Guid value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Guid _  
)
```

```
public:  
virtual void WriteValue(  
    Guid value  
)
```

### ▣ Parameters

#### **value (Guid)**

The [Guid](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Int16)**

C#

Writes a [Int16](#) value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    short value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Short _  
)
```

```
public:  
virtual void WriteValue(  
    short value  
)
```

### ▣ Parameters

#### **value (Int16)**

The [Int16](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Int32)**

C#

Writes a [Int32](#) value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    int value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Integer _  
)
```

```
public:  
virtual void WriteValue(  
    int value  
)
```

### ▣ Parameters

#### **value (Int32)**

The [Int32](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Int64)**

C#

Writes a [Int64](#) value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    long value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Long _  
)
```

```
public:  
virtual void WriteValue(  
    long long value  
)
```

### ▣ Parameters

#### **value (Int64)**

The [Int64](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►

### WriteValue(Nullable<Boolean>)

Writes a [Nullable<T>](#) value.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public virtual void WriteValue(     Nullable&lt;bool&gt; value )</pre>		
	<pre>Public Overridable Sub WriteValue ( _     value As Nullable(Of Boolean) _ )</pre>	
		<pre>public: virtual void WriteValue(     Nullable&lt;bool&gt; value )</pre>

#### ▣ Parameters

**value** ([Nullable<Boolean>](#))

The [Nullable<T>](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►

### WriteValue(Nullable<Byte>)

Writes a [Nullable<T>](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<byte> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable\(Of Byte\) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<unsigned char> value  
)
```

#### ▣ Parameters

**value** ([Nullable<Byte>](#))

The [Nullable<T>](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(Nullable<Char>)**Writes a [Nullable<T>](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<char> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable\(Of Char\) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<wchar\_t> value  
)
```

▣ **Parameters****value** ([Nullable<Char>](#))The [Nullable<T>](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(Nullable<DateTime>)**

Writes a [Nullable<T>](#) value.

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<DateTime> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable\(Of DateTime\) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<DateTime> value  
)
```

**▣ Parameters****value ([Nullable<DateTime>](#))**

The [Nullable<T>](#) value to write.

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►

### WriteValue(Nullable<DateTimeOffset>)

Writes a [Nullable<T>](#) value.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public virtual void WriteValue(     Nullable&lt;DateTimeOffset&gt; value )</pre>		
	<pre>Public Overridable Sub WriteValue ( _     value As Nullable(Of DateTimeOffset) _ )</pre>	
		<pre>public: virtual void WriteValue(     Nullable&lt;DateTimeOffset&gt; value )</pre>

#### ▣ Parameters

**value** ([Nullable<DateTimeOffset>](#))

The [Nullable<T>](#) value to write.

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(Nullable<Decimal>)**Writes a [Nullable<T>](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<decimal> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable\(Of Decimal\) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<Decimal> value  
)
```

▣ **Parameters****value** ([Nullable<Decimal>](#))The [Nullable<T>](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(Nullable<Double>)**Writes a [Nullable<T>](#) value.**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<double> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable\(Of Double\) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<double> value  
)
```

**▣ Parameters****value ([Nullable<Double>](#))**The [Nullable<T>](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(Nullable<Guid>)**Writes a [Nullable<T>](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<Guid> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable\(Of Guid\) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<Guid> value  
)
```

▣ **Parameters****value ([Nullable<Guid>](#))**The [Nullable<T>](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(Nullable<Int16>)**Writes a [Nullable<T>](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<short> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable\(Of Short\) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<short> value  
)
```

▣ **Parameters****value** ([Nullable<Int16>](#))The [Nullable<T>](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(Nullable<Int32>)**Writes a [Nullable<T>](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<int> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable\(Of Integer\) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<int> value  
)
```

▣ **Parameters****value** ([Nullable<Int32>](#))The [Nullable<T>](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(Nullable<Int64>)**Writes a [Nullable<T>](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<long> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable\(Of Long\) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<long long> value  
)
```

▣ **Parameters****value** ([Nullable<Int64>](#))The [Nullable<T>](#) value to write.

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(Nullable<SByte>)**Writes a [Nullable<T>](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<sbyte> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable\(Of SByte\) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<signed char> value  
)
```

▣ **Parameters****value** ([Nullable<SByte>](#))The [Nullable<T>](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(Nullable<Single>)**Writes a [Nullable<T>](#) value.**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<float> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable\(Of Single\) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<float> value  
)
```

**▣ Parameters****value ([Nullable<Single>](#))**The [Nullable<T>](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(Nullable<TimeSpan>)**

C# ▼

Writes a [Nullable<T>](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<TimeSpan> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable\(Of TimeSpan\) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<TimeSpan> value  
)
```

▣ **Parameters****value** ([Nullable<TimeSpan>](#))The [Nullable<T>](#) value to write.

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(Nullable<UInt16>)**Writes a [Nullable<T>](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<ushort> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable\(Of UShort\) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<unsigned short> value  
)
```

▣ **Parameters****value** ([Nullable<UInt16>](#))The [Nullable<T>](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►

### WriteValue(Nullable<UInt32>)

Writes a [Nullable<T>](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<uint> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable\(Of UIntInteger\) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<unsigned int> value  
)
```

#### ▣ Parameters

**value** ([Nullable<UInt32>](#))

The [Nullable<T>](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValue(Nullable<UInt64>)**Writes a [Nullable<T>](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<ulong> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable(Of ULong) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<unsigned long long> value  
)
```

▣ **Parameters****value** ([Nullable](#)<[UInt64](#)>)The [Nullable<T>](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Object)**

C#

Writes a [Object](#) value. An error will be raised if the value cannot be written as a single JSON token.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Object value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Object _  
)
```

```
public:  
virtual void WriteValue(  
    Object^ value  
)
```

**Parameters****value ([Object](#))**

The [Object](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(SByte)**

C#

Writes a [SByte](#) value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    sbyte value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As SByte _  
)
```

```
public:  
virtual void WriteValue(  
    signed char value  
)
```

### ▣ Parameters

#### **value (SByte)**

The [SByte](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Single)**

C#

Writes a [Single](#) value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    float value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Single _  
)
```

```
public:  
virtual void WriteValue(  
    float value  
)
```

### ▣ Parameters

#### **value (Single)**

The [Single](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(String)**

C#

Writes a [String](#) value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    string value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As String _  
)
```

```
public:  
virtual void WriteValue(  
    String^ value  
)
```

### ▣ Parameters

#### **value (String)**

The [String](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►

### WriteValue(TimeSpan)

Writes a [TimeSpan](#) value.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public virtual void WriteValue(     <a href="#">TimeSpan</a> value )</pre>		
	<pre>Public Overridable Sub WriteValue ( _     value As <a href="#">TimeSpan</a> _ )</pre>	
		<pre>public: virtual void WriteValue(     <a href="#">TimeSpan</a> value )</pre>

#### ▣ Parameters

##### **value** ([TimeSpan](#))

The [TimeSpan](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(UInt16)**

C#

Writes a [UInt16](#) value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    ushort value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As UShort _  
)
```

```
public:  
virtual void WriteValue(  
    unsigned short value  
)
```

### ▣ Parameters

#### **value (UInt16)**

The [UInt16](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(UInt32)**

C#

Writes a [UInt32](#) value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    uint value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As UInteger _  
)
```

```
public:  
virtual void WriteValue(  
    unsigned int value  
)
```

### ▣ Parameters

#### **value (UInt32)**

The [UInt32](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(UInt64)**

C#

Writes a [UInt64](#) value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    ulong value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As ULong _  
)
```

```
public:  
virtual void WriteValue(  
    unsigned long long value  
)
```

### ▣ Parameters

#### **value (UInt64)**

The [UInt64](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Uri)**

C#

Writes a [Uri](#) value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Uri value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Uri _  
)
```

```
public:  
virtual void WriteValue(  
    Uri^ value  
)
```

### ▣ Parameters

#### **value (Uri)**

The [Uri](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**Method**[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►**WriteValueDelimiter()**

Writes the JSON value delimiter.

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
protected virtual void WriteValueDelimiter()
```

```
Protected Overridable Sub WriteValueDelimiter
```

```
protected:  
virtual void WriteValueDelimiter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## (ws)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►

### WriteWhitespace(String)

C#

Writes out the given white space.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteWhitespace(  
    string ws  
)
```

```
Public Overridable Sub Writewhitespace ( _  
    ws As String _  
)
```

```
public:  
virtual void WriteWhitespace(  
    String^ ws  
)
```

#### ▣ Parameters

**ws** ([String](#))

The string of white space characters.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonWriterException

## Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonWriterException**

C#

The exception thrown when an error occurs while reading Json text.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
[SerializableAttribute]  
public class JsonWriterException : JsonException
```

```
<SerializableAttribute> _  
Public Class JsonWriterException _  
    Inherits JsonException
```

```
[SerializableAttribute]  
public ref class JsonWriterException : public JsonEx
```

### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherite

Icon	Member	Description
	<a href="#">JsonWriterException()</a>	Initializes a new instance of the <b>JsonWriterException</b> class.
	<a href="#">JsonWriterException(String)</a>	Initializes a new instance of the

		<b>JsonWriterException</b> class with a specified error message.
	<a href="#">JsonWriterException(String, Exception)</a>	Initializes a new instance of the <b>JsonWriterException</b> class with a specified error message and a reference to the inner exception that is the cause of this exception.
	<a href="#">JsonWriterException(SerializationInfo, StreamingContext)</a>	Initializes a new instance of the <b>JsonWriterException</b> class.
	<a href="#">Data</a>	Gets a collection of key/value pairs that provide additional user-defined information about the exception.  (Inherited from <a href="#">Exception</a> )
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup.

		<p>operations before the <a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetBaseException()</a>	<p>When overridden in a derived class, returns the <a href="#">Exception</a> that is the root cause of one or more subsequent exceptions.</p> <p>(Inherited from <a href="#">Exception</a>)</p>
	<a href="#">GetHashCode()</a>	<p>Serves as a hash function for a particular type.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetObjectData(SerializationInfo, StreamingContext)</a>	<p>When overridden in a derived class, sets the <a href="#">SerializationInfo</a> with information about the exception.</p> <p>(Inherited from <a href="#">Exception</a>)</p>
	<a href="#">GetType()</a>	<p>Gets the runtime type of the current instance.</p> <p>(Inherited from <a href="#">Exception</a>)</p>
	<a href="#">HelpLink</a>	<p>Gets or sets a link to the help file associated with this exception.</p> <p>(Inherited from <a href="#">Exception</a>)</p>
	<a href="#">HResult</a>	<p>Gets or sets HRESULT, a coded numerical value that is assigned to a</p>

		specific exception.  (Inherited from <a href="#">Exception</a> )
	<a href="#">InnerException</a>	Gets the <a href="#">Exception</a> instance that caused the current exception.  (Inherited from <a href="#">Exception</a> )
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Message</a>	Gets a message that describes the current exception.  (Inherited from <a href="#">Exception</a> )
	<a href="#">Path</a>	Gets the path to the JSON where the error occurred.
	<a href="#">Source</a>	Gets or sets the name of the application or the object that causes the error.  (Inherited from <a href="#">Exception</a> )
	<a href="#">StackTrace</a>	Gets a string representation of the frames on the call stack at the time the current exception was thrown.  (Inherited from <a href="#">Exception</a> )
	<a href="#">TargetSite</a>	Gets the method that

		throws the current exception.  (Inherited from <a href="#">Exception</a> )
	<a href="#">ToString()</a>	Creates and returns a string representation of the current exception.  (Inherited from <a href="#">Exception</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [Exception](#)

└─ [JsonException](#)

└─ **JsonWriterException**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriterException](#) ►

### JsonWriterException()

#### Members

Icon	Member	Description
	<a href="#">JsonWriterException()</a>	Initializes a new instance of the <a href="#">JsonWriterException</a> class.
	<a href="#">JsonWriterException(String)</a>	Initializes a new instance of the <a href="#">JsonWriterException</a> class with a specified error message.
	<a href="#">JsonWriterException(String, Exception)</a>	Initializes a new instance of the <a href="#">JsonWriterException</a> class with a specified error message and a reference to the inner exception that is the cause of this exception.
	<a href="#">JsonWriterException(SerializationInfo, StreamingContext)</a>	Initializes a new instance of the <a href="#">JsonWriterException</a> class.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriterException](#) ►

### JsonWriterException()

Initializes a new instance of the [JsonWriterException](#) class.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonWriterException()</pre>		
	<pre>Public Sub New</pre>	
		<pre>public: JsonWriterException()</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonWriterException

## Constructor (info, context)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriterException](#) ►

### JsonWriterException(SerializationInfo, StreamingContext)

Initializes a new instance of the [JsonWriterException](#) class.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonWriterException(     <a href="#">SerializationInfo</a> info,     <a href="#">StreamingContext</a> context )</pre>		
	<pre>Public Sub New ( _     info As <a href="#">SerializationInfo</a>, _     context As <a href="#">StreamingContext</a> _ )</pre>	
		<pre>public: JsonWriterException(     <a href="#">SerializationInfo</a>^ info,     <a href="#">StreamingContext</a> context )</pre>

#### ▣ Parameters

##### **info** ([SerializationInfo](#))

The [SerializationInfo](#) that holds the serialized object data about the exception being thrown.

##### **context** ([StreamingContext](#))

The [StreamingContext](#) that contains contextual information about the source or destination.

#### ▣ Exceptions

Exception	Condition
<a href="#">ArgumentNullException</a>	The <i>info</i> parameter is null.
<a href="#">SerializationException</a>	The class name is null or <a href="#">HResult</a> is zero (0).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor (message)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriterException](#) ►

### JsonWriterException(String)

Initializes a new instance of the [JsonWriterException](#) class with a specified error message.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonWriterException(     <a href="#">string</a> message )</pre>		
	<pre>Public Sub New (      <a href="#">message</a> As <a href="#">String</a>  )</pre>	
		<pre>public: JsonWriterException(     <a href="#">String</a>^ message )</pre>

#### ▣ Parameters

##### **message** ([String](#))

The error message that explains the reason for the exception.

## JsonWriterException Constructor (message, innerException)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriterException](#) ►

### JsonWriterException(String, Exception)

Initializes a new instance of the [JsonWriterException](#) class with a specified error message and a reference to the inner exception that is the cause of this exception.

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonWriterException(     string message,     Exception innerException )</pre>	<pre>Public Sub New ( _     message As String, _     innerException As Exception _ )</pre>	<pre>public: JsonWriterException(     String^ message,     Exception^ innerException )</pre>

#### Parameters

##### **message** ([String](#))

The error message that explains the reason for the exception.

##### **innerException** ([Exception](#))

The exception that is the cause of the current exception, or a null reference (Nothing in Visual Basic) if no inner exception is specified.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

Gets the path to the JSON where the error occurred.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public string Path { get; private set; }</pre>		
<pre>Public Property Path As String     Get     Private Set</pre>		
<pre>public: property String^ Path {     String^ get ();     private: void set (String^ value); }</pre>		

## ▣ Value

The path to the JSON where the error occurred.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Enumeration

[Namespaces](#) ► [Newtonsoft.Json](#) ► **MemberSerialization**

Specifies the member serialization options for the [JsonSerializer](#).

### Declaration Syntax




```
public enum MemberSerialization
```

```
Public Enumeration MemberSerialization
```

```
public enum class MemberSerialization
```

### Members

Member	Description
<b>OptOut</b>	All public members are serialized by default. Members can be excluded using <a href="#">JsonIgnoreAttribute</a> or <a href="#">NonSerializedAttribute</a> . This is the default member serialization mode.
<b>OptIn</b>	Only members must be marked with <a href="#">JsonPropertyAttribute</a> or <a href="#">DataMemberAttribute</a> are serialized. This member serialization mode can also be set by marking the class with <a href="#">DataContractAttribute</a> .
<b>Fields</b>	All public and private fields are serialized. Members can be excluded using <a href="#">JsonIgnoreAttribute</a> or <a href="#">NonSerializedAttribute</a> . This member serialization mode can also be set by

marking the class with [SerializableAttribute](#) and setting IgnoreSerializableAttribute on [DefaultContractResolver](#) to false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Enumeration

[Namespaces](#) ► [Newtonsoft.Json](#) ► **MissingMemberHandling**

C# ▼

Specifies missing member handling options for the [JsonSerializer](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum MissingMemberHandling
```

```
Public Enumeration MissingMemberHandling
```

```
public enum class MissingMemberHandling
```

### Members

Member	Description
<b>Ignore</b>	Ignore a missing member and do not attempt to deserialize it.
<b>Error</b>	Throw a <a href="#">JsonSerializationException</a> when a missing member is encountered during deserialization.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Enumeration

[Namespaces](#) ► [Newtonsoft.Json](#) ► [NullValueHandling](#) C#

Specifies null value handling options for the [JsonSerializer](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum NullValueHandling
```

```
Public Enumeration NullValueHandling
```

```
public enum class NullValueHandling
```

### ▣ Members

Member	Description
<b>Include</b>	Include null values when serializing and deserializing objects.
<b>Ignore</b>	Ignore null values when serializing and deserializing objects.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# ObjectCreationHandling Enumeration

[Namespaces](#) ► [Newtonsoft.Json](#) ► **ObjectCreationHandling**

Specifies how object creation is handled by the [JsonSerializer](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum ObjectCreationHandling
```

```
Public Enumeration ObjectCreationHandling
```

```
public enum class ObjectCreationHandling
```

## Members

Member	Description
<b>Auto</b>	Reuse existing objects, create new object when needed.
<b>Reuse</b>	Only reuse existing objects.
<b>Replace</b>	Always create new objects.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Documentation **PreserveReferencesHandling Enumeration**

[Namespaces](#) ► [Newtonsoft.Json](#) ► **PreserveReferencesHandling**

Specifies reference handling options for the [JsonSerializer](#).

▣ **Declaration Syntax**

C#                      Visual Basic                      Visual C++

```
[FlagsAttribute]
public enum PreserveReferencesHandling
```

```
<FlagsAttribute> _
Public Enumeration PreserveReferencesHandling
```

```
[FlagsAttribute]
public enum class PreserveReferencesHandling
```

▣ **Members**

Member	Description
<b>None</b>	Do not preserve references when serializing types.
<b>Objects</b>	Preserve references when serializing into a JSON object structure.
<b>Arrays</b>	Preserve references when serializing into a JSON array structure.
<b>All</b>	Preserve references when serializing.

# ReferenceLoopHandling Enumeration

[Namespaces](#) ► [Newtonsoft.Json](#) ► **ReferenceLoopHandling**

Specifies reference loop handling options for the [JsonSerializer](#).

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum ReferenceLoopHandling
```

```
Public Enumeration ReferenceLoopHandling
```

```
public enum class ReferenceLoopHandling
```

## Members

Member	Description
<b>Error</b>	Throw a <a href="#">JsonSerializationException</a> when a loop is encountered.
<b>Ignore</b>	Ignore loop references and do not serialize.
<b>Serialize</b>	Serialize loop references.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Indicating whether a property is required.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum Required
```

```
Public Enumeration Required
```

```
public enum class Required
```

## ▣ Members

Member	Description
<b>Default</b>	The property is not required. The default state.
<b>AllowNull</b>	The property must be defined in JSON but can be a null value.
<b>Always</b>	The property must be defined in JSON and cannot be a null value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Enumeration

[Namespaces](#) ► [Newtonsoft.Json](#) ► [TypeNameHandling](#)

Specifies type name handling options for the [JsonSerializer](#).

### Declaration Syntax




```
[FlagsAttribute]  
public enum TypeNameHandling
```

```
<FlagsAttribute> _  
Public Enumeration TypeNameHandling
```

```
[FlagsAttribute]  
public enum class TypeNameHandling
```

### Members

Member	Description
<b>None</b>	Do not include the .NET type name when serializing types.
<b>Objects</b>	Include the .NET type name when serializing into a JSON object structure.
<b>Arrays</b>	Include the .NET type name when serializing into a JSON array structure.
<b>All</b>	Always include the .NET type name when serializing.
<b>Auto</b>	Include the .NET type name when the type of the object being serialized is not the same as its declared type.

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

Specifies the state of the [JsonWriter](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum WriteState
```

```
Public Enumeration WriteState
```

```
public enum class WriteState
```

## Members

Member	Description
<b>Error</b>	An exception has been thrown, which has left the <a href="#">JsonWriter</a> in an invalid state. You may call the <a href="#">Close()</a> method to put the <a href="#">JsonWriter</a> in the <b>Closed</b> state. Any other <a href="#">JsonWriter</a> method calls results in a <a href="#">InvalidOperationException</a> being thrown.
<b>Closed</b>	The <a href="#">Close()</a> method has been called.
<b>Object</b>	An object is being written.
<b>Array</b>	A array is being written.
<b>Constructor</b>	A constructor is being written.
<b>Property</b>	A property is being written.
<b>Start</b>	A write method has not been called.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Namespace

[Namespaces](#) ► **Newtonsoft.Json.Bson**

C#

The **Newtonsoft.Json.Linq** namespace provides classes that are used to implement BSON.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
namespace Newtonsoft.Json.Bson
```

```
Namespace Newtonsoft.Json.Bson
```

```
namespace Newtonsoft.Json.Bson
```

### ▣ Types

All Types

Classes

Icon	Type	Description
	<a href="#">BsonObjectId</a>	Represents a BSON Oid (object id).
	<a href="#">BsonReader</a>	Represents a reader that provides fast, non-cached, forward-only access to serialized Json data.
	<a href="#">BsonWriter</a>	Represents a writer that provides a fast, non-cached, forward-only way of generating Json data.

# BsonObjectId Class

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► **BsonObjectId** C#

Represents a BSON Oid (object id).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public class BsonObjectId
```

```
Public Class BsonObjectId
```

```
public ref class BsonObjectId
```

## Members

All Members

Constructors

Methods

Properties

Public  
 Protected

Instance  
 Static

Declare  
 Inherited

Icon	Member	Description
	<a href="#">BsonObjectId(Byte[])</a>	Initializes a new instance of the <b>BsonObjectId</b> class.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.

		(Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Value</a>	Gets or sets the value of the <a href="#">Oid</a> .

## ▣ Inheritance Hierarchy

[Object](#)

└─ **BsonObjectId**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonObjectId](#) ►

### BsonObjectId(Byte[])

Initializes a new instance of the [BsonObjectId](#) class.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public BsonObjectId(     byte[] value )</pre>		
	<pre>Public Sub New ( _     value As Byte() _ )</pre>	
		<pre>public: BsonObjectId(     array&lt;unsigned char&gt;^ value )</pre>

#### ▣ Parameters

**value (Byte[])**

The ObjectId value.

Gets or sets the value of the Oid.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public byte[] Value { get; private set; }</pre>		
<pre>Public Property Value As Byte()     Get     Private Set</pre>		
<pre>public: property array&lt;unsigned char&gt;^ Value {     array&lt;unsigned char&gt;^ get (); private: void set (array&lt;unsigned char&gt;^ val }</pre>		

## ▣ Value

The value of the Oid.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# BsonReader Class

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► **BsonReader** C#

Represents a reader that provides fast, non-cached, forward-only access to serialized Json data.

## Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class BsonReader : JsonReader
```

```
Public Class BsonReader _
    Inherits JsonReader
```

```
public ref class BsonReader : public JsonReader
```

## Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">BsonReader(Stream)</a>	Initializes a new instance of the <b>BsonReader</b> class.
	<a href="#">BsonReader(BinaryReader)</a>	Initializes a new instance of the <b>BsonReader</b> class.
	<a href="#">BsonReader(Stream, Boolean, DateTimeKind)</a>	Initializes a new instance of the <b>BsonReader</b> class.

	<a href="#">BsonReader(BinaryReader, Boolean, DateTimeKind)</a>	Initializes a new instance of the <b>BsonReader</b> class.
	<a href="#">Close()</a>	Changes the <a href="#">JsonReader.State</a> to Closed.  (Overrides <a href="#">JsonReader.Close()</a> .)
	<a href="#">CloseInput</a>	Gets or sets a value indicating wh the underlying stream or <a href="#">TextRea</a> should be closed when the reader closed.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">Culture</a>	Gets or sets the culture used whe reading JSON. Defaults to <a href="#">InvariantCulture</a> .  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">CurrentState</a>	Gets the current reader state.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">DateParseHandling</a>	Get or set how date formatted stri e.g. "\Date(1198908717056)\V" ar "2012-03-21T05:40Z", are parsed when reading JSON.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">DateTimeKindHandling</a>	Gets or sets the <a href="#">DateTimeKind</a> us when reading <a href="#">DateTime</a> values fr BSON.
	<a href="#">DateTimeZoneHandling</a>	Get or set how <a href="#">DateTime</a> time zor

		are handling when reading JSON (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">Depth</a>	Gets the depth of the current token in the JSON document. (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">Dispose(Boolean)</a>	Releases unmanaged and - optionally managed resources (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> . (Inherited from <a href="#">Object.</a> )
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection. (Inherited from <a href="#">Object.</a> )
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type. (Inherited from <a href="#">Object.</a> )
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object.</a> )
	<a href="#">JsonNet35BinaryCompatibility</a>	Gets or sets a value indicating whether binary data reading should be compatible with incorrect Json.NET 3.5 written binary.
	<a href="#">MaxDepth</a>	Gets or sets the maximum depth

		<p>allowed when reading JSON. Reading past this depth will throw a <a href="#">JsonReaderException</a>.</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
	<a href="#">MemberwiseClone()</a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">Path</a>	<p>Gets the path of the current JSON token.</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
	<a href="#">QuoteChar</a>	<p>Gets the quotation mark character used to enclose the value of a string.</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
	<a href="#">Read()</a>	<p>Reads the next JSON token from stream.</p> <p>(Overrides <a href="#">JsonReader.Read()</a>.)</p>
	<a href="#">ReadAsBytes()</a>	<p>Reads the next JSON token from stream as a <b>Byte[]</b>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsBytes()</a>.)</p>
	<a href="#">ReadAsDateTime()</a>	<p>Reads the next JSON token from stream as a <a href="#">Nullable&lt;T&gt;</a>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsDateTime()</a>.)</p>
	<a href="#">ReadAsDateTimeOffset()</a>	<p>Reads the next JSON token from stream as a <a href="#">Nullable&lt;T&gt;</a>.</p> <p>(Overrides</p>

		<a href="#">JsonReader.ReadAsDateTimeOff</a>
	<a href="#">ReadAsDecimal()</a>	<p>Reads the next JSON token from stream as a <a href="#">Nullable&lt;T&gt;</a>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsDecimal()</a>.)</p>
	<a href="#">ReadAsInt32()</a>	<p>Reads the next JSON token from stream as a <a href="#">Nullable&lt;T&gt;</a>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsInt32()</a>.)</p>
	<a href="#">ReadAsString()</a>	<p>Reads the next JSON token from stream as a <a href="#">String</a>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsString()</a>.)</p>
	<a href="#">ReadRootValueAsArray</a>	<p>Gets or sets a value indicating wh the root object will be read as a J array.</p>
	<a href="#">SetStateBasedOnCurrent()</a>	<p>Sets the state based on current to type.</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
	<a href="#">SetToken(JsonToken)</a>	<p>Sets the current token.</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
	<a href="#">SetToken(JsonToken, Object)</a>	<p>Sets the current token and value.</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
	<a href="#">Skip()</a>	<p>Skips the children of the current to</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
		

	<a href="#">TokenType</a>	Gets the type of the current JSON token.  (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">Value</a>	Gets the text value of the current token.  (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">ValueType</a>	Gets The Common Language Runtime (CLR) type for the current JSON token.  (Inherited from <a href="#">JsonReader.</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonReader](#)

└─ **BsonReader**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# BsonReader Constructor

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

## BsonReader()

### Members

Icon	Member	Description
	<a href="#">BsonReader(Stream)</a>	Initializes a new instance of the <a href="#">BsonReader</a> class.
	<a href="#">BsonReader(BinaryReader)</a>	Initializes a new instance of the <a href="#">BsonReader</a> class.
	<a href="#">BsonReader(Stream, Boolean, DateTimeKind)</a>	Initializes a new instance of the <a href="#">BsonReader</a> class.
	<a href="#">BsonReader(BinaryReader, Boolean, DateTimeKind)</a>	Initializes a new instance of the <a href="#">BsonReader</a> class.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# BsonReader Constructor (reader)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

## BsonReader(BinaryReader)

Initializes a new instance of the [BsonReader](#) class.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public BsonReader(  
    BinaryReader reader  
)
```

```
Public Sub New ( _  
    reader As BinaryReader _  
)
```

```
public:  
BsonReader(  
    BinaryReader^ reader  
)
```

### Parameters

**reader** ([BinaryReader](#))

The reader.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# BsonReader Constructor (reader, readRootValueAsArray, dateTimeKindHandling)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

**BsonReader(BinaryReader, Boolean, DateTimeKind)**

C# ▼

Initializes a new instance of the [BsonReader](#) class.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public BsonReader(  
    BinaryReader reader,  
    bool readRootValueAsArray,  
    DateTimeKind dateTimeKindHandling  
)
```

```
Public Sub New ( _  
    reader As BinaryReader, _  
    readRootValueAsArray As Boolean, _  
    dateTimeKindHandling As DateTimeKind _  
)
```

```
public:  
BsonReader(  
    BinaryReader^ reader,  
    bool readRootValueAsArray,  
    DateTimeKind dateTimeKindHandling  
)
```

## ▣ Parameters

**reader** ([BinaryReader](#))

The reader.

**readRootValueAsArray** ([Boolean](#))

if set to `true` the root object will be read as a JSON array.

***dateTimeKindHandling* ([DateTimeKind](#))**

The [DateTimeKind](#) used when reading [DateTime](#) values from BSO

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# BsonReader Constructor (stream)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

## BsonReader(Stream)

Initializes a new instance of the [BsonReader](#) class.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public BsonReader(  
    Stream stream  
)
```

```
Public Sub New ( _  
    stream As Stream _  
)
```

```
public:  
BsonReader(  
    Stream^ stream  
)
```

### Parameters

***stream*** ([Stream](#))

The stream.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# BsonReader Constructor (stream, readRootValueAsArray, dateTimeKindHandling)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

**BsonReader(Stream, Boolean, DateTimeKind)**

C#

Initializes a new instance of the [BsonReader](#) class.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public BsonReader(  
    Stream stream,  
    bool readRootValueAsArray,  
    DateTimeKind dateTimeKindHandling  
)
```

```
Public Sub New ( _  
    stream As Stream, _  
    readRootValueAsArray As Boolean, _  
    dateTimeKindHandling As DateTimeKind _  
)
```

```
public:  
BsonReader(  
    Stream^ stream,  
    bool readRootValueAsArray,  
    DateTimeKind dateTimeKindHandling  
)
```

## Parameters

**stream** ([Stream](#))

The stream.

**readRootValueAsArray** ([Boolean](#))

if set to `true` the root object will be read as a JSON array.

***dateTimeKindHandling* ([DateTimeKind](#))**

The [DateTimeKind](#) used when reading [DateTime](#) values from BSO

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ► **Close()**

C#

Changes the [JsonReader.State](#) to Closed.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void Close()
```

```
Public Overrides Sub Close
```

```
public:  
virtual void Close() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

### DateTimeKindHandling

Gets or sets the [DateTimeKind](#) used when reading [DateTime](#) values from BSON.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public DateTimeKind DateTimeKindHandling { get; set;
```

```
Public Property DateTimeKindHandling As DateTimeKind  
    Get  
    Set
```

```
public:  
property DateTimeKind DateTimeKindHandling {  
    DateTimeKind get ();  
    void set (DateTimeKind value);  
}
```

#### Value

The [DateTimeKind](#) used when reading [DateTime](#) values from BSON.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**JsonNet35BinaryCompatibility**

Gets or sets a value indicating whether binary data reading should compatible with incorrect Json.NET 3.5 written binary.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public bool JsonNet35BinaryCompatibility { get; set;
```

```
Public Property JsonNet35BinaryCompatibility As Bool  
    Get  
    Set
```

```
public:  
property bool JsonNet35BinaryCompatibility {  
    bool get ();  
    void set (bool value);  
}
```

**Value**

**true** if binary data reading will be compatible with incorrect Json.NET 3.5 written binary; otherwise, **false**.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ► **Read()**

C#

Reads the next JSON token from the stream.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override bool Read()
```

```
Public Overrides Function Read As Boolean
```

```
public:  
virtual bool Read() override
```

## ▣ Return Value

true if the next token was read successfully; false if there are no more tokens to read.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

## ReadAsBytes()

Reads the next JSON token from the stream as a **Byte**.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override byte[] ReadAsBytes()
```

```
Public Overrides Function ReadAsBytes As Byte()
```

```
public:  
virtual array<unsigned char>^ ReadAsBytes() override
```

### ▣ Return Value

A **Byte** or a null reference if the next JSON token is null. This method will return **null** at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

## ReadAsDateTime()

Reads the next JSON token from the stream as a [Nullable<T>](#).

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<DateTime> ReadAsDateTime()
```

```
Public Overrides Function ReadAsDateTime As Nullable
```

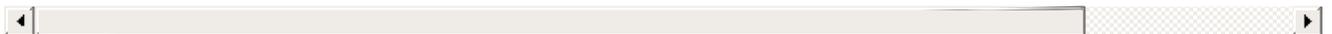
```
public:  
virtual Nullable<DateTime> ReadAsDateTime() override
```

### ▣ Return Value

A [String](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Method

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

### ReadAsDateTimeOffset()

Reads the next JSON token from the stream as a [Nullable<T>](#).

C#

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<DateTimeOffset> ReadAsDateT
```

```
Public Overrides Function ReadAsDateTimeOffset As Nu
```

```
public:  
virtual Nullable<DateTimeOffset> ReadAsDateT
```

#### Return Value

A [Nullable<T>](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

## ReadAsDecimal()

Reads the next JSON token from the stream as a [Nullable<T>](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<decimal> ReadAsDecimal()
```

```
Public Overrides Function ReadAsDecimal As Nullable(
```

```
public:  
virtual Nullable<Decimal> ReadAsDecimal() override
```

### ▣ Return Value

A [Nullable<T>](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

## ReadAsInt32()

Reads the next JSON token from the stream as a [Nullable<T>](#).

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<int> ReadAsInt32()
```

```
Public Overrides Function ReadAsInt32 As Nullable\(Of
```

```
public:  
virtual Nullable<int> ReadAsInt32() override
```

### ▣ Return Value

A [Nullable<T>](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

## ReadAsString()

Reads the next JSON token from the stream as a [String](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override string ReadAsString()
```

```
Public Overrides Function ReadAsString As String
```

```
public:  
virtual String^ ReadAsString() override
```

### ▣ Return Value

A [String](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

### ReadRootValueAsArray

Gets or sets a value indicating whether the root object will be read as a JSON array.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool ReadRootValueAsArray { get; set; }
```

```
Public Property ReadRootValueAsArray As Boolean  
    Get  
    Set
```

```
public:  
property bool ReadRootValueAsArray {  
    bool get ();  
    void set (bool value);  
}
```

#### Value

`true` if the root object will be read as a JSON array; otherwise, `false`

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# BsonWriter Class

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► **BsonWriter** C#

Represents a writer that provides a fast, non-cached, forward-only way generating Json data.

## Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class BsonWriter : JsonWriter
```

```
Public Class BsonWriter _
    Inherits JsonWriter
```

```
public ref class BsonWriter : public JsonWriter
```

## Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
			<input checked="" type="checkbox"/> Declare
			<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">BsonWriter(Stream)</a>	Initializes a new instance <b>BsonWriter</b> class.
	<a href="#">BsonWriter(BinaryWriter)</a>	Initializes a new instance <b>BsonWriter</b> class.
	<a href="#">Close()</a>	Closes this stream and the stream.

		(Overrides <a href="#">JsonWriter.Cl</a>
	<a href="#">CloseOutput</a>	Gets or sets a value indicating the underlying stream or should be closed when the stream is closed.  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">DateFormatHandling</a>	Get or set how dates are formatted as text.  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">DateTimeKindHandling</a>	Gets or sets the <a href="#">DateTimeKind</a> when writing <a href="#">DateTime</a> values. When set to <a href="#">Unspecified</a> , the <a href="#">DateTimeKind</a> will occur.
	<a href="#">DateTimeZoneHandling</a>	Get or set how <a href="#">DateTime</a> values are handled when writing JSON.  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">Equals(Object)</a>	Determines whether the current <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is garbage collected.  (Inherited from <a href="#">Object</a> .)
	<a href="#">Flush()</a>	Flushes whatever is in the underlying streams and the underlying stream.

		(Overrides <a href="#">JsonWriter.Flush()</a> )
	<a href="#">Formatting</a>	Indicates how JSON text is formatted.  (Inherited from <a href="#">JsonWriter.Formatting</a> )
	<a href="#">GetHashCode()</a>	Serves as a hash function for an object.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Path</a>	Gets the path of the writer.  (Inherited from <a href="#">JsonWriter.Path</a> )
	<a href="#">Top</a>	Gets the top.  (Inherited from <a href="#">JsonWriter.Top</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteComment(String)</a>	Writes out a comment.  <b>C#</b> <pre>/* ... */</pre> containing the specified text.  (Overrides <a href="#">JsonWriter.WriteComment(String)</a> )

		<a href="#">JsonWriter.WriteComme</a>
	<a href="#">WriteEnd(JsonToken)</a>	Writes the end.  (Overrides <a href="#">JsonWriter.WriteEnd(Jso</a>
	<a href="#">WriteEnd()</a>	Writes the end of the cur or array.  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteEndArray()</a>	Writes the end of an arra  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteEndConstructor()</a>	Writes the end construct  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteEndObject()</a>	Writes the end of a Json  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteIndent()</a>	Writes indent characters.  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteIndentSpace()</a>	Writes an indent space.  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteNull()</a>	Writes a null value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteObjectId(Byte[])</a>	Writes a <b>Byte[]</b> value tha BSON object id.
	<a href="#">WritePropertyName(String)</a>	Writes the property name name/value pair on a Jsc

		(Overrides <a href="#">JsonWriter.WriteProperty</a> )
	<a href="#">WriteRaw(String)</a>	Writes raw JSON.  (Overrides <a href="#">JsonWriter.W</a> )
	<a href="#">WriteRawValue(String)</a>	Writes raw JSON where expected and updates th  (Overrides <a href="#">JsonWriter.WriteRawVal</a> )
	<a href="#">WriteRegex(String, String)</a>	Writes a BSON regex.
	<a href="#">WriteStartArray()</a>	Writes the beginning of a  (Overrides <a href="#">JsonWriter.W</a> )
	<a href="#">WriteStartConstructor(String)</a>	Writes the start of a cons given name.  (Overrides <a href="#">JsonWriter.WriteStartCo</a> )
	<a href="#">WriteStartObject()</a>	Writes the beginning of a  (Overrides <a href="#">JsonWriter.WriteStartObj</a> )
	<a href="#">WriteState</a>	Gets the state of the writ  (Inherited from <a href="#">JsonWrite</a> )
	<a href="#">WriteToken(JsonReader)</a>	Writes the current <a href="#">JsonR</a>  (Inherited from <a href="#">JsonWrite</a> )
	<a href="#">WriteUndefined()</a>	Writes an undefined valu  (Overrides <a href="#">JsonWriter.W</a> )

 <a href="#">WriteValue(String)</a>	Writes a <a href="#">String</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(S</a>
 <a href="#">WriteValue(Int32)</a>	Writes a <a href="#">Int32</a> value.  (Overrides <a href="#">JsonWriter.W</a>
 <a href="#">WriteValue(UInt32)</a>	Writes a <a href="#">UInt32</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(U</a>
 <a href="#">WriteValue(Int64)</a>	Writes a <a href="#">Int64</a> value.  (Overrides <a href="#">JsonWriter.W</a>
 <a href="#">WriteValue(UInt64)</a>	Writes a <a href="#">UInt64</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(U</a>
 <a href="#">WriteValue(Single)</a>	Writes a <a href="#">Single</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(S</a>
 <a href="#">WriteValue(Double)</a>	Writes a <a href="#">Double</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
 <a href="#">WriteValue(Boolean)</a>	Writes a <a href="#">Boolean</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(B</a>
 <a href="#">WriteValue(Int16)</a>	Writes a <a href="#">Int16</a> value.  (Overrides <a href="#">JsonWriter.W</a>
 <a href="#">WriteValue(UInt16)</a>	Writes a <a href="#">UInt16</a> value.

		(Overrides <a href="#">JsonWriter.WriteValue(U</a>
	<a href="#">WriteValue(Char)</a>	Writes a <a href="#">Char</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(Byte)</a>	Writes a <a href="#">Byte</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(SByte)</a>	Writes a <a href="#">SByte</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(S</a>
	<a href="#">WriteValue(Decimal)</a>	Writes a <a href="#">Decimal</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(DateTime)</a>	Writes a <a href="#">DateTime</a> value  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(DateTimeOffset)</a>	Writes a <a href="#">DateTimeOffset</a>  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(Byte[])</a>	Writes a <b>Byte[]</b> value.  (Overrides <a href="#">JsonWriter.WriteValue(B</a>
	<a href="#">WriteValue(Guid)</a>	Writes a <a href="#">Guid</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(TimeSpan)</a>	Writes a <a href="#">TimeSpan</a> value  (Overrides

		<a href="#">JsonWriter.WriteValue(Ti</a>
	<a href="#">WriteValue(Uri)</a>	Writes a <a href="#">Uri</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(Nullable&lt;Int32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;UInt32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Int64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;UInt64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Single&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Double&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Boolean&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Int16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;UInt16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Char&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>

	<a href="#">WriteValue(Nullable&lt;Byte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value. (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteValue(Nullable&lt;SByte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value. (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteValue(Nullable&lt;Decimal&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value. (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteValue(Nullable&lt;DateTime&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value. (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteValue(Nullable&lt;DateTimeOffset&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value. (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteValue(Nullable&lt;Guid&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value. (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteValue(Nullable&lt;TimeSpan&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value. (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteValue(Object)</a>	Writes a <a href="#">Object</a> value. An exception is thrown if the value cannot be written as a JSON token. (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteValueDelimiter()</a>	Writes the JSON value delimiter. (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteWhitespace(String)</a>	Writes out the given whitespace. (Inherited from <a href="#">JsonWriter</a> )

## Inheritance Hierarchy

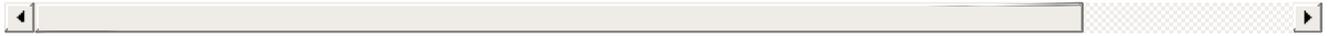
## [Object](#)

└─ [JsonWriter](#)

└─ **BsonWriter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ► **BsonWriter()**

## Members

Icon	Member	Description
	<a href="#">BsonWriter(Stream)</a>	Initializes a new instance of the <a href="#">BsonWriter</a> class.
	<a href="#">BsonWriter(BinaryWriter)</a>	Initializes a new instance of the <a href="#">BsonWriter</a> class.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# BsonWriter Constructor (writer)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

## BsonWriter(BinaryWriter)

Initializes a new instance of the [BsonWriter](#) class.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public BsonWriter(  
    BinaryWriter writer  
)
```

```
Public Sub New ( _  
    writer As BinaryWriter _  
)
```

```
public:  
BsonWriter(  
    BinaryWriter^ writer  
)
```

### ▣ Parameters

**writer** ([BinaryWriter](#))

The writer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# BsonWriter Constructor (stream)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

## BsonWriter(Stream)

Initializes a new instance of the [BsonWriter](#) class.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public BsonWriter(  
    Stream stream  
)
```

```
Public Sub New ( _  
    stream As Stream _  
)
```

```
public:  
BsonWriter(  
    Stream^ stream  
)
```

### ▣ Parameters

#### ***stream*** ([Stream](#))

The stream.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ► **Close()**

C#

Closes this stream and the underlying stream.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void Close()
```

```
Public Overrides Sub Close
```

```
public:  
virtual void Close() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### DateTimeKindHandling

Gets or sets the [DateTimeKind](#) used when writing [DateTime](#) values to BSON. When set to [Unspecified](#) no conversion will occur.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public DateTimeKind DateTimeKindHandling { get; set;
```

```
Public Property DateTimeKindHandling As DateTimeKind  
    Get  
    Set
```

```
public:  
property DateTimeKind DateTimeKindHandling {  
    DateTimeKind get ();  
    void set (DateTimeKind value);  
}
```

#### Value

The [DateTimeKind](#) used when writing [DateTime](#) values to BSON.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ► **Flush()**

C#

Flushes whatever is in the buffer to the underlying streams and also flushes the underlying stream.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void Flush()
```

```
Public Overrides Sub Flush
```

```
public:  
virtual void Flush() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (text)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### WriteComment(String)

C#

Writes out a comment

C#

 Cop

```
/* ... */
```

containing the specified text.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteComment(  
    string text  
)
```

```
Public Overrides Sub WriteComment ( _  
    text As String _  
)
```

```
public:  
virtual void WriteComment(  
    String^ text  
) override
```

#### ▣ Parameters

**text** ([String](#))

Text to place inside the comment.

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

## WriteEnd(JsonToken)

C# ▼

Writes the end.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected override void WriteEnd(  
    JsonToken token  
)
```

```
Protected Overrides Sub WriteEnd ( _  
    token As JsonToken _  
)
```

```
protected:  
virtual void WriteEnd(  
    JsonToken token  
) override
```

### ▣ Parameters

***token*** ([JsonToken](#))

The token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ► **WriteNull()**

C#

Writes a null value.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteNull()
```

```
Public Overrides Sub WriteNull
```

```
public:  
virtual void WriteNull() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### WriteObjectId(Byte[])

Writes a **Byte[]** value that represents a BSON object id.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public void WriteObjectId(     byte[] value )</pre>	<pre>Public Sub WriteObjectId ( _     value As Byte() _ )</pre>	<pre>public: void WriteObjectId(     array&lt;unsigned char&gt;^ value )</pre>

#### ▣ Parameters

**value** ([Byte\[\]](#))

**[Missing <param name="value"/> documentation for "M:Newtonsoft.Json.Bson.BsonWriter.WriteObjectId(System.B**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Method (name)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### WritePropertyName(String)

Writes the property name of a name/value pair on a Json object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WritePropertyName(  
    string name  
)
```

```
Public Overrides Sub WritePropertyName ( _  
    name As String _  
)
```

```
public:  
virtual void WritePropertyName(  
    String^ name  
) override
```

#### ▣ Parameters

**name** ([String](#))

The name of the property.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

## WriteRaw(String)

Writes raw JSON.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public override void WriteRaw(  
    string json  
)
```

```
Public Overrides Sub WriteRaw ( _  
    json As String _  
)
```

```
public:  
virtual void WriteRaw(  
    String^ json  
) override
```

### ▣ Parameters

*json* (**String**)

The raw JSON to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(json)**[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►**WriteRawValue(String)**

Writes raw JSON where a value is expected and updates the writer's state. C# ▼

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteRawValue(  
    string json  
)
```

```
Public Overrides Sub WriteRawValue ( _  
    json As String _  
)
```

```
public:  
virtual void WriteRawValue(  
    String^ json  
) override
```

▣ **Parameters****json (String)**

The raw JSON to write.

## (pattern, options)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### WriteRegex(String, String)

Writes a BSON regex.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public void WriteRegex(     string pattern,     string options )</pre>	<pre>Public Sub WriteRegex ( _     pattern As String, _     options As String _ )</pre>	<pre>public: void WriteRegex(     String^ pattern,     String^ options )</pre>

#### ▣ Parameters

##### **pattern** ([String](#))

The regex pattern.

##### **options** ([String](#))

The regex options.

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

## WriteStartArray()

Writes the beginning of a Json array.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteStartArray()
```

```
Public Overrides Sub WriteStartArray
```

```
public:  
virtual void WriteStartArray() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**Method (name)**[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►**WriteStartConstructor(String)**

Writes the start of a constructor with the given name.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteStartConstructor(  
    string name  
)
```

```
Public Overrides Sub WriteStartConstructor ( _  
    name As String _  
)
```

```
public:  
virtual void WriteStartConstructor(  
    String^ name  
) override
```

**Parameters*****name*** ([String](#))

The name of the constructor.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

## WriteStartObject()

Writes the beginning of a Json object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteStartObject()
```

```
Public Overrides Sub WriteStartObject
```

```
public:  
virtual void WriteStartObject() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►**WriteUndefined()**

Writes an undefined value.

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteUndefined()
```

```
Public Overrides Sub WriteUndefined
```

```
public:  
virtual void WriteUndefined() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Members

Icon	Member	Description
	<a href="#">WriteValue(String)</a>	Writes a <a href="#">String</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(S</a>
	<a href="#">WriteValue(Int32)</a>	Writes a <a href="#">Int32</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(In</a>
	<a href="#">WriteValue(UInt32)</a>	Writes a <a href="#">UInt32</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(U</a>
	<a href="#">WriteValue(Int64)</a>	Writes a <a href="#">Int64</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(In</a>
	<a href="#">WriteValue(UInt64)</a>	Writes a <a href="#">UInt64</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(U</a>
	<a href="#">WriteValue(Single)</a>	Writes a <a href="#">Single</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(S</a>
	<a href="#">WriteValue(Double)</a>	Writes a <a href="#">Double</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(Boolean)</a>	Writes a <a href="#">Boolean</a> value.

		(Overrides <a href="#">JsonWriter.WriteValue(B</a>
	<a href="#">WriteValue(Int16)</a>	Writes a <a href="#">Int16</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(In</a>
	<a href="#">WriteValue(UInt16)</a>	Writes a <a href="#">UInt16</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(U</a>
	<a href="#">WriteValue(Char)</a>	Writes a <a href="#">Char</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(C</a>
	<a href="#">WriteValue(Byte)</a>	Writes a <a href="#">Byte</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(SByte)</a>	Writes a <a href="#">SByte</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(S</a>
	<a href="#">WriteValue(Decimal)</a>	Writes a <a href="#">Decimal</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(DateTime)</a>	Writes a <a href="#">DateTime</a> value  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(DateTimeOffset)</a>	Writes a <a href="#">DateTimeOffset</a>  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
		

	<a href="#">WriteValue(Byte[])</a>	Writes a <b>Byte[]</b> value.  (Overrides <a href="#">JsonWriter.WriteValue(B</a>
	<a href="#">WriteValue(Guid)</a>	Writes a <a href="#">Guid</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(G</a>
	<a href="#">WriteValue(TimeSpan)</a>	Writes a <a href="#">TimeSpan</a> value  (Overrides <a href="#">JsonWriter.WriteValue(Ti</a>
	<a href="#">WriteValue(Uri)</a>	Writes a <a href="#">Uri</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(Nullable&lt;Int32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;UInt32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Int64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;UInt64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Single&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Double&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Boolean&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val

	<a href="#">WriteValue(Nullable&lt;Int16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;UInt16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Char&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Byte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;SByte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Decimal&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;DateTime&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;DateTimeOffset&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Guid&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;TimeSpan&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Object)</a>	Writes a <a href="#">Object</a> value. An exception is raised if the value cannot be written.

single JSON token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### WriteValue(Boolean)

Writes a [Boolean](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    bool value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Boolean _  
)
```

```
public:  
virtual void WriteValue(  
    bool value  
) override
```

#### ▣ Parameters

**value** ([Boolean](#))

The [Boolean](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►**WriteValue(Byte)**Writes a [Byte](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    byte value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Byte _  
)
```

```
public:  
virtual void WriteValue(  
    unsigned char value  
) override
```

▣ **Parameters****value ([Byte](#))**The [Byte](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►**WriteValue(Byte[])**

C# ▼

Writes a **Byte[]** value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    byte[] value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Byte() _  
)
```

```
public:  
virtual void WriteValue(  
    array<unsigned char>^ value  
) override
```

▣ **Parameters****value** ([Byte\[\]](#))The **Byte[]** value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►**WriteValue(Char)**Writes a [Char](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    char value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Char _  
)
```

```
public:  
virtual void WriteValue(  
    wchar\_t value  
) override
```

▣ **Parameters****value ([Char](#))**The [Char](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►**WriteValue(DateTime)**Writes a [DateTime](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    DateTime value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As DateTime _  
)
```

```
public:  
virtual void WriteValue(  
    DateTime value  
) override
```

▣ **Parameters****value ([DateTime](#))**The [DateTime](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### WriteValue(DateTimeOffset)

C#

Writes a [DateTimeOffset](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    DateTimeOffset value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As DateTimeOffset _  
)
```

```
public:  
virtual void WriteValue(  
    DateTimeOffset value  
) override
```

#### ▣ Parameters

**value** ([DateTimeOffset](#))

The [DateTimeOffset](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### WriteValue(Decimal)

C#

Writes a [Decimal](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    decimal value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Decimal _  
)
```

```
public:  
virtual void WriteValue(  
    Decimal value  
) override
```

#### ▣ Parameters

##### **value** ([Decimal](#))

The [Decimal](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### WriteValue(Double)

C#

Writes a [Double](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    double value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Double _  
)
```

```
public:  
virtual void WriteValue(  
    double value  
) override
```

#### ▣ Parameters

##### **value** ([Double](#))

The [Double](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►**WriteValue(Guid)**Writes a [Guid](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    Guid value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Guid _  
)
```

```
public:  
virtual void WriteValue(  
    Guid value  
) override
```

▣ **Parameters****value ([Guid](#))**The [Guid](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### WriteValue(Int16)

C#

Writes a [Int16](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    short value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Short _  
)
```

```
public:  
virtual void WriteValue(  
    short value  
) override
```

#### ▣ Parameters

##### **value** ([Int16](#))

The [Int16](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### WriteValue(Int32)

C#

Writes a [Int32](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    int value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Integer _  
)
```

```
public:  
virtual void WriteValue(  
    int value  
) override
```

#### ▣ Parameters

##### **value** ([Int32](#))

The [Int32](#) value to write.

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►**WriteValue(Int64)**Writes a [Int64](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    long value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Long _  
)
```

```
public:  
virtual void WriteValue(  
    long long value  
) override
```

▣ **Parameters****value ([Int64](#))**The [Int64](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►**WriteValue(SByte)**Writes a [SByte](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    sbyte value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As SByte _  
)
```

```
public:  
virtual void WriteValue(  
    signed char value  
) override
```

▣ **Parameters****value (SByte)**The [SByte](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### WriteValue(Single)

C#

Writes a [Single](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    float value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Single _  
)
```

```
public:  
virtual void WriteValue(  
    float value  
) override
```

#### ▣ Parameters

**value** ([Single](#))

The [Single](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►**WriteValue(String)**Writes a [String](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    string value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As String _  
)
```

```
public:  
virtual void WriteValue(  
    String^ value  
) override
```

▣ **Parameters****value ([String](#))**The [String](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►**WriteValue(TimeSpan)**

Writes a [TimeSpan](#) value.

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    TimeSpan value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As TimeSpan _  
)
```

```
public:  
virtual void WriteValue(  
    TimeSpan value  
) override
```

**▣ Parameters****value ([TimeSpan](#))**

The [TimeSpan](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►**WriteValue(UInt16)**Writes a [UInt16](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    ushort value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As UShort _  
)
```

```
public:  
virtual void WriteValue(  
    unsigned short value  
) override
```

▣ **Parameters****value ([UInt16](#))**The [UInt16](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### WriteValue(UInt32)

Writes a [UInt32](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    uint value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As UInteger _  
)
```

```
public:  
virtual void WriteValue(  
    unsigned int value  
) override
```

#### ▣ Parameters

**value** ([UInt32](#))

The [UInt32](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### WriteValue(UInt64)

C#

Writes a [UInt64](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    ulong value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As ULong _  
)
```

```
public:  
virtual void WriteValue(  
    unsigned long long value  
) override
```

#### ▣ Parameters

**value** ([UInt64](#))

The [UInt64](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### WriteValue(Uri)

Writes a [Uri](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    Uri value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Uri _  
)
```

```
public:  
virtual void WriteValue(  
    Uri^ value  
) override
```

#### ▣ Parameters

##### **value ([Uri](#))**

The [Uri](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

The **Newtonsoft.Json.Converters** namespace provides classes that inherit from [JsonConverter](#).

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
namespace Newtonsoft.Json.Converters
```

```
Namespace Newtonsoft.Json.Converters
```

```
namespace Newtonsoft.Json.Converters
```

▣ **Types**

All Types

Classes

Icon	Type	Description
	<a href="#">BinaryConverter</a>	Converts a binary value to and from a base 64 string value.
	<a href="#">BsonObjectIdConverter</a>	Converts a <a href="#">BsonObjectId</a> to and from JSON and BSON.
	<a href="#">CustomCreationConverter&lt;T&gt;</a>	Create a custom object
	<a href="#">DataSetConverter</a>	Converts a <a href="#">DataSet</a> to and from JSON.

	<a href="#"><u><b>DataTableConverter</b></u></a>	Converts a <a href="#"><u>DataTable</u></a> to and from JSON.
	<a href="#"><u><b>DateTimeConverterBase</b></u></a>	Provides a base class for converting a <a href="#"><u>DateTime</u></a> to and from JSON.
	<a href="#"><u><b>EntityKeyMemberConverter</b></u></a>	Converts an Entity Framework EntityKey to and from JSON.
	<a href="#"><u><b>ExpandoObjectConverter</b></u></a>	Converts an ExpandoObject to and from JSON.
	<a href="#"><u><b>IsoDateTimeConverter</b></u></a>	Converts a <a href="#"><u>DateTime</u></a> to and from the ISO 8601 date format (e.g. 2008-04-12T12:53Z).
	<a href="#"><u><b>JavaScriptDateTimeConverter</b></u></a>	Converts a <a href="#"><u>DateTime</u></a> to and from a JavaScript date constructor (e.g. new Date(52231943)).
	<a href="#"><u><b>KeyValuePairConverter</b></u></a>	Converts a <a href="#"><u>KeyValuePair&lt;TKey, TValue&gt;</u></a> to and from JSON.
	<a href="#"><u><b>RegexConverter</b></u></a>	Converts a <a href="#"><u>Regex</u></a> to and from JSON and BSON.

	<b><u><a href="#">StringEnumConverter</a></u></b>	<p>Converts an <a href="#">Enum</a> to and from its name string value.</p> <p>Converts an <a href="#">Enum</a> to and from its name string value.</p>
	<b><u><a href="#">VersionConverter</a></u></b>	<p>Converts a <a href="#">Version</a> to and from a string (e.g. "1.2.3.4").</p>
	<b><u><a href="#">XmlNodeConverter</a></u></b>	<p>Converts XML to and from JSON.</p>

---

# BinaryConverter Class

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► **BinaryConverter**

Converts a binary value to and from a base 64 string value.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public class BinaryConverter : JsonConverter
```

```
Public Class BinaryConverter _  
    Inherits JsonConverter
```

```
public ref class BinaryConverter : public JsonConver
```

## Members

All Members

Constructors

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declare
- Inherited

Icon	Member	Description
	<a href="#">BinaryConverter()</a>	Initializes a new instance of the <b>BinaryConverter</b> class
	<a href="#">CanConvert(Type)</a>	Determines whether this instance can convert the specified object type.  (Overrides <a href="#">JsonConverter.CanConvert(Type)</a> .)
	<a href="#">CanRead</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can read JSON.

		(Inherited from <a href="#">JsonConverter.</a> )
	<a href="#">CanWrite</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can write JSON.  (Inherited from <a href="#">JsonConverter.</a> )
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object.</a> )
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object.</a> )
	<a href="#">GetSchema()</a>	Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a> .  (Inherited from <a href="#">JsonConverter.</a> )
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object.</a> )
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">ReadJson(JsonReader, Type, Object, JsonSerializer)</a>	Reads the JSON representation of the object.  (Overrides <a href="#">JsonConverter.ReadJson(JsonReader</a>

		<a href="#">Type, Object, JsonSerializer).</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of the object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonWriter, Object, JsonSerializer).</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonConverter](#)

└─ **BinaryConverter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [BinaryConverter](#) ►

### BinaryConverter()

Initializes a new instance of the [BinaryConverter](#) class

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public BinaryConverter()
```

```
Public Sub New
```

```
public:  
BinaryConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [BinaryConverter](#) ►

### CanConvert(Type)

Determines whether this instance can convert the specified object type.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override <a href="#">bool</a> CanConvert(     <a href="#">Type</a> <i>objectType</i> )</pre>		
	<pre>Public Overrides Function CanConvert (      <i>objectType</i> As <a href="#">Type</a>  ) As <a href="#">Boolean</a></pre>	
		<pre>public: virtual <a href="#">bool</a> CanConvert(     <a href="#">Type</a>^ <i>objectType</i> ) override</pre>

#### ▣ Parameters

##### ***objectType*** ([Type](#))

Type of the object.

#### ▣ Return Value

**true** if this instance can convert the specified object type; otherwise, **false**.

## ReadJson Method (reader, objectType, existingValue, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [BinaryConverter](#) ►

**ReadJson(JsonReader, Type, Object, JsonSerializer)**

Reads the JSON representation of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Object ReadJson(  
    JsonReader reader,  
    Type objectType,  
    Object existingValue,  
    JsonSerializer serializer  
)
```

```
Public Overrides Function ReadJson ( _  
    reader As JsonReader, _  
    objectType As Type, _  
    existingValue As Object, _  
    serializer As JsonSerializer _  
) As Object
```

```
public:  
virtual Object^ ReadJson(  
    JsonReader^ reader,  
    Type^ objectType,  
    Object^ existingValue,  
    JsonSerializer^ serializer  
) override
```

### ▣ Parameters

**reader** ([JsonReader](#))

The [JsonReader](#) to read from.

***objectType*** ([Type](#))

Type of the object.

***existingValue*** ([Object](#))

The existing value of object being read.

***serializer*** ([JsonSerializer](#))

The calling serializer.

#### ▣ **Return Value**

The object value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [BinaryConverter](#) ►

### WriteJson(JsonWriter, Object, JsonSerializer)

Writes the JSON representation of the object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteJson(  
    JsonWriter writer,  
    Object value,  
    JsonSerializer serializer  
)
```

```
Public Overrides Sub WriteJson ( _  
    writer As JsonWriter, _  
    value As Object, _  
    serializer As JsonSerializer _  
)
```

```
public:  
virtual void WriteJson(  
    JsonWriter^ writer,  
    Object^ value,  
    JsonSerializer^ serializer  
) override
```

#### ▣ Parameters

##### **writer** ([JsonWriter](#))

The [JsonWriter](#) to write to.

##### **value** ([Object](#))

The value.

***serializer* ([JsonSerializer](#))**

The calling serializer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

### BsonObjectIdConverter

Converts a [BsonObjectId](#) to and from JSON and BSON.

#### [-] Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public class BsonObjectIdConverter : <a href="#">JsonConverter</a></pre>		

```
Public Class BsonObjectIdConverter _
    Inherits JsonConverter
```

```
public ref class BsonObjectIdConverter : public Json
```

#### [-] Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">BsonObjectIdConverter()</a>	Initializes a new instance of the <b>BsonObjectIdConverter</b> class
	<a href="#">CanConvert(Type)</a>	Determines whether this instance can convert the specified object type.  (Overrides <a href="#">JsonConverter.CanConvert(Type)</a> .)

	<a href="#">CanRead</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can read JSON.  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">CanWrite</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can write JSON.  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetSchema()</a>	Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a> .  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ReadJson(JsonReader, Type, Object,</a>	Reads the JSON representation of the object.

	<a href="#">JsonSerializer</a> )	(Overrides <a href="#">JsonConverter.ReadJson(JsonReader Type, Object, JsonSerializer).</a> )
◆	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
◆	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of the object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonWriter, Object, JsonSerializer).</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonConverter](#)

└─ **BsonObjectIdConverter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [BsonObjectIdConverter](#) ► **BsonObjectIdConverter()**

Initializes a new instance of the [BsonObjectIdConverter](#) class

C#  
class

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public BsonObjectIdConverter()
```

```
Public Sub New
```

```
public:  
BsonObjectIdConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [BsonObjectIdConverter](#) ► **CanConvert(Type)**

Determines whether this instance can convert the specified object type.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override <a href="#">bool</a> CanConvert(     <a href="#">Type</a> objectType )</pre>		
	<pre>Public Overrides Function CanConvert (      <a href="#">objectType</a> As <a href="#">Type</a>  ) As <a href="#">Boolean</a></pre>	
		<pre>public: virtual <a href="#">bool</a> CanConvert(     <a href="#">Type</a>^ objectType ) override</pre>

### ▣ Parameters

***objectType*** ([Type](#))

Type of the object.

### ▣ Return Value

**true** if this instance can convert the specified object type; otherwise, **false**.

## ReadJson Method (reader, objectType, existingValue, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[BsonObjectIdConverter](#) ► **ReadJson(JsonReader, Type, Object, JsonSerializer)**

Reads the JSON representation of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Object ReadJson(  
    JsonReader reader,  
    Type objectType,  
    Object existingValue,  
    JsonSerializer serializer  
)
```

```
Public Overrides Function ReadJson ( _  
    reader As JsonReader, _  
    objectType As Type, _  
    existingValue As Object, _  
    serializer As JsonSerializer _  
) As Object
```

```
public:  
virtual Object^ ReadJson(  
    JsonReader^ reader,  
    Type^ objectType,  
    Object^ existingValue,  
    JsonSerializer^ serializer  
) override
```

### ▣ Parameters

**reader** ([JsonReader](#))

The [JsonReader](#) to read from.

***objectType* ([Type](#))**

Type of the object.

***existingValue* ([Object](#))**

The existing value of object being read.

***serializer* ([JsonSerializer](#))**

The calling serializer.

▣ **Return Value**

The object value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [BsonObjectIdConverter](#) ► **WriteJson(JsonWriter, Object, JsonSerializer)**

Writes the JSON representation of the object.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteJson(  
    JsonWriter writer,  
    Object value,  
    JsonSerializer serializer  
)
```

```
Public Overrides Sub WriteJson ( _  
    writer As JsonWriter, _  
    value As Object, _  
    serializer As JsonSerializer _  
)
```

```
public:  
virtual void WriteJson(  
    JsonWriter^ writer,  
    Object^ value,  
    JsonSerializer^ serializer  
) override
```

## ▣ Parameters

**writer** ([JsonWriter](#))

The [JsonWriter](#) to write to.

**value** ([Object](#))

The value.

***serializer* ([JsonSerializer](#))**

The calling serializer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

Documentation **CustomCreationConverter<T> Class**

[Namespaces](#) ▶ [Newtonsoft.Json.Converters](#) ▶

**CustomCreationConverter<T>**

Create a custom object

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
----	--------------	------------

```
public abstract class CustomCreationConverter<T> : J
```

```
Public MustInherit Class CustomCreationConverter(Of
    Inherits JsonSerializer
```

```
generic<typename T>
public ref class CustomCreationConverter abstract :
```

▣ **Generic Template Parameters**

*T*

[Missing <typeparam name="T"/> documentation for "T:Newtonsoft.Json.Converters.CustomCreationConverter`1"]

▣ **Members**

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declared <input checked="" type="checkbox"/> Inherited
<input checked="" type="checkbox"/> Protected			
Icon	Member	Description	
	<a href="#">CustomCreationConverter&lt;T&gt;()</a>	Initializes a new instance of the <b>CustomCreationConverter&lt;T&gt;</b>	

	<a href="#">CanConvert(Type)</a>	<p>Determines whether this instance can convert the specified object type</p> <p>(Overrides <a href="#">JsonConverter.CanConvert(Type)</a>)</p>
	<a href="#">CanRead</a>	<p>Gets a value indicating whether <a href="#">JsonConverter</a> can read JSON.</p> <p>(Inherited from <a href="#">JsonConverter</a>.)</p>
	<a href="#">CanWrite</a>	<p>Gets a value indicating whether <a href="#">JsonConverter</a> can write JSON.</p> <p>(Overrides <a href="#">JsonConverter.CanV</a>)</p>
	<a href="#">Create(Type)</a>	<p>Creates an object which will be populated by the serializer.</p>
	<a href="#">Equals(Object)</a>	<p>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">Finalize()</a>	<p>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetHashCode()</a>	<p>Serves as a hash function for a particular type.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetSchema()</a>	<p>Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a>.</p> <p>(Inherited from <a href="#">JsonConverter</a>.)</p>

	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">ReadJson(JsonReader, Type, Object, JsonSerializer)</a>	Reads the JSON representation of the object.  (Overrides <a href="#">JsonConverter.ReadJson(JsonReader, Type, Object, JsonSerializer)</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of the object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonWriter, Object, JsonSerializer)</a> .)

## ▣ Inheritance Hierarchy

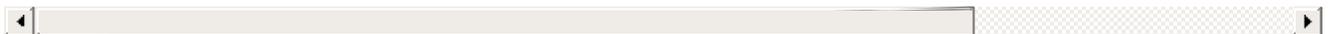
[Object](#)

└─ [JsonConverter](#)

└─ **CustomCreationConverter<T>**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[CustomCreationConverter<T>](#) ► **CustomCreationConverter<T>()**

Initializes a new instance of the [CustomCreationConverter<T>](#) class

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected CustomCreationConverter()
```

```
Protected Sub New
```

```
protected:  
CustomCreationConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[CustomCreationConverter<T>](#) ► **CanConvert(Type)**

Determines whether this instance can convert the specified object type.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override bool CanConvert(  
    Type objectType  
)
```

```
Public Overrides Function CanConvert ( _  
    objectType As Type _  
) As Boolean
```

```
public:  
virtual bool CanConvert(  
    Type^ objectType  
) override
```

### ▣ Parameters

***objectType*** ([Type](#))

Type of the object.

### ▣ Return Value

**true** if this instance can convert the specified object type; otherwise, **false**.

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [CustomCreationConverter<T>](#) ► **CanWrite**

Gets a value indicating whether this [JsonConverter](#) can write JSON.

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public override bool CanWrite { get; }
```

```
Public Overrides ReadOnly Property CanWrite As Boolean  
Get
```

```
public:  
virtual property bool CanWrite {  
    bool get () override;  
}
```

## Value

`true` if this [JsonConverter](#) can write JSON; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[CustomCreationConverter<T>](#) ► **Create(Type)**

Creates an object which will then be populated by the serializer.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public abstract T Create(     Type objectType )</pre>		
	<pre>Public MustOverride Function Create ( _     objectType As Type _ ) As T</pre>	
		<pre>public: virtual T Create(     Type^ objectType ) abstract</pre>

### ▣ Parameters

**objectType** ([Type](#))

Type of the object.

### ▣ Return Value

[Missing <returns> documentation for

"M:Newtonsoft.Json.Converters.CustomCreationConverter`1.Crea



## ReadJson Method (reader, objectType, existingValue, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[CustomCreationConverter<T>](#) ► **ReadJson(JsonReader, Type, Object, JsonSerializer)**

Reads the JSON representation of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Object ReadJson(  
    JsonReader reader,  
    Type objectType,  
    Object existingValue,  
    JsonSerializer serializer  
)
```

```
Public Overrides Function ReadJson ( _  
    reader As JsonReader, _  
    objectType As Type, _  
    existingValue As Object, _  
    serializer As JsonSerializer _  
) As Object
```

```
public:  
virtual Object^ ReadJson(  
    JsonReader^ reader,  
    Type^ objectType,  
    Object^ existingValue,  
    JsonSerializer^ serializer  
) override
```

### ▣ Parameters

*reader* ([JsonReader](#))

The [JsonReader](#) to read from.

***objectType* ([Type](#))**

Type of the object.

***existingValue* ([Object](#))**

The existing value of object being read.

***serializer* ([JsonSerializer](#))**

The calling serializer.

▣ **Return Value**

The object value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[CustomCreationConverter<T>](#) ► **WriteJson(JsonWriter, Object, JsonSerializer)**

Writes the JSON representation of the object.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteJson(  
    JsonWriter writer,  
    Object value,  
    JsonSerializer serializer  
)
```

```
Public Overrides Sub WriteJson ( _  
    writer As JsonWriter, _  
    value As Object, _  
    serializer As JsonSerializer _  
)
```

```
public:  
virtual void WriteJson(  
    JsonWriter^ writer,  
    Object^ value,  
    JsonSerializer^ serializer  
) override
```

## ▣ Parameters

**writer** ([JsonWriter](#))

The [JsonWriter](#) to write to.

**value** ([Object](#))

The value.

***serializer* ([JsonSerializer](#))**

The calling serializer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

Converts a [DataSet](#) to and from JSON.

## Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class DataSetConverter : JsonConverter
```

```
Public Class DataSetConverter _
    Inherits JsonConverter
```

```
public ref class DataSetConverter : public JsonConve
```

## Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherite

Icon	Member	Description
	<a href="#">DataSetConverter()</a>	Initializes a new instance of the <b>DataSetConverter</b> class
	<a href="#">CanConvert(Type)</a>	Determines whether this instance can convert the specified value type.  (Overrides <a href="#">JsonConverter.CanConvert(Type)</a> .)
	<a href="#">CanRead</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can read JSON.

		(Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">CanWrite</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can write JSON.  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetSchema()</a>	Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a> .  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ReadJson(JsonReader, Type, Object, JsonSerializer)</a>	Reads the JSON representation of the object.  (Overrides <a href="#">JsonConverter.ReadJson(JsonReader</a>

		<a href="#">Type, Object, JsonSerializer).</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of the object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonWriter, Object, JsonSerializer)</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonConverter](#)

└─ **DataSetConverter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [DataSetConverter](#) ►

### DataSetConverter()

Initializes a new instance of the [DataSetConverter](#) class

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public DataSetConverter()
```

```
Public Sub New
```

```
public:  
DataSetConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (valueType)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [DataSetConverter](#) ►

### CanConvert(Type)

Determines whether this instance can convert the specified value type.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override <a href="#">bool</a> CanConvert(     <a href="#">Type</a> valueType )</pre>		

```
Public Overrides Function CanConvert ( _  
    valueType As Type _  
) As Boolean
```

```
public:  
virtual bool CanConvert(  
    Type^ valueType  
) override
```

#### ▣ Parameters

##### **valueType (Type)**

Type of the value.

#### ▣ Return Value

**true** if this instance can convert the specified value type; otherwise, **false**.

## ReadJson Method (reader, objectType, existingValue, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [DataSetConverter](#) ►

**ReadJson(JsonReader, Type, Object, JsonSerializer)**

C#

Reads the JSON representation of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Object ReadJson(  
    JsonReader reader,  
    Type objectType,  
    Object existingValue,  
    JsonSerializer serializer  
)
```

```
Public Overrides Function ReadJson ( _  
    reader As JsonReader, _  
    objectType As Type, _  
    existingValue As Object, _  
    serializer As JsonSerializer _  
) As Object
```

```
public:  
virtual Object^ ReadJson(  
    JsonReader^ reader,  
    Type^ objectType,  
    Object^ existingValue,  
    JsonSerializer^ serializer  
) override
```

### ▣ Parameters

**reader** ([JsonReader](#))

The [JsonReader](#) to read from.

***objectType*** ([Type](#))

Type of the object.

***existingValue*** ([Object](#))

The existing value of object being read.

***serializer*** ([JsonSerializer](#))

The calling serializer.

#### ▣ **Return Value**

The object value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [DataSetConverter](#) ►

### WriteJson(JsonWriter, Object, JsonSerializer)

C#

Writes the JSON representation of the object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteJson(  
    JsonWriter writer,  
    Object value,  
    JsonSerializer serializer  
)
```

```
Public Overrides Sub WriteJson ( _  
    writer As JsonWriter, _  
    value As Object, _  
    serializer As JsonSerializer _  
)
```

```
public:  
virtual void WriteJson(  
    JsonWriter^ writer,  
    Object^ value,  
    JsonSerializer^ serializer  
) override
```

#### ▣ Parameters

##### **writer** ([JsonWriter](#))

The [JsonWriter](#) to write to.

##### **value** ([Object](#))

The value.

***serializer* ([JsonSerializer](#))**

The calling serializer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► **DataTableConverter**

Converts a [DataTable](#) to and from JSON.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class DataTableConverter : JsonConverter
```

```
Public Class DataTableConverter _  
    Inherits JsonConverter
```

```
public ref class DataTableConverter : public JsonCon
```

### Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declare <input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">DataTableConverter()</a>	Initializes a new instance of the <b>DataTableConverter</b> class
	<a href="#">CanConvert(Type)</a>	Determines whether this instance can convert the specified value type.  (Overrides <a href="#">JsonConverter.CanConvert(Type)</a> .)
	<a href="#">CanRead</a>	Gets a value indicating whether this

		<p><a href="#">JsonConverter</a> can read JSON.</p> <p>(Inherited from <a href="#">JsonConverter</a>.)</p>
	<a href="#">CanWrite</a>	<p>Gets a value indicating whether this <a href="#">JsonConverter</a> can write JSON.</p> <p>(Inherited from <a href="#">JsonConverter</a>.)</p>
	<a href="#">Equals(Object)</a>	<p>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">Finalize()</a>	<p>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetHashCode()</a>	<p>Serves as a hash function for a particular type.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetSchema()</a>	<p>Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a>.</p> <p>(Inherited from <a href="#">JsonConverter</a>.)</p>
	<a href="#">GetType()</a>	<p>Gets the <a href="#">Type</a> of the current instance.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">MemberwiseClone()</a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">ReadJson(JsonReader, Type, Object, JsonSerializer)</a>	<p>Reads the JSON representation of the object.</p>

		(Overrides <a href="#">JsonConverter.ReadJson(JsonReader.Type, Object, JsonSerializer).</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of the object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonWriter, Object, JsonSerializer).</a> )

## ▣ Inheritance Hierarchy

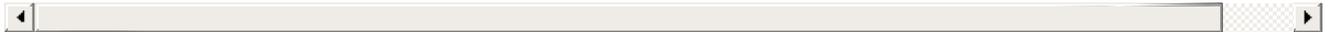
[Object](#)

└─ [JsonConverter](#)

└─ **DataTableConverter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [DataTableConverter](#) ►

### DataTableConverter()

Initializes a new instance of the [DataTableConverter](#) class

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public DataTableConverter()</pre>		
	<pre>Public Sub New</pre>	
		<pre>public: DataTableConverter()</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (valueType)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [DataTableConverter](#) ►

### CanConvert(Type)

Determines whether this instance can convert the specified value type. C#

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override bool CanConvert(  
    Type valueType  
)
```

```
Public Overrides Function CanConvert ( _  
    valueType As Type _  
) As Boolean
```

```
public:  
virtual bool CanConvert(  
    Type^ valueType  
) override
```

#### ▣ Parameters

**valueType (Type)**

Type of the value.

#### ▣ Return Value

**true** if this instance can convert the specified value type; otherwise, **false**.

## ReadJson Method (reader, objectType, existingValue, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [DataTableConverter](#) ►

**ReadJson(JsonReader, Type, Object, JsonSerializer)**

C#

Reads the JSON representation of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Object ReadJson(  
    JsonReader reader,  
    Type objectType,  
    Object existingValue,  
    JsonSerializer serializer  
)
```

```
Public Overrides Function ReadJson ( _  
    reader As JsonReader, _  
    objectType As Type, _  
    existingValue As Object, _  
    serializer As JsonSerializer _  
) As Object
```

```
public:  
virtual Object^ ReadJson(  
    JsonReader^ reader,  
    Type^ objectType,  
    Object^ existingValue,  
    JsonSerializer^ serializer  
) override
```

### ▣ Parameters

**reader** ([JsonReader](#))

The [JsonReader](#) to read from.

***objectType*** ([Type](#))

Type of the object.

***existingValue*** ([Object](#))

The existing value of object being read.

***serializer*** ([JsonSerializer](#))

The calling serializer.

#### ▣ **Return Value**

The object value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [DataTableConverter](#) ►

### WriteJson(JsonWriter, Object, JsonSerializer)

Writes the JSON representation of the object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteJson(  
    JsonWriter writer,  
    Object value,  
    JsonSerializer serializer  
)
```

```
Public Overrides Sub WriteJson ( _  
    writer As JsonWriter, _  
    value As Object, _  
    serializer As JsonSerializer _  
)
```

```
public:  
virtual void WriteJson(  
    JsonWriter^ writer,  
    Object^ value,  
    JsonSerializer^ serializer  
) override
```

#### ▣ Parameters

##### **writer** ([JsonWriter](#))

The [JsonWriter](#) to write to.

##### **value** ([Object](#))

The value.

***serializer* ([JsonSerializer](#))**

The calling serializer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

Json.NET - Quick Starts & API Documentation **DateTimeConverterBase**  
**Class**

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

### DateTimeConverterBase

Provides a base class for converting a [DateTime](#) to and from JSON. C# ▼

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public abstract class DateTimeConverterBase : JsonConvert
```

```
Public MustInherit Class DateTimeConverterBase _  
    Inherits JsonConvert
```

```
public ref class DateTimeConverterBase abstract : pu
```

#### ▣ Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declared <input checked="" type="checkbox"/> Inherited
Icon	Member	Description	
	<a href="#">DateTimeConverterBase()</a>	Initializes a new instance of the <b>DateTimeConverterBase</b> class	
	<a href="#">CanConvert(Type)</a>	Determines whether this instance can convert the specified object type.  (Overrides <a href="#">JsonConverter.CanConvert(Type)</a> .)	
			

	<a href="#">CanRead</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can read JSON.  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">CanWrite</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can write JSON.  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetSchema()</a>	Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a> .  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ReadJson(JsonReader, Type, Object,</a>	Reads the JSON representation of the object.

	<a href="#">JsonSerializer</a>	(Inherited from <a href="#">JsonConverter</a> .)
☰	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
☰	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of the object.  (Inherited from <a href="#">JsonConverter</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonConverter](#)

└─ **DateTimeConverterBase**

└─ [IsoDateTimeConverter](#)

└─ [JavaScriptDateTimeConverter](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[DateTimeConverterBase](#) ► **DateTimeConverterBase()**

Initializes a new instance of the [DateTimeConverterBase](#) class

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected DateTimeConverterBase()
```

```
Protected Sub New
```

```
protected:  
DateTimeConverterBase()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [DateTimeConverterBase](#) ► **CanConvert(Type)**

Determines whether this instance can convert the specified object type.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override <a href="#">bool</a> CanConvert(     <a href="#">Type</a> <i>objectType</i> )</pre>		
	<pre>Public Overrides Function CanConvert (      <i>objectType</i> As <a href="#">Type</a>  ) As <a href="#">Boolean</a></pre>	
		<pre>public: virtual <a href="#">bool</a> CanConvert(     <a href="#">Type</a>^ <i>objectType</i> ) override</pre>

### ▣ Parameters

#### ***objectType*** ([Type](#))

Type of the object.

### ▣ Return Value

**true** if this instance can convert the specified object type; otherwise, **false**.

Documentation **EntityKeyMemberConverter Class**

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

**EntityKeyMemberConverter**

Converts an Entity Framework EntityKey to and from JSON.

▣ **Declaration Syntax**

```
C#
public class EntityKeyMemberConverter : JsonConvert
```

```
Public Class EntityKeyMemberConverter _
    Inherits JsonConvert
```

```
public ref class EntityKeyMemberConverter : public J
```

▣ **Members**

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declare <input checked="" type="checkbox"/> Inherite

Icon	Member	Description
	<a href="#">EntityKeyMemberConverter()</a>	Initializes a new instance of the <b>EntityKeyMemberConverter</b> clas
	<a href="#">CanConvert(Type)</a>	Determines whether this instance ( convert the specified object type.  (Overrides <a href="#">JsonConverter.CanConvert(Type)</a> ).
	<a href="#">CanRead</a>	Gets a value indicating whether thi

		<p><a href="#">JsonConverter</a> can read JSON.</p> <p>(Inherited from <a href="#">JsonConverter</a>.)</p>
	<a href="#">CanWrite</a>	<p>Gets a value indicating whether the <a href="#">JsonConverter</a> can write JSON.</p> <p>(Inherited from <a href="#">JsonConverter</a>.)</p>
	<a href="#">Equals(Object)</a>	<p>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">Finalize()</a>	<p>Allows an <a href="#">Object</a> to attempt to free resources and perform other clean operations before the <a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetHashCode()</a>	<p>Serves as a hash function for a particular type.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetSchema()</a>	<p>Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a>.</p> <p>(Inherited from <a href="#">JsonConverter</a>.)</p>
	<a href="#">GetType()</a>	<p>Gets the <a href="#">Type</a> of the current instance.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">MemberwiseClone()</a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">ReadJson(JsonReader, Type, Object, JsonSerializer)</a>	<p>Reads the JSON representation of an object.</p>

		(Overrides <a href="#">JsonConverter.ReadJson(JsonRe: Type, Object, JsonSerializer).</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents th current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonWri Object, JsonSerializer).</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonConverter](#)

└─ **EntityKeyMemberConverter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



Documentation **EntityKeyMemberConverter Constructor**

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[EntityKeyMemberConverter](#) ► **EntityKeyMemberConverter()**

Initializes a new instance of the [EntityKeyMemberConverter](#) class

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public EntityKeyMemberConverter()
```

```
Public Sub New
```

```
public:  
EntityKeyMemberConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [EntityKeyMemberConverter](#) ► **CanConvert(Type)**

Determines whether this instance can convert the specified object type.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override <a href="#">bool</a> CanConvert(     <a href="#">Type</a> <i>objectType</i> )</pre>		
	<pre>Public Overrides Function CanConvert (      <i>objectType</i> As <a href="#">Type</a>  ) As <a href="#">Boolean</a></pre>	
		<pre>public: virtual <a href="#">bool</a> CanConvert(     <a href="#">Type</a>^ <i>objectType</i> ) override</pre>

### ▣ Parameters

#### ***objectType*** ([Type](#))

Type of the object.

### ▣ Return Value

**true** if this instance can convert the specified object type; otherwise, **false**.

## ReadJson Method (reader, objectType, existingValue, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[EntityKeyMemberConverter](#) ► **ReadJson(JsonReader, Type, Object, JsonSerializer)**

Reads the JSON representation of the object.

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Object ReadJson(  
    JsonReader reader,  
    Type objectType,  
    Object existingValue,  
    JsonSerializer serializer  
)
```

```
Public Overrides Function ReadJson ( _  
    reader As JsonReader, _  
    objectType As Type, _  
    existingValue As Object, _  
    serializer As JsonSerializer _  
) As Object
```

```
public:  
virtual Object^ ReadJson(  
    JsonReader^ reader,  
    Type^ objectType,  
    Object^ existingValue,  
    JsonSerializer^ serializer  
) override
```

### ▣ Parameters

**reader** ([JsonReader](#))

The [JsonReader](#) to read from.

***objectType* ([Type](#))**

Type of the object.

***existingValue* ([Object](#))**

The existing value of object being read.

***serializer* ([JsonSerializer](#))**

The calling serializer.

▣ **Return Value**

The object value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[EntityKeyMemberConverter](#) ► **WriteJson(JsonWriter, Object, JsonSerializer)**

Writes the JSON representation of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteJson(  
    JsonWriter writer,  
    Object value,  
    JsonSerializer serializer  
)
```

```
Public Overrides Sub WriteJson ( _  
    writer As JsonWriter, _  
    value As Object, _  
    serializer As JsonSerializer _  
)
```

```
public:  
virtual void WriteJson(  
    JsonWriter^ writer,  
    Object^ value,  
    JsonSerializer^ serializer  
) override
```

### ▣ Parameters

**writer** ([JsonWriter](#))

The [JsonWriter](#) to write to.

**value** ([Object](#))

The value.

***serializer* ([JsonSerializer](#))**

The calling serializer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

### ExpandableObjectConverter

Converts an ExpandableObject to and from JSON.

#### Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class ExpandableObjectConverter : JsonConverter
```

```
Public Class ExpandableObjectConverter _
    Inherits JsonConverter
```

```
public ref class ExpandableObjectConverter : public Json
```

#### Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">ExpandableObjectConverter()</a>	Initializes a new instance of the <b>ExpandableObjectConverter</b> class
	<a href="#">CanConvert(Type)</a>	Determines whether this instance can convert the specified object type.  (Overrides <a href="#">JsonConverter.CanConvert(Type)</a> .)

	<a href="#">CanRead</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can read JSON.  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">CanWrite</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can write JSON.  (Overrides <a href="#">JsonConverter.CanWrite</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetSchema()</a>	Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a> .  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ReadJson(JsonReader, Type, Object,</a>	Reads the JSON representation of the object.

	<a href="#">JsonSerializer</a> )	(Overrides <a href="#">JsonConverter.ReadJson(JsonReadType, Object, JsonSerializer).</a> )
◆	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
◆	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of the object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonWriter, Object, JsonSerializer).</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonConverter](#)

└─ **ExpandableObjectConverter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# ExpandableObjectConverter Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[ExpandableObjectConverter](#) ► **ExpandableObjectConverter()**

Initializes a new instance of the [ExpandableObjectConverter](#) class

C#

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public ExpandableObjectConverter()
```

```
Public Sub New
```

```
public:  
ExpandableObjectConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [ExpandoObjectConverter](#) ► **CanConvert(Type)**

Determines whether this instance can convert the specified object type.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override <a href="#">bool</a> CanConvert(     <a href="#">Type</a> <i>objectType</i> )</pre>		
	<pre>Public Overrides Function CanConvert (      <i>objectType</i> As <a href="#">Type</a>  ) As <a href="#">Boolean</a></pre>	
		<pre>public: virtual <a href="#">bool</a> CanConvert(     <a href="#">Type</a>^ <i>objectType</i> ) override</pre>

### ▣ Parameters

#### ***objectType*** ([Type](#))

Type of the object.

### ▣ Return Value

**true** if this instance can convert the specified object type; otherwise, **false**.

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [ExpandoObjectConverter](#) ► **CanWrite**

Gets a value indicating whether this [JsonConverter](#) can write JSON.

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public override bool CanWrite { get; }
```

```
Public Overrides ReadOnly Property CanWrite As Boole  
Get
```

```
public:  
virtual property bool CanWrite {  
    bool get () override;  
}
```

## Value

`true` if this [JsonConverter](#) can write JSON; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## ReadJson Method (reader, objectType, existingValue, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[ExpandableObjectConverter](#) ► **ReadJson(JsonReader, Type, Object, JsonSerializer)**

Reads the JSON representation of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Object ReadJson(  
    JsonReader reader,  
    Type objectType,  
    Object existingValue,  
    JsonSerializer serializer  
)
```

```
Public Overrides Function ReadJson ( _  
    reader As JsonReader, _  
    objectType As Type, _  
    existingValue As Object, _  
    serializer As JsonSerializer _  
) As Object
```

```
public:  
virtual Object^ ReadJson(  
    JsonReader^ reader,  
    Type^ objectType,  
    Object^ existingValue,  
    JsonSerializer^ serializer  
) override
```

### ▣ Parameters

*reader* ([JsonReader](#))

The [JsonReader](#) to read from.

***objectType* ([Type](#))**

Type of the object.

***existingValue* ([Object](#))**

The existing value of object being read.

***serializer* ([JsonSerializer](#))**

The calling serializer.

▣ **Return Value**

The object value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[ExpandoObjectConverter](#) ► **WriteJson(JsonWriter, Object, JsonSerializer)**

Writes the JSON representation of the object.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteJson(  
    JsonWriter writer,  
    Object value,  
    JsonSerializer serializer  
)
```

```
Public Overrides Sub WriteJson ( _  
    writer As JsonWriter, _  
    value As Object, _  
    serializer As JsonSerializer _  
)
```

```
public:  
virtual void WriteJson(  
    JsonWriter^ writer,  
    Object^ value,  
    JsonSerializer^ serializer  
) override
```

### Parameters

**writer** ([JsonWriter](#))

The [JsonWriter](#) to write to.

**value** ([Object](#))

The value.

***serializer* ([JsonSerializer](#))**

The calling serializer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

### IsoDateTimeConverter

C#

Converts a [DateTime](#) to and from the ISO 8601 date format (e.g. 2008-04-12T12:53Z).

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class IsoDateTimeConverter : DateTimeConverte
```

```
Public Class IsoDateTimeConverter _
    Inherits DateTimeConverterBase
```

```
public ref class IsoDateTimeConverter : public DateT
```

#### Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited
Icon	Member	Description	
	<a href="#">IsoDateTimeConverter()</a>	Initializes a new instance of the <b>IsoDateTimeConverter</b> class	
	<a href="#">CanConvert(Type)</a>	Determines whether this instance can convert the specified object type.  (Inherited from <a href="#">DateTimeConverterBase</a> .)	

	<a href="#">CanRead</a>	<p>Gets a value indicating whether this <a href="#">JsonConverter</a> can read JSON.</p> <p>(Inherited from <a href="#">JsonConverter</a>.)</p>
	<a href="#">CanWrite</a>	<p>Gets a value indicating whether this <a href="#">JsonConverter</a> can write JSON.</p> <p>(Inherited from <a href="#">JsonConverter</a>.)</p>
	<a href="#">Culture</a>	<p>Gets or sets the culture used when converting a date to and from JSON.</p>
	<a href="#">DateTimeFormat</a>	<p>Gets or sets the date time format used when converting a date to and from JSON.</p>
	<a href="#">DateTimeStyles</a>	<p>Gets or sets the date time styles used when converting a date to and from JSON.</p>
	<a href="#">Equals(Object)</a>	<p>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">Finalize()</a>	<p>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetHashCode()</a>	<p>Serves as a hash function for a particular type.</p>

		(Inherited from <a href="#">Object</a> .)
	<a href="#">GetSchema()</a>	Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a> .  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ReadJson(JsonReader, Type, Object, JsonSerializer)</a>	Reads the JSON representation of the object.  (Overrides <a href="#">JsonConverter.ReadJson(JsonReader, Type, Object, JsonSerializer)</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of the object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonWriter, Object, JsonSerializer)</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

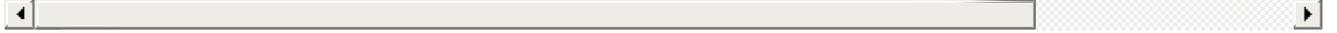
└─ [JsonConverter](#)

└─ [DateTimeConverterBase](#)

└─ **IsoDateTimeConverter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [IsoDateTimeConverter](#)

### ► IsoDateTimeConverter()

Initializes a new instance of the [IsoDateTimeConverter](#) class

C#

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IsoDateTimeConverter()
```

```
Public Sub New
```

```
public:  
IsoDateTimeConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [IsoDateTimeConverter](#)

## ► Culture

Gets or sets the culture used when converting a date to and from JSON

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public CultureInfo Culture { get; set; }
```

```
Public Property Culture As CultureInfo  
    Get  
    Set
```

```
public:  
property CultureInfo^ Culture {  
    CultureInfo^ get ();  
    void set (CultureInfo^ value);  
}
```

### ▣ Value

The culture used when converting a date to and from JSON.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [IsoDateTimeConverter](#)

### ► DateTimeFormat

Gets or sets the date time format used when converting a date to and from JSON.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string DateTimeFormat { get; set; }
```

```
Public Property DateTimeFormat As String  
    Get  
    Set
```

```
public:  
property String^ DateTimeFormat {  
    String^ get ();  
    void set (String^ value);  
}
```

#### ▣ Value

The date time format used when converting a date to and from JSON.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [IsoDateTimeConverter](#)

## ► DateTimeStyles

Gets or sets the date time styles used when converting a date to and from JSON.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public DateTimeStyles DateTimeStyles { get; set; }
```

```
Public Property DateTimeStyles As DateTimeStyles  
    Get  
    Set
```

```
public:  
property DateTimeStyles DateTimeStyles {  
    DateTimeStyles get ();  
    void set (DateTimeStyles value);  
}
```

### ▣ Value

The date time styles used when converting a date to and from JSON.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## ReadJson Method (reader, objectType, existingValue, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [IsoDateTimeConverter](#)

### ► ReadJson(JsonReader, Type, Object, JsonSerializer)

Reads the JSON representation of the object.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override <a href="#">Object</a> ReadJson(     <a href="#">JsonReader</a> reader,     <a href="#">Type</a> objectType,     <a href="#">Object</a> existingValue,     <a href="#">JsonSerializer</a> serializer )</pre>		
	<pre>Public Overrides Function ReadJson ( _     reader As <a href="#">JsonReader</a>, _     objectType As <a href="#">Type</a>, _     existingValue As <a href="#">Object</a>, _     serializer As <a href="#">JsonSerializer</a> _ ) As <a href="#">Object</a></pre>	
		<pre>public: virtual <a href="#">Object</a>^ ReadJson(     <a href="#">JsonReader</a>^ reader,     <a href="#">Type</a>^ objectType,     <a href="#">Object</a>^ existingValue,     <a href="#">JsonSerializer</a>^ serializer ) override</pre>

#### ▣ Parameters

##### **reader** ([JsonReader](#))

The [JsonReader](#) to read from.

***objectType*** ([Type](#))

Type of the object.

***existingValue*** ([Object](#))

The existing value of object being read.

***serializer*** ([JsonSerializer](#))

The calling serializer.

#### ▣ **Return Value**

The object value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [IsoDateTimeConverter](#)

### ► WriteJson(JsonWriter, Object, JsonSerializer)

Writes the JSON representation of the object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteJson(  
    JsonWriter writer,  
    Object value,  
    JsonSerializer serializer  
)
```

```
Public Overrides Sub WriteJson ( _  
    writer As JsonWriter, _  
    value As Object, _  
    serializer As JsonSerializer _  
)
```

```
public:  
virtual void WriteJson(  
    JsonWriter^ writer,  
    Object^ value,  
    JsonSerializer^ serializer  
) override
```

#### ▣ Parameters

##### **writer** ([JsonWriter](#))

The [JsonWriter](#) to write to.

##### **value** ([Object](#))

The value.

***serializer* ([JsonSerializer](#))**

The calling serializer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

Documentation **JavaScriptDateTimeConverter Class**

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

**JavaScriptDateTimeConverter**

Converts a [DateTime](#) to and from a JavaScript date constructor (e.g. new Date(52231943)).

[-] **Declaration Syntax**

C# Visual Basic Visual C++

```
public class JavaScriptDateTimeConverter : DateTimeC
```

```
Public Class JavaScriptDateTimeConverter _
    Inherits DateTimeConverterBase
```

```
public ref class JavaScriptDateTimeConverter : publi
```

[-] **Members**

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
			<input checked="" type="checkbox"/> Declare
			<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JavaScriptDateTimeConverter()</a>	Initializes a new instance of the <b>JavaScriptDateTimeConverter</b>
	<a href="#">CanConvert(Type)</a>	Determines whether this instance can convert the specified object type  (Inherited from <a href="#">DateTimeConverterBase</a> .)

	<a href="#">CanRead</a>	Gets a value indicating whether <a href="#">JsonConverter</a> can read JSON.  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">CanWrite</a>	Gets a value indicating whether <a href="#">JsonConverter</a> can write JSON.  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specific <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetSchema()</a>	Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a> .  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ReadJson(JsonReader, Type, Object, JsonSerializer)</a>	Reads the JSON representation of an object.

		(Overrides <a href="#">JsonConverter.ReadJson(JsonF Type, Object, JsonSerializer).</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonV Object, JsonSerializer).</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonConverter](#)

└─ [DateTimeConverterBase](#)

└─ **JavaScriptDateTimeConverter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[JavaScriptDateTimeConverter](#) ► **JavaScriptDateTimeConverter()**

Initializes a new instance of the [JavaScriptDateTimeConverter](#) class

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JavaScriptDateTimeConverter()
```

```
Public Sub New
```

```
public:  
JavaScriptDateTimeConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## ReadJson Method (reader, objectType, existingValue, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[JavaScriptDateTimeConverter](#) ► **ReadJson(JsonReader, Type, Object, JsonSerializer)**

Reads the JSON representation of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Object ReadJson(  
    JsonReader reader,  
    Type objectType,  
    Object existingValue,  
    JsonSerializer serializer  
)
```

```
Public Overrides Function ReadJson ( _  
    reader As JsonReader, _  
    objectType As Type, _  
    existingValue As Object, _  
    serializer As JsonSerializer _  
) As Object
```

```
public:  
virtual Object^ ReadJson(  
    JsonReader^ reader,  
    Type^ objectType,  
    Object^ existingValue,  
    JsonSerializer^ serializer  
) override
```

### ▣ Parameters

*reader* ([JsonReader](#))

The [JsonReader](#) to read from.

***objectType* ([Type](#))**

Type of the object.

***existingValue* ([Object](#))**

The existing property value of the JSON that is being converted.

***serializer* ([JsonSerializer](#))**

The calling serializer.

▣ **Return Value**

The object value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[JavaScriptDateTimeConverter](#) ► **WriteJson(JsonWriter, Object, JsonSerializer)**

Writes the JSON representation of the object.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteJson(  
    JsonWriter writer,  
    Object value,  
    JsonSerializer serializer  
)
```

```
Public Overrides Sub WriteJson ( _  
    writer As JsonWriter, _  
    value As Object, _  
    serializer As JsonSerializer _  
)
```

```
public:  
virtual void WriteJson(  
    JsonWriter^ writer,  
    Object^ value,  
    JsonSerializer^ serializer  
) override
```

## ▣ Parameters

**writer** ([JsonWriter](#))

The [JsonWriter](#) to write to.

**value** ([Object](#))

The value.

***serializer* ([JsonSerializer](#))**

The calling serializer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# KeyValuePairConverter

## Class

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

### KeyValuePairConverter

Converts a [KeyValuePair<TKey, TValue>](#) to and from JSON.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class KeyValuePairConverter : JsonConverter
```

```
Public Class KeyValuePairConverter _  
    Inherits JsonConverter
```

```
public ref class KeyValuePairConverter : public Json
```

#### Members

All Members

Constructors

Methods

Properties

<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">KeyValuePairConverter()</a>	Initializes a new instance of the <b>KeyValuePairConverter</b> class
	<a href="#">CanConvert(Type)</a>	Determines whether this instance can convert the specified object type.  (Overrides <a href="#">JsonConverter.CanConvert(Type)</a> .)

	<a href="#">CanRead</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can read JSON.  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">CanWrite</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can write JSON.  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetSchema()</a>	Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a> .  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ReadJson(JsonReader, Type, Object,</a>	Reads the JSON representation of the object.

	<a href="#">JsonSerializer</a> )	(Overrides <a href="#">JsonConverter.ReadJson(JsonReader Type, Object, JsonSerializer).</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of the object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonWriter, Object, JsonSerializer).</a> )

## ▣ Inheritance Hierarchy

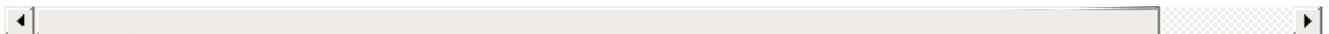
[Object](#)

└─ [JsonConverter](#)

└─ **KeyValuePairConverter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# KeyValuePairConverter

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[KeyValuePairConverter](#) ► **KeyValuePairConverter()**

Initializes a new instance of the [KeyValuePairConverter](#) class

C#  
class

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public KeyValuePairConverter()
```

```
Public Sub New
```

```
public:  
KeyValuePairConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [KeyValuePairConverter](#) ► **CanConvert(Type)**

Determines whether this instance can convert the specified object type.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override <a href="#">bool</a> CanConvert(     <a href="#">Type</a> objectType )</pre>		
	<pre>Public Overrides Function CanConvert (      <a href="#">objectType</a> As <a href="#">Type</a>  ) As <a href="#">Boolean</a></pre>	
		<pre>public: virtual <a href="#">bool</a> CanConvert(     <a href="#">Type</a>^ objectType ) override</pre>

### ▣ Parameters

***objectType*** ([Type](#))

Type of the object.

### ▣ Return Value

**true** if this instance can convert the specified object type; otherwise, **false**.

## ReadJson Method (reader, objectType, existingValue, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

[KeyValuePairConverter](#) ► **ReadJson(JsonReader, Type, Object, JsonSerializer)**

Reads the JSON representation of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Object ReadJson(  
    JsonReader reader,  
    Type objectType,  
    Object existingValue,  
    JsonSerializer serializer  
)
```

```
Public Overrides Function ReadJson ( _  
    reader As JsonReader, _  
    objectType As Type, _  
    existingValue As Object, _  
    serializer As JsonSerializer _  
) As Object
```

```
public:  
virtual Object^ ReadJson(  
    JsonReader^ reader,  
    Type^ objectType,  
    Object^ existingValue,  
    JsonSerializer^ serializer  
) override
```

### ▣ Parameters

**reader** ([JsonReader](#))

The [JsonReader](#) to read from.

***objectType* ([Type](#))**

Type of the object.

***existingValue* ([Object](#))**

The existing value of object being read.

***serializer* ([JsonSerializer](#))**

The calling serializer.

▣ **Return Value**

The object value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [KeyValuePairConverter](#) ► **WriteJson(JsonWriter, Object, JsonSerializer)**

Writes the JSON representation of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteJson(  
    JsonWriter writer,  
    Object value,  
    JsonSerializer serializer  
)
```

```
Public Overrides Sub WriteJson ( _  
    writer As JsonWriter, _  
    value As Object, _  
    serializer As JsonSerializer _  
)
```

```
public:  
virtual void WriteJson(  
    JsonWriter^ writer,  
    Object^ value,  
    JsonSerializer^ serializer  
) override
```

### ▣ Parameters

**writer** ([JsonWriter](#))

The [JsonWriter](#) to write to.

**value** ([Object](#))

The value.

***serializer* ([JsonSerializer](#))**

The calling serializer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# RegexConverter Class

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► **RegexConverter**

Converts a [Regex](#) to and from JSON and BSON.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public class RegexConverter : JsonConverter
```

```
Public Class RegexConverter _  
    Inherits JsonConverter
```

```
public ref class RegexConverter : public JsonConvert
```

## Members

All Members

Constructors

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declared
- Inherited

Icon	Member	Description
	<a href="#">RegexConverter()</a>	Initializes a new instance of the <b>RegexConverter</b> class
	<a href="#">CanConvert(Type)</a>	Determines whether this instance can convert the specified object type.  (Overrides <a href="#">JsonConverter.CanConvert(Type)</a> .)
	<a href="#">CanRead</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can read JSON.

		(Inherited from <a href="#">JsonConverter.</a> )
	<a href="#">CanWrite</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can write JSON.  (Inherited from <a href="#">JsonConverter.</a> )
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object.</a> )
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object.</a> )
	<a href="#">GetSchema()</a>	Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a> .  (Inherited from <a href="#">JsonConverter.</a> )
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object.</a> )
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">ReadJson(JsonReader, Type, Object, JsonSerializer)</a>	Reads the JSON representation of the object.  (Overrides <a href="#">JsonConverter.ReadJson(JsonReader</a>

		<a href="#">Type, Object, JsonSerializer).</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of the object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonWriter, Object, JsonSerializer)</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonConverter](#)

└─ **RegexConverter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [RegexConverter](#) ►

### RegexConverter()

Initializes a new instance of the [RegexConverter](#) class

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public RegexConverter()
```

```
Public Sub New
```

```
public:  
RegexConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [RegexConverter](#) ►

### CanConvert(Type)

Determines whether this instance can convert the specified object type.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override <a href="#">bool</a> CanConvert(     <a href="#">Type</a> objectType )</pre>		
	<pre>Public Overrides Function CanConvert (      <a href="#">objectType</a> As <a href="#">Type</a>  ) As <a href="#">Boolean</a></pre>	
		<pre>public: virtual <a href="#">bool</a> CanConvert(     <a href="#">Type</a>^ objectType ) override</pre>

#### ▣ Parameters

##### ***objectType*** ([Type](#))

Type of the object.

#### ▣ Return Value

**true** if this instance can convert the specified object type; otherwise, **false**.

## ReadJson Method (reader, objectType, existingValue, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [RegexConverter](#) ►

**ReadJson(JsonReader, Type, Object, JsonSerializer)**

Reads the JSON representation of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Object ReadJson(  
    JsonReader reader,  
    Type objectType,  
    Object existingValue,  
    JsonSerializer serializer  
)
```

```
Public Overrides Function ReadJson ( _  
    reader As JsonReader, _  
    objectType As Type, _  
    existingValue As Object, _  
    serializer As JsonSerializer _  
) As Object
```

```
public:  
virtual Object^ ReadJson(  
    JsonReader^ reader,  
    Type^ objectType,  
    Object^ existingValue,  
    JsonSerializer^ serializer  
) override
```

### ▣ Parameters

**reader** ([JsonReader](#))

The [JsonReader](#) to read from.

***objectType*** ([Type](#))

Type of the object.

***existingValue*** ([Object](#))

The existing value of object being read.

***serializer*** ([JsonSerializer](#))

The calling serializer.

#### ▣ **Return Value**

The object value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [RegexConverter](#) ►

### WriteJson(JsonWriter, Object, JsonSerializer)

Writes the JSON representation of the object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteJson(  
    JsonWriter writer,  
    Object value,  
    JsonSerializer serializer  
)
```

```
Public Overrides Sub WriteJson ( _  
    writer As JsonWriter, _  
    value As Object, _  
    serializer As JsonSerializer _  
)
```

```
public:  
virtual void WriteJson(  
    JsonWriter^ writer,  
    Object^ value,  
    JsonSerializer^ serializer  
) override
```

#### ▣ Parameters

##### **writer** ([JsonWriter](#))

The [JsonWriter](#) to write to.

##### **value** ([Object](#))

The value.

***serializer* ([JsonSerializer](#))**

The calling serializer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►

### StringEnumConverter

Converts an [Enum](#) to and from its name string value.

Converts an [Enum](#) to and from its name string value.

#### Declaration Syntax

C#	Visual Basic	Visual C++
<code>public class StringEnumConverter : <a href="#">JsonConverter</a></code>		

```
Public Class StringEnumConverter _
    Inherits JsonConverter
```

```
public ref class StringEnumConverter : public JsonCo
```

#### Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declare <input checked="" type="checkbox"/> Inherite

Icon	Member	Description
	<a href="#">StringEnumConverter()</a>	Initializes a new instance of the <b>StringEnumConverter</b> class
	<a href="#">CamelCaseText</a>	Gets or sets a value indicating whether the written enum text should be camel case.

	<a href="#">CanConvert(Type)</a>	Determines whether this instance can convert the specified object type.  (Overrides <a href="#">JsonConverter.CanConvert(Type)</a> .)
	<a href="#">CanRead</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can read JSON.  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">CanWrite</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can write JSON.  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetSchema()</a>	Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a> .  (Inherited from <a href="#">JsonConverter</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)

	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ReadJson(JsonReader, Type, Object, JsonSerializer)</a>	Reads the JSON representation of the object.  (Overrides <a href="#">JsonConverter.ReadJson(JsonReader, Type, Object, JsonSerializer)</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of the object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonWriter, Object, JsonSerializer)</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonConverter](#)

└─ **StringEnumConverter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [StringEnumConverter](#)

### ► StringEnumConverter()

Initializes a new instance of the [StringEnumConverter](#) class

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public StringEnumConverter()</pre>		
	<pre>Public Sub New</pre>	
		<pre>public: StringEnumConverter()</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [StringEnumConverter](#)

## ► CamelCaseText

Gets or sets a value indicating whether the written enum text should be camel case.

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool CamelCaseText { get; set; }
```

```
Public Property CamelCaseText As Boolean  
    Get  
    Set
```

```
public:  
property bool CamelCaseText {  
    bool get ();  
    void set (bool value);  
}
```

### ▣ Value

`true` if the written enum text will be camel case; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [StringEnumConverter](#)

### ► CanConvert(Type)

Determines whether this instance can convert the specified object type.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override <a href="#">bool</a> CanConvert(     <a href="#">Type</a> objectType )</pre>		
	<pre>Public Overrides Function CanConvert (      <a href="#">objectType</a> As <a href="#">Type</a>  ) As <a href="#">Boolean</a></pre>	
		<pre>public: virtual <a href="#">bool</a> CanConvert(     <a href="#">Type</a>^ objectType ) override</pre>

#### ▣ Parameters

##### ***objectType*** ([Type](#))

Type of the object.

#### ▣ Return Value

**true** if this instance can convert the specified object type; otherwise, **false**.

## ReadJson Method (reader, objectType, existingValue, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [StringEnumConverter](#)

### ► ReadJson(JsonReader, Type, Object, JsonSerializer)

Reads the JSON representation of the object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Object ReadJson(  
    JsonReader reader,  
    Type objectType,  
    Object existingValue,  
    JsonSerializer serializer  
)
```

```
Public Overrides Function ReadJson ( _  
    reader As JsonReader, _  
    objectType As Type, _  
    existingValue As Object, _  
    serializer As JsonSerializer _  
) As Object
```

```
public:  
virtual Object^ ReadJson(  
    JsonReader^ reader,  
    Type^ objectType,  
    Object^ existingValue,  
    JsonSerializer^ serializer  
) override
```

#### ▣ Parameters

**reader** ([JsonReader](#))

The [JsonReader](#) to read from.

***objectType*** ([Type](#))

Type of the object.

***existingValue*** ([Object](#))

The existing value of object being read.

***serializer*** ([JsonSerializer](#))

The calling serializer.

#### ▣ **Return Value**

The object value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [StringEnumConverter](#)

### ► WriteJson(JsonWriter, Object, JsonSerializer)

Writes the JSON representation of the object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteJson(  
    JsonWriter writer,  
    Object value,  
    JsonSerializer serializer  
)
```

```
Public Overrides Sub WriteJson ( _  
    writer As JsonWriter, _  
    value As Object, _  
    serializer As JsonSerializer _  
)
```

```
public:  
virtual void WriteJson(  
    JsonWriter^ writer,  
    Object^ value,  
    JsonSerializer^ serializer  
) override
```

#### ▣ Parameters

##### **writer** ([JsonWriter](#))

The [JsonWriter](#) to write to.

##### **value** ([Object](#))

The value.

***serializer* ([JsonSerializer](#))**

The calling serializer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# VersionConverter Class

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► **VersionConverter**

Converts a [Version](#) to and from a string (e.g. "1.2.3.4").

## Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class VersionConverter : JsonConverter
```

```
Public Class VersionConverter _  
    Inherits JsonConverter
```

```
public ref class VersionConverter : public JsonConve
```

## Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherite

Icon	Member	Description
	<a href="#">VersionConverter()</a>	Initializes a new instance of the <b>VersionConverter</b> class
	<a href="#">CanConvert(Type)</a>	Determines whether this instance can convert the specified object type.  (Overrides <a href="#">JsonConverter.CanConvert(Type)</a> .)
	<a href="#">CanRead</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can read JSON.

		(Inherited from <a href="#">JsonConverter.</a> )
	<a href="#">CanWrite</a>	Gets a value indicating whether this <a href="#">JsonConverter</a> can write JSON.  (Inherited from <a href="#">JsonConverter.</a> )
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object.</a> )
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object.</a> )
	<a href="#">GetSchema()</a>	Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a> .  (Inherited from <a href="#">JsonConverter.</a> )
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object.</a> )
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">ReadJson(JsonReader, Type, Object, JsonSerializer)</a>	Reads the JSON representation of the object.  (Overrides <a href="#">JsonConverter.ReadJson(JsonReader</a>

		<a href="#">Type, Object, JsonSerializer).</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of the object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonWriter, Object, JsonSerializer).</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonConverter](#)

└─ **VersionConverter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [VersionConverter](#) ►

### VersionConverter()

Initializes a new instance of the [VersionConverter](#) class

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public VersionConverter()
```

```
Public Sub New
```

```
public:  
VersionConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [VersionConverter](#) ►

### CanConvert(Type)

Determines whether this instance can convert the specified object type.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override <a href="#">bool</a> CanConvert(     <a href="#">Type</a> objectType )</pre>		
	<pre>Public Overrides Function CanConvert (      <a href="#">objectType</a> As <a href="#">Type</a>  ) As <a href="#">Boolean</a></pre>	
		<pre>public: virtual <a href="#">bool</a> CanConvert(     <a href="#">Type</a>^ objectType ) override</pre>

#### ▣ Parameters

##### ***objectType*** ([Type](#))

Type of the object.

#### ▣ Return Value

**true** if this instance can convert the specified object type; otherwise, **false**.

## ReadJson Method (reader, objectType, existingValue, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [VersionConverter](#) ►

**ReadJson(JsonReader, Type, Object, JsonSerializer)**

Reads the JSON representation of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Object ReadJson(  
    JsonReader reader,  
    Type objectType,  
    Object existingValue,  
    JsonSerializer serializer  
)
```

```
Public Overrides Function ReadJson ( _  
    reader As JsonReader, _  
    objectType As Type, _  
    existingValue As Object, _  
    serializer As JsonSerializer _  
) As Object
```

```
public:  
virtual Object^ ReadJson(  
    JsonReader^ reader,  
    Type^ objectType,  
    Object^ existingValue,  
    JsonSerializer^ serializer  
) override
```

### ▣ Parameters

**reader** ([JsonReader](#))

The [JsonReader](#) to read from.

***objectType*** ([Type](#))

Type of the object.

***existingValue*** ([Object](#))

The existing property value of the JSON that is being converted.

***serializer*** ([JsonSerializer](#))

The calling serializer.

#### ▣ **Return Value**

The object value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [VersionConverter](#) ►

### WriteJson(JsonWriter, Object, JsonSerializer)

Writes the JSON representation of the object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteJson(  
    JsonWriter writer,  
    Object value,  
    JsonSerializer serializer  
)
```

```
Public Overrides Sub WriteJson ( _  
    writer As JsonWriter, _  
    value As Object, _  
    serializer As JsonSerializer _  
)
```

```
public:  
virtual void WriteJson(  
    JsonWriter^ writer,  
    Object^ value,  
    JsonSerializer^ serializer  
) override
```

#### ▣ Parameters

##### **writer** ([JsonWriter](#))

The [JsonWriter](#) to write to.

##### **value** ([Object](#))

The value.

***serializer* ([JsonSerializer](#))**

The calling serializer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

Converts XML to and from JSON.

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class XmlNodeConverter : JsonConverter
```

```
Public Class XmlNodeConverter _
    Inherits JsonConverter
```

```
public ref class XmlNodeConverter : public JsonConve
```

▣ **Members**

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declared <input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">XmlNodeConverter()</a>	Initializes a new instance of the <b>XmlNodeConverter</b> class
	<a href="#">CanConvert(Type)</a>	Determines whether this instance convert the specified value type.  (Overrides <a href="#">JsonConverter.CanConvert(Type)</a> )
	<a href="#">CanRead</a>	Gets a value indicating whether the <a href="#">JsonConverter</a> can read JSON.

		(Inherited from <a href="#">JsonConverter.</a> )
	<a href="#">CanWrite</a>	Gets a value indicating whether the <a href="#">JsonConverter</a> can write JSON.  (Inherited from <a href="#">JsonConverter.</a> )
	<a href="#">DeserializeRootElementName</a>	Gets or sets the name of the root element to insert when deserializing XML if the JSON structure has produces multiple root elements.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object.</a> )
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object.</a> )
	<a href="#">GetSchema()</a>	Gets the <a href="#">JsonSchema</a> of the JSON produced by the <a href="#">JsonConverter</a> .  (Inherited from <a href="#">JsonConverter.</a> )
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object.</a> )
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .

		(Inherited from <a href="#">Object</a> .)
	<a href="#">OmitRootObject</a>	Gets or sets a value indicating whether to write the root JSON object.
	<a href="#">ReadJson(JsonReader, Type, Object, JsonSerializer)</a>	Reads the JSON representation of an object.  (Overrides <a href="#">JsonConverter.ReadJson(JsonReader, Type, Object, JsonSerializer)</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteArrayAttribute</a>	Gets or sets a flag to indicate whether to write the Json.NET array attribute. This attribute helps preserve array structure when converting the written XML to JSON.
	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of an object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonWriter, Object, JsonSerializer)</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonConverter](#)

└─ **XmlNodeConverter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [XmlNodeConverter](#) ►

### XmlNodeConverter()

Initializes a new instance of the [XmlNodeConverter](#) class

C#

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public XmlNodeConverter()
```

```
Public Sub New
```

```
public:  
XmlNodeConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (valueType)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [XmlNodeConverter](#) ►

### CanConvert(Type)

Determines whether this instance can convert the specified value type. C#

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override bool CanConvert(  
    Type valueType  
)
```

```
Public Overrides Function CanConvert ( _  
    valueType As Type _  
) As Boolean
```

```
public:  
virtual bool CanConvert(  
    Type^ valueType  
) override
```

#### ▣ Parameters

**valueType (Type)**

Type of the value.

#### ▣ Return Value

**true** if this instance can convert the specified value type; otherwise, **false**.

**DeserializeRootElementName**

Gets or sets the name of the root element to insert when deserializing to XML if the JSON structure has produces multiple root elements.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public string DeserializeRootElementName { get; set;
```

```
Public Property DeserializeRootElementName As String  
    Get  
    Set
```

```
public:  
property String^ DeserializeRootElementName {  
    String^ get ();  
    void set (String^ value);  
}
```

**Value**

The name of the deserialize root element.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [XmlNodeConverter](#) ►

## OmitRootObject

Gets or sets a value indicating whether to write the root JSON object.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool OmitRootObject { get; set; }
```

```
Public Property OmitRootObject As Boolean  
    Get  
    Set
```

```
public:  
property bool OmitRootObject {  
    bool get ();  
    void set (bool value);  
}
```

### Value

`true` if the JSON root object is omitted; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## ReadJson Method (reader, objectType, existingValue, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [XmlNodeConverter](#) ►

**ReadJson(JsonReader, Type, Object, JsonSerializer)**

C#

Reads the JSON representation of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Object ReadJson(  
    JsonReader reader,  
    Type objectType,  
    Object existingValue,  
    JsonSerializer serializer  
)
```

```
Public Overrides Function ReadJson ( _  
    reader As JsonReader, _  
    objectType As Type, _  
    existingValue As Object, _  
    serializer As JsonSerializer _  
) As Object
```

```
public:  
virtual Object^ ReadJson(  
    JsonReader^ reader,  
    Type^ objectType,  
    Object^ existingValue,  
    JsonSerializer^ serializer  
) override
```

### ▣ Parameters

**reader** ([JsonReader](#))

The [JsonReader](#) to read from.

***objectType*** ([Type](#))

Type of the object.

***existingValue*** ([Object](#))

The existing value of object being read.

***serializer*** ([JsonSerializer](#))

The calling serializer.

#### ▣ **Return Value**

The object value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [XmlNodeConverter](#) ►

### WriteArrayAttribute

Gets or sets a flag to indicate whether to write the Json.NET array attribute. This attribute helps preserve arrays when converting the written XML back to JSON.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool WriteArrayAttribute { get; set; }
```

```
Public Property WriteArrayAttribute As Boolean  
    Get  
    Set
```

```
public:  
property bool WriteArrayAttribute {  
    bool get ();  
    void set (bool value);  
}
```

#### Value

`true` if the array attribute is written to the XML; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [XmlNodeConverter](#) ►

### WriteJson(JsonWriter, Object, JsonSerializer)

Writes the JSON representation of the object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteJson(  
    JsonWriter writer,  
    Object value,  
    JsonSerializer serializer  
)
```

```
Public Overrides Sub WriteJson ( _  
    writer As JsonWriter, _  
    value As Object, _  
    serializer As JsonSerializer _  
)
```

```
public:  
virtual void WriteJson(  
    JsonWriter^ writer,  
    Object^ value,  
    JsonSerializer^ serializer  
) override
```

#### ▣ Parameters

##### **writer** ([JsonWriter](#))

The [JsonWriter](#) to write to.

##### **value** ([Object](#))

The value.

***serializer* ([JsonSerializer](#))**

The calling serializer.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Namespace

[Namespaces](#) ► **Newtonsoft.Json.Linq**

C#

The **Newtonsoft.Json.Linq** namespace provides classes that are used to implement LINQ to JSON.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<code>namespace Newtonsoft.Json.Linq</code>		

```
Namespace Newtonsoft.Json.Linq
```

```
namespace Newtonsoft.Json.Linq
```

### ▣ Types

All Types	Classes	Structures	Interfaces	Enumerations
-----------	---------	------------	------------	--------------

Icon	Type	Description
	<a href="#">Extensions</a>	Contains the LINQ to JSON extension methods.
	<a href="#">IEnumerable&lt;T&gt;</a>	Represents a collection of <a href="#">JToken</a> objects.
	<a href="#">JArray</a>	Represents a JSON array.
	<a href="#">JConstructor</a>	Represents a JSON constructor.

 <a href="#"><u>JContainer</u></a>	Represents a token that can contain other tokens.
 <a href="#"><u>JEnumerable&lt;T&gt;</u></a>	Represents a collection of <a href="#"><u>JToken</u></a> objects.
 <a href="#"><u>JObject</u></a>	Represents a JSON object.
 <a href="#"><u>JProperty</u></a>	Represents a JSON property.
 <a href="#"><u>JPropertyDescriptor</u></a>	Represents a view of a <a href="#"><u>JProperty</u></a> .
 <a href="#"><u>JRaw</u></a>	Represents a raw JSON string.
 <a href="#"><u>JToken</u></a>	Represents an abstract JSON token
 <a href="#"><u>JTokenEqualityComparer</u></a>	Compares tokens to determine whether they are equal.
 <a href="#"><u>JTokenReader</u></a>	Represents a reader that provides fast, non-cached, forward-only access to serialized Json data.
 <a href="#"><u>JTokenType</u></a>	Specifies the type of token.

	<b><u>JTokenWriter</u></b>	Represents a writer that provides a fast, non-cached, forward-only way of generating Json data.
	<b><u>JValue</u></b>	Represents a value in JSON (string, integer, date, etc).

---

Contains the LINQ to JSON extension methods.

## [-] Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public static class Extensions
```

```
<ExtensionAttribute> _  
Public NotInheritable Class Extensions
```

```
[ExtensionAttribute]  
public ref class Extensions abstract sealed
```

## [-] Members

All Members	Methods		
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">Ancestors&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns a collection of tokens that contains the ancestors of every token in the source collection.
	<a href="#">AsJEnumerable(IEnumerable&lt;JToken&gt;)</a>	Returns the input typed as <a href="#">IEnumerable&lt;T&gt;</a> .
	<a href="#">AsJEnumerable&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns the input typed as <a href="#">IEnumerable&lt;T&gt;</a> .

⇒ S	<a href="#">Children&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns a collection of child tokens of every array in the source collection.
⇒ S	<a href="#">Children&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Returns a collection of converted child tokens of every array in the source collection.
⇒ S	<a href="#">Descendants&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns a collection of tokens that contains the descendants of every token in the source collection.
⇒ S	<a href="#">Properties(IEnumerable&lt;JObject&gt;)</a>	Returns a collection of child properties of every object in the source collection.
⇒ S	<a href="#">Value&lt;U&gt;(IEnumerable&lt;JToken&gt;)</a>	Converts the value.
⇒ S	<a href="#">Value&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Converts the value.
⇒ S	<a href="#">Values(IEnumerable&lt;JToken&gt;, Object)</a>	Returns a collection of child values of every

		object in the source collection with the given key.
	<a href="#">Values(IEnumerable&lt;JToken&gt;)</a>	Returns a collection of child values of every object in the source collection.
	<a href="#">Values&lt;U&gt;(IEnumerable&lt;JToken&gt;, Object)</a>	Returns a collection of converted child values of every object in the source collection with the given key.
	<a href="#">Values&lt;U&gt;(IEnumerable&lt;JToken&gt;)</a>	Returns a collection of converted child values of every object in the source collection.

## ▣ Inheritance Hierarchy

[Object](#)

└─ **Extensions**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(source)**

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ► **Ancestors<T> (IEnumerable<T>)**

Returns a collection of tokens that contains the ancestors of every token in the source collection.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static IEnumerable<JToken> Ancestors<T>(
    this IEnumerable<T> source
)
where T : JToken
```

```
<ExtensionAttribute> _
Public Shared Function Ancestors(Of T As JToken) ( _
    source As IEnumerable(Of T) _
) As IEnumerable(Of JToken)
```

```
[ExtensionAttribute]
public:
generic<typename T>
where T : JToken
static IEnumerable<JToken>^ Ancestors(
    IEnumerable<T>^ source
)
```

### ▣ Generic Template Parameters

**T**

The type of the objects in source, constrained to [JToken](#).

### ▣ Parameters

**source** ([IEnumerable<T>](#))

An [IEnumerable<T>](#) of [JToken](#) that contains the source collection.

#### ▣ **Return Value**

An [IEnumerable<T>](#) of [JToken](#) that contains the ancestors of every node in the source collection.

#### ▣ **Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable<T>](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## AsJEnumerable()

### Members

Icon	Member	Description
	<a href="#">AsJEnumerable(IEnumerable&lt;JToken&gt;)</a>	Returns the input typed as <a href="#">IEnumerable&lt;T&gt;</a> .
	<a href="#">AsJEnumerable&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns the input typed as <a href="#">IEnumerable&lt;T&gt;</a> .

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(source)**[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ►**AsJEnumerable(IEnumerable<JToken>)**

C#

Returns the input typed as [IJEnumerable<T>](#).▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public static IJEnumerable<JToken> AsJEnumerable(  
    this IEnumerable<JToken> source  
)
```

```
<ExtensionAttribute> _  
Public Shared Function AsJEnumerable ( _  
    source As IEnumerable\(Of JToken\) _  
) As IJEnumerable\(Of JToken\)
```

```
[ExtensionAttribute]  
public:  
static IJEnumerable<JToken>^ AsJEnumerable(  
    IEnumerable<JToken>^ source  
)
```

▣ **Parameters****source (IEnumerable<JToken>)**An [IEnumerable<T>](#) of [JToken](#) that contains the source collection.▣ **Return Value**The input typed as [IJEnumerable<T>](#).▣ **Usage Note**In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable<JToken>](#). When you use instance

method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## Method (source)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ►

### AsJEnumerable<T>(IEnumerable<T>)

Returns the input typed as [IEnumerable<T>](#).

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static IEnumerable<T> AsJEnumerable<T>(
    this IEnumerable<T> source
)
where T : JToken
```

```
<ExtensionAttribute> _
Public Shared Function AsJEnumerable(Of T As JToken)
    source As IEnumerable(Of T) _
) As IEnumerable(Of T)
```

```
[ExtensionAttribute]
public:
generic<typename T>
where T : JToken
static IEnumerable<T>^ AsJEnumerable(
    IEnumerable<T>^ source
)
```

#### ▣ Generic Template Parameters

*T*

The source collection type.

#### ▣ Parameters

**source** ([IEnumerable](#)<T>)

An [IEnumerable](#)<T> of [JToken](#) that contains the source collection.

## ▣ Return Value

The input typed as [IJEnumerable<T>](#).

## ▣ Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable<T>](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



C#

[-] **Members**

Icon	Member	Description
	<a href="#">Children&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns a collection of child tokens of every array in the source collection.
	<a href="#">Children&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Returns a collection of converted child tokens of every array in the source collection.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(source)**

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ► **Children<T>**  
(IEnumerable<T>)

Returns a collection of child tokens of every array in the  source collection.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public static IEnumerable<JToken> Children<T>(
    this IEnumerable<T> source
)
where T : JToken
```

```
<ExtensionAttribute> _
Public Shared Function Children(Of T As JToken) ( _
    source As IEnumerable(Of T) _
) As IEnumerable(Of JToken)
```

```
[ExtensionAttribute]
public:
generic<typename T>
where T : JToken
static IEnumerable<JToken> Children(
    IEnumerable<T> source
)
```

**Generic Template Parameters****T**

The source collection type.

**Parameters****source** (IEnumerable<T>)

An [IEnumerable<T>](#) of [JToken](#) that contains the source collection.

#### ▣ **Return Value**

An [IEnumerable<T>](#) of [JToken](#) that contains the values of every node in the source collection.

#### ▣ **Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable<T>](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(source)**

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ► **Children<T, U>(IEnumerable<T>)**

C#

Returns a collection of converted child tokens of every array in the source collection.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public static IEnumerable<U> Children<T, U>(
    this IEnumerable<T> source
)
where T : JToken
```

```
<ExtensionAttribute> _
Public Shared Function Children(Of T As JToken, U) (
    source As IEnumerable(Of T) _
) As IEnumerable(Of U)
```

```
[ExtensionAttribute]
public:
generic<typename T, typename U>
where T : JToken
static IEnumerable<U>^ Children(
    IEnumerable<T>^ source
)
```

**Generic Template Parameters****T**

The source collection type.

**U**

The type to convert the values to.

## ▣ Parameters

**source** ([IEnumerable<T>](#))

An [IEnumerable<T>](#) of [JToken](#) that contains the source collection.

## ▣ Return Value

An [IEnumerable<T>](#) that contains the converted values of every node in the source collection.

## ▣ Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable<T>](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# Json.NET - Quick Starts & API Documentation **Descendants<T> Method** (source)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ►

## Descendants<T>(IEnumerable<T>)

Returns a collection of tokens that contains the descendants of every token in the source collection. C# ▼

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static IEnumerable<JToken> Descendants<T>(
    this IEnumerable<T> source
)
where T : JContainer
```

```
<ExtensionAttribute> _
Public Shared Function Descendants(Of T As JContainer
    source As IEnumerable\(Of T\) _
) As IEnumerable\(Of JToken\)
```

```
[ExtensionAttribute]
public:
generic<typename T>
where T : JContainer
static IEnumerable<JToken> Descendants(
    IEnumerable<T> source
)
```

### ▣ Generic Template Parameters

**T**

The type of the objects in source, constrained to [JContainer](#).

### ▣ Parameters

**source** ([IEnumerable<T>](#))

An [IEnumerable<T>](#) of [JToken](#) that contains the source collection.

#### ▣ **Return Value**

An [IEnumerable<T>](#) of [JToken](#) that contains the descendants of every node in the source collection.

#### ▣ **Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable<T>](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



**(source)**[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ►**Properties(IEnumerable<JObject>)**

Returns a collection of child properties of every object in the source collection.

C#

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public static IEnumerable<JProperty> Properties(  
    this IEnumerable<JObject> source  
)
```

```
<ExtensionAttribute> _  
Public Shared Function Properties ( _  
    source As IEnumerable(Of JObject) _  
) As IEnumerable(Of JProperty)
```

```
[ExtensionAttribute]  
public:  
static IEnumerable<JProperty>^ Properties(  
    IEnumerable<JObject>^ source  
)
```

▣ **Parameters****source** ([IEnumerable<JObject>](#))An [IEnumerable<T>](#) of [JObject](#) that contains the source collection.▣ **Return Value**An [IEnumerable<T>](#) of [JProperty](#) that contains the properties of every object in the source collection.▣ **Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable<JObject>](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[-] **Members**

Icon	Member	Description
	<a href="#">Value&lt;U&gt;(IEnumerable&lt;JToken&gt;)</a>	Converts the value.
	<a href="#">Value&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Converts the value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ► **Value<U>**  
(IEnumerable<JToken>)

Converts the value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static U Value<U>(
    this IEnumerable<JToken> value
)
```

```
<ExtensionAttribute> _
Public Shared Function Value(Of U) ( _
    value As IEnumerable(Of JToken) _
) As U
```

```
[ExtensionAttribute]
public:
generic<typename U>
static U Value(
    IEnumerable<JToken>^ value
)
```

### ▣ Generic Template Parameters

**U**

The type to convert the value to.

### ▣ Parameters

**value (IEnumerable<JToken>)**

A [JToken](#) cast as a [IEnumerable<T>](#) of [JToken](#).

### ▣ Return Value

A converted value.

## ▣ Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable<JToken>](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ► **Value<T, U> (IEnumerable<T>)**

Converts the value.

**▣ Declaration Syntax****C#****Visual Basic****Visual C++**

```
public static U Value<T, U>(
    this IEnumerable<T> value
)
where T : JToken
```

```
<ExtensionAttribute> _
Public Shared Function Value(Of T As JToken, U) ( _
    value As IEnumerable(Of T) _
) As U
```

```
[ExtensionAttribute]
public:
generic<typename T, typename U>
where T : JToken
static U Value(
    IEnumerable<T>^ value
)
```

**▣ Generic Template Parameters****T**

The source collection type.

**U**

The type to convert the value to.

**▣ Parameters**

**value** ([IEnumerable<T>](#))

A [JToken](#) cast as a [IEnumerable<T>](#) of [JToken](#).

▣ **Return Value**

A converted value.

▣ **Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable<T>](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Members

Icon	Member	Description
	<a href="#">Values(IEnumerable&lt;JToken&gt;, Object)</a>	Returns a collection of child values of every object in the source collection with the given key.
	<a href="#">Values(IEnumerable&lt;JToken&gt;)</a>	Returns a collection of child values of every object in the source collection.
	<a href="#">Values&lt;U&gt;(IEnumerable&lt;JToken&gt;, Object)</a>	Returns a collection of converted child values of every object in the source collection with the given key.
	<a href="#">Values&lt;U&gt;(IEnumerable&lt;JToken&gt;)</a>	Returns a collection of converted child values of every object in the source collection.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ►

## Values(IEnumerable<JToken>)

Returns a collection of child values of every object in the source collection.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public static IEnumerable<JToken> Values(  
    this IEnumerable<JToken> source  
)
```

```
<ExtensionAttribute> _  
Public Shared Function Values ( _  
    source As IEnumerable(Of JToken) _  
) As IEnumerable(Of JToken)
```

```
[ExtensionAttribute]  
public:  
static IEnumerable<JToken> Values(  
    IEnumerable<JToken> source  
)
```

### Parameters

#### **source** (IEnumerable<JToken>)

An [IEnumerable<T>](#) of [JToken](#) that contains the source collection.

### Return Value

An [IEnumerable<T>](#) of [JToken](#) that contains the values of every node in the source collection.

### Usage Note

In Visual Basic and C#, you can call this method as an instance method

on any object of type [IEnumerable<JToken>](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(source)**

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ► **Values<U>**  
(IEnumerable<JToken>)

Returns a collection of converted child values of every object in the source collection.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public static IEnumerable<U> Values<U>(
    this IEnumerable<JToken> source
)
```

```
<ExtensionAttribute> _
Public Shared Function Values(Of U) ( _
    source As IEnumerable(Of JToken) _
) As IEnumerable(Of U)
```

```
[ExtensionAttribute]
public:
generic<typename U>
static IEnumerable<U>^ Values(
    IEnumerable<JToken>^ source
)
```

**Generic Template Parameters****U**

The type to convert the values to.

**Parameters****source** (IEnumerable<JToken>)

An IEnumerable<T> of JToken that contains the source collection.

### ▣ **Return Value**

An [IEnumerable<T>](#) that contains the converted values of every node in the source collection.

### ▣ **Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable<JToken>](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(source, key)**

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ► **Values<U> (IEnumerable<JToken>, Object)**

Returns a collection of converted child values of every object in the source collection with the given key.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static IEnumerable<U> Values<U>(
    this IEnumerable<JToken> source,
    Object key
)
```

```
<ExtensionAttribute> _
Public Shared Function Values(Of U) ( _
    source As IEnumerable(Of JToken), _
    key As Object _
) As IEnumerable(Of U)
```

```
[ExtensionAttribute]
public:
generic<typename U>
static IEnumerable<U>^ Values(
    IEnumerable<JToken>^ source,
    Object^ key
)
```

### ▣ Generic Template Parameters

**U**

The type to convert the values to.

### ▣ Parameters

**source** ([IEnumerable<JToken>](#))

An [IEnumerable<T>](#) of [JToken](#) that contains the source collection.

**key** ([Object](#))

The token key.

#### ▣ **Return Value**

An [IEnumerable<T>](#) that contains the converted values of every node in the source collection with the given key.

#### ▣ **Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable<JToken>](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Values Method (source, key)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ►

### Values(IEnumerable<JToken>, Object)

Returns a collection of child values of every object in the source collection with the given key.

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static <a href="#">IEnumerable&lt;JToken&gt;</a> Values(     this <a href="#">IEnumerable&lt;JToken&gt;</a> source,     <a href="#">Object</a> key )</pre>		
	<pre>&lt;<a href="#">ExtensionAttribute</a>&gt; _ Public Shared Function Values ( _     source As <a href="#">IEnumerable</a>(Of <a href="#">JToken</a>), _     key As <a href="#">Object</a> _ ) As <a href="#">IEnumerable</a>(Of <a href="#">JToken</a>)</pre>	
		<pre>[<a href="#">ExtensionAttribute</a>] public: static <a href="#">IEnumerable&lt;JToken&gt;</a>^ Values(     <a href="#">IEnumerable&lt;JToken&gt;</a>^ source,     <a href="#">Object</a>^ key )</pre>

#### Parameters

##### source ([IEnumerable<JToken>](#))

An [IEnumerable<T>](#) of [JToken](#) that contains the source collection.

##### key ([Object](#))

The token key.

### ▣ **Return Value**

An [IEnumerable<T>](#) of [JToken](#) that contains the values of every node in the source collection with the given key.

### ▣ **Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable<JToken>](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Interface

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► **IEnumerable<T>**

Represents a collection of [JToken](#) objects.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public interface IEnumerable<out T> : IEnumerable<T>
    IEnumerable
where T : JToken
```

```
Public Interface IEnumerable(Of Out T As JToken) _
    Inherits IEnumerable(Of T), IEnumerable
```

```
generic<typename T>
where T : JToken
public interface class IEnumerable : IEnumerable<T>
    IEnumerable
```

### Generic Template Parameters

**T**

The type of token

### Members

All Members	Methods	Properties	
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited
Icon	Member	Description	
	<a href="#">Ancestors&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns a collection of tokens that contains the	

		ancestors of every token in the source collection.  (Inherited from <a href="#">Extensions.</a> )
	<a href="#">AsJEnumerable&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns the input typed a <b>IEnumerable&lt;T&gt;</b> .  (Inherited from <a href="#">Extensions.</a> )
	<a href="#">Children&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns a collection of child tokens of every array in the source collection.  (Inherited from <a href="#">Extensions.</a> )
	<a href="#">Children&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Returns a collection of converted child tokens of every array in the source collection.  (Inherited from <a href="#">Extensions.</a> )
	<a href="#">GetEnumerator()</a>	Returns an enumerator that iterates through the collection.  (Inherited from <a href="#">IEnumerable&lt;T&gt;.</a> )
	<a href="#">GetEnumerator()</a>	Returns an enumerator that iterates through a collection.  (Inherited from <a href="#">IEnumerable.</a> )



	<a href="#">Item[Object]</a>	Gets the <b>IEnumerable&lt;T&gt;</b> with the specified key.
	<a href="#">Value&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Converts the value.  (Inherited from <a href="#">Extensions.</a> )

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [IEnumerable<T>](#) ►

## Item[Object]

Gets the [IEnumerable<T>](#) with the specified key.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre><code><a href="#">IEnumerable&lt;JToken&gt;</a> this[     <a href="#">Object</a> key ] { get; }</code></pre>		
<pre><code>ReadOnly Default Property Item (      key As <a href="#">Object</a>   ) As <a href="#">IEnumerable</a>(Of <a href="#">JToken</a>)     Get</code></pre>		
<pre><code>property <a href="#">IEnumerable&lt;JToken&gt;</a>^ default[<a href="#">Object</a>^ key]     <a href="#">IEnumerable&lt;JToken&gt;</a>^ get (<a href="#">Object</a>^ key); }</code></pre>		

### ▣ Parameters

**key** ([Object](#))

### ▣ Value

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JArray Class

Represents a JSON array.

## Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public class JArray : JContainer, IList&lt;JToken&gt;,     ICollection&lt;JToken&gt;, IEnumerable&lt;JToken&gt;, IE</pre>		

```
Public Class JArray _
    Inherits JContainer _
    Implements IList(Of JToken), ICollection(Of
    IEnumerable(Of JToken), IEnumerable
```

```
public ref class JArray : public JContainer,
    IList<JToken^>, ICollection<JToken^>, IEnume
    IEnumerable
```

## Members

All Members	Constructors	Methods	Properties	Events
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance		<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static		<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JArray()</a>	Initializes the JArray class.
	<a href="#">JArray(JArray)</a>	Initializes the JArray class with an existing JArray object.

	<a href="#">JArray(Object[])</a>	Initia the J spec
	<a href="#">JArray(Object)</a>	Initia the J spec
	<a href="#">Add(JToken)</a>	Adds <a href="#">IColl</a>
	<a href="#">Add(Object)</a>	Adds as ch  (Inhe
	<a href="#">AddAfterSelf(Object)</a>	Adds imme  (Inhe
	<a href="#">AddBeforeSelf(Object)</a>	Adds imme token  (Inhe
	<a href="#">AddFirst(Object)</a>	Adds as th <a href="#">JTok</a>  (Inhe
	<a href="#">AddingNew</a>	Occu adde

		(Inhe
	<a href="#">AfterSelf()</a>	Retu sibir token
		(Inhe
	<a href="#">Ancestors()</a>	Retu ance token
		(Inhe
	<a href="#">Ancestors&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Retu token ance the s
		(Inhe
	<a href="#">AsJEnumerable(IEnumerable&lt;JToken&gt;)</a>	Retu <a href="#">JEn</a>
		(Inhe
	<a href="#">AsJEnumerable&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Retu <a href="#">JEn</a>
		(Inhe
	<a href="#">BeforeSelf()</a>	Retu sibir token
		(Inhe
	<a href="#">Children()</a>	Retu child docu
		(Inhe

	<a href="#">Children&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Retu token sour  (Inhe
	<a href="#">Children&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Retu conv ever colle  (Inhe
	<a href="#">ChildrenTokens</a>	Gets token  (Ove <a href="#">JCor</a>
	<a href="#">Clear()</a>	Rem <a href="#">IColl</a>
	<a href="#">CollectionChanged</a>	Occu of th chan reset  (Inhe
	<a href="#">Contains(JToken)</a>	Dete <a href="#">IColl</a> spec
	<a href="#">Count</a>	Gets JSOI  (Inhe
		

	<a href="#">CreateReader()</a>	Cre this t  (Inhe
	<a href="#">CreateWriter()</a>	Cre can l to th  (Inhe
	<a href="#">DeepClone()</a>	Cre the <a href="#">J</a> are r  (Inhe
	<a href="#">Descendants()</a>	Retu desc token  (Inhe
	<a href="#">Equals(Object)</a>	Dete spec the c  (Inhe
	<a href="#">Finalize()</a>	Allo to fre perf oper is re colle  (Inhe
	<a href="#">First</a>	Get t this t

		(Inhe
 	<a href="#">FromObject(Object)</a>	Cre obje
 	<a href="#">FromObject(Object, JsonSerializer)</a>	Cre obje
	<a href="#">GetHashCode()</a>	Serv for a  (Inhe
	<a href="#">GetMetaObject(Expression)</a>	Retu <a href="#">Dyna</a> respo oper this c  (Inhe
	<a href="#">GetType()</a>	Gets insta  (Inhe
	<a href="#">HasValues</a>	Gets whet child  (Inhe
	<a href="#">IndexOf(JToken)</a>	Dete spec
	<a href="#">Insert(Int32, JToken)</a>	Inser <a href="#">IList</a>

		index
	<a href="#">IsValid(JToken, JsonSchema)</a>	Dete <a href="#">JTok</a>  (Inhe
	<a href="#">IsValid(JToken, JsonSchema, IList&lt;String&gt;)</a>	Dete <a href="#">JTok</a>  (Inhe
	<a href="#">Item[Object]</a>	Gets spec  (Ove <a href="#">JTok</a>
	<a href="#">Item[Int32]</a>	Gets the s
	<a href="#">Last</a>	Get t this t  (Inhe
	<a href="#">ListChanged</a>	Occu chan list c  (Inhe
 	<a href="#">Load(JsonReader)</a>	Load <a href="#">Json</a>
	<a href="#">MemberwiseClone()</a>	Crea the c

		(Inhe
	<a href="#">Next</a>	Gets of thi  (Inhe
	<a href="#">OnAddingNew(AddingNewEventArgs)</a>	Rais even  (Inhe
	<a href="#">OnCollectionChanged(NotifyCollectionChangedEventArgs)</a>	Rais <a href="#">Colle</a>  (Inhe
	<a href="#">OnListChanged(ListChangedEventArgs)</a>	Rais even  (Inhe
	<a href="#">Parent</a>	Gets  (Inhe
	<a href="#">Parse(String)</a>	Loac that (
	<a href="#">Previous</a>	Gets token  (Inhe
	<a href="#">Remove(JToken)</a>	Rem occu obje <a href="#">IColl</a>

	<a href="#">Remove()</a>	Rem pare  (Inhe
	<a href="#">RemoveAll()</a>	Rem from  (Inhe
	<a href="#">RemoveAt(Int32)</a>	Rem at th
	<a href="#">Replace(JToken)</a>	Repl spec  (Inhe
	<a href="#">ReplaceAll(Object)</a>	Repl of thi spec  (Inhe
	<a href="#">Root</a>	Gets <a href="#">JTok</a>  (Inhe
	<a href="#">SelectToken(String)</a>	Sele matc  (Inhe
	<a href="#">SelectToken(String, Boolean)</a>	Sele matc  (Inhe
	<a href="#">ToObject&lt;T&gt;()</a>	Crea type

		(Inhe
	<a href="#">ToObject&lt;T&gt;(JsonSerializer)</a>	Cre type the s
		(Inhe
	<a href="#">ToString()</a>	Retu for th
		(Inhe
	<a href="#">ToString(Formatting, JsonSerializer[])</a>	Retu token form
		(Inhe
	<a href="#">Type</a>	Gets <a href="#">JTok</a>
		(Ove
	<a href="#">Validate(JToken, JsonSchema)</a>	Valid <a href="#">JTok</a>
		(Inhe
	<a href="#">Validate(JToken, JsonSchema, ValidationEventHandler)</a>	Valid <a href="#">JTok</a>
		(Inhe
	<a href="#">Value&lt;T&gt;(Object)</a>	Gets spec the s
		(Inhe
	<a href="#">Value&lt;U&gt;(IEnumerable&lt;JToken&gt;)</a>	Conv

		(Inhe
	<a href="#">Value&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Conv (Inhe
	<a href="#">Values&lt;T&gt;()</a>	Retu child docu (Inhe
	<a href="#">Values(IEnumerable&lt;JToken&gt;, Object)</a>	Retu value sour giver (Inhe
	<a href="#">Values(IEnumerable&lt;JToken&gt;)</a>	Retu value sour (Inhe
	<a href="#">Values&lt;U&gt;(IEnumerable&lt;JToken&gt;, Object)</a>	Retu conv ever colle (Inhe
	<a href="#">WriteTo(JsonWriter, JsonSerializer[])</a>	Write <a href="#">Json</a> (Ove <a href="#">JTok</a> <a href="#">Json</a>

## Inheritance Hierarchy

[Object](#)

└─ [JToken](#)

└─ [JContainer](#)

└─ **JArray**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Members

Icon	Member	Description
	<a href="#">JArray()</a>	Initializes a new instance of the <a href="#">JArray</a> class.
	<a href="#">JArray(JArray)</a>	Initializes a new instance of the <a href="#">JArray</a> class from another <a href="#">JArray</a> object.
	<a href="#">JArray(Object[])</a>	Initializes a new instance of the <a href="#">JArray</a> class with the specified content.
	<a href="#">JArray(Object)</a>	Initializes a new instance of the <a href="#">JArray</a> class with the specified content.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **JArray()**

C#

Initializes a new instance of the [JArray](#) class.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JArray()
```

```
Public Sub New
```

```
public:  
JArray()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JArray Constructor (other)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► [JArray\(JArray\)](#)

Initializes a new instance of the [JArray](#) class from another [JArray](#) object

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JArray(  
    JArray other  
)
```

```
Public Sub New ( _  
    other As JArray _  
)
```

```
public:  
JArray(  
    JArray^ other  
)
```

## Parameters

### ***other*** ([JArray](#))

A [JArray](#) object to copy from.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JArray Constructor (content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► [JArray\(Object\)](#)

Initializes a new instance of the [JArray](#) class with the specified content.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JArray(  
    Object content  
)
```

```
Public Sub New (   
    content As Object   
)
```

```
public:  
JArray(  
    Object^ content  
)
```

## Parameters

### **content (Object)**

The contents of the array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JArray Constructor (content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **JArray(Object[])**

Initializes a new instance of the [JArray](#) class with the specified content.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JArray(  
    params Object[] content  
)
```

```
Public Sub New ( _  
    ParamArray content As Object() _  
)
```

```
public:  
JArray(  
    ... array<Object>^ content  
)
```

## Parameters

### *content* ([Object](#)[])

The contents of the array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Add Method (item)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **Add(JToken)**

C#

Adds an item to the [ICollection<T>](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Add(  
    JToken item  
)
```

```
Public Sub Add ( _  
    item As JToken _  
)
```

```
public:  
virtual void Add(  
    JToken^ item  
) sealed
```

### Parameters

#### *item* (JToken)

The object to add to the [ICollection<T>](#).

### Exceptions

Exception	Condition
<a href="#">NotSupportedException</a>	The <a href="#">ICollection&lt;T&gt;</a> is read-only.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **ChildrenTokens**

C#

Gets the container's children tokens.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
protected override IList<JToken> ChildrenTokens { ge
```

```
Protected Overrides ReadOnly Property ChildrenTokens  
Get
```

```
protected:  
virtual property IList<JToken> ChildrenTokens {  
    IList<JToken> get () override;  
}
```

## Value

The container's children tokens.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **Clear()**

C# ▼

Removes all items from the [ICollection<T>](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Clear()
```

```
Public Sub Clear
```

```
public:  
virtual void Clear() sealed
```

## ▣ Exceptions

Exception	Condition
<a href="#">NotSupportedException</a>	The <a href="#">ICollection&lt;T&gt;</a> is read-only.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **Contains(JToken)**

C#

Determines whether the [ICollection<T>](#) contains a specific value.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Contains(  
    JToken item  
)
```

```
Public Function Contains ( _  
    item As JToken _  
) As Boolean
```

```
public:  
virtual bool Contains(  
    JToken^ item  
) sealed
```

## Parameters

### *item* (JToken)

The object to locate in the [ICollection<T>](#).

## Return Value

true if *item* is found in the [ICollection<T>](#); otherwise, false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

C#

## Members

Icon	Member	Description
	<a href="#">FromObject(Object)</a>	Creates a <a href="#">JArray</a> from an object.
	<a href="#">FromObject(Object, JsonSerializer)</a>	Creates a <a href="#">JArray</a> from an object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ►

## FromObject(Object)

Creates a [JArray](#) from an object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JArray FromObject(  
    Object o  
)
```

```
Public Shared Function FromObject ( _  
    o As Object _  
) As JArray
```

```
public:  
static JArray^ FromObject(  
    Object^ o  
)
```

### ▣ Parameters

#### o ([Object](#))

The object that will be used to create [JArray](#).

### ▣ Return Value

A [JArray](#) with the values of the specified object

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# FromObject Method (o, JsonSerializer)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ►

## FromObject(Object, JsonSerializer)

C#

Creates a [JArray](#) from an object.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JArray FromObject(  
    Object o,  
    JsonSerializer jsonSerializer  
)
```

```
Public Shared Function FromObject ( _  
    o As Object, _  
    jsonSerializer As JsonSerializer _  
) As JArray
```

```
public:  
static JArray^ FromObject(  
    Object^ o,  
    JsonSerializer^ jsonSerializer  
)
```

### Parameters

#### o ([Object](#))

The object that will be used to create [JArray](#).

#### jsonSerializer ([JsonSerializer](#))

The [JsonSerializer](#) that will be used to read the object.

### Return Value

A [JArray](#) with the values of the specified object

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

Determines the index of a specific item in the [IList<T>](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public int IndexOf(     JToken item )</pre>		
	<pre>Public Function IndexOf ( _     item As JToken _ ) As Integer</pre>	
		<pre>public: virtual int IndexOf(     JToken^ item ) sealed</pre>

## ▣ Parameters

### *item* ([JToken](#))

The object to locate in the [IList<T>](#).

## ▣ Return Value

The index of *item* if found in the list; otherwise, -1.

# Insert Method (index, item)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **Insert(Int32, JToken)**

Inserts an item to the [IList<T>](#) at the specified index.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Insert(  
    int index,  
    JToken item  
)
```

```
Public Sub Insert ( _  
    index As Integer, _  
    item As JToken _  
)
```

```
public:  
virtual void Insert(  
    int index,  
    JToken^ item  
) sealed
```

## Parameters

### *index* ([Int32](#))

The zero-based index at which *item* should be inserted.

### *item* ([JToken](#))

The object to insert into the [IList<T>](#).

## Exceptions

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

[ArgumentOutOfRangeException](#) index is not a valid index in the [IList<T>](#).

[NotSupportedException](#)

The [IList<T>](#) is read-only.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Members

Icon	Member	Description
	<a href="#">Item[Object]</a>	Gets the <a href="#">JToken</a> with the specified key  (Overrides <a href="#">JToken.Item[Object]</a> .)
	<a href="#">Item[Int32]</a>	Gets or sets the <a href="#">JToken</a> at the specified index.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **Item[Int32]**

C#

Gets or sets the [JToken](#) at the specified index.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JToken this[
    int index
] { get; set; }
```

```
Public Default Property Item ( _
    index As Integer _
) As JToken
    Get
    Set
```

```
public:
virtual property JToken^ default[int index] {
    JToken^ get (int index) sealed;
    void set (int index, JToken^ value) sealed;
}
```

## ▣ Parameters

*index* ([Int32](#))

## ▣ Value

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets the [JToken](#) with the specified key.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override JToken this[
    Object key
] { get; set; }
```

```
Public Overrides Default Property Item ( _
    key As Object _
) As JToken
    Get
    Set
```

```
public:
virtual property JToken^ default[Object^ key] {
    JToken^ get (Object^ key) override;
    void set (Object^ key, JToken^ value) overri
}
```

### ▣ Parameters

**key** ([Object](#))

### ▣ Value

The [JToken](#) with the specified key.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► [Load\(JsonReader\)](#)

C#

Loads an [JArray](#) from a [JsonReader](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JArray Load(  
    JsonReader reader  
)
```

```
Public Shared Function Load ( _  
    reader As JsonReader _  
) As JArray
```

```
public:  
static JArray^ Load(  
    JsonReader^ reader  
)
```

## ▣ Parameters

### *reader* ([JsonReader](#))

A [JsonReader](#) that will be read for the content of the [JArray](#).

## ▣ Return Value

A [JArray](#) that contains the JSON that was read from the specified [JsonReader](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **Parse(String)**

Load a [JArray](#) from a string that contains JSON.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JArray Parse(  
    string json  
)
```

```
Public Shared Function Parse ( _  
    json As String _  
) As JArray
```

```
public:  
static JArray^ Parse(  
    String^ json  
)
```

## ▣ Parameters

**json (String)**

A [String](#) that contains JSON.

## ▣ Return Value

A [JArray](#) populated from the string that contains JSON.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **Remove(JToken)**

Removes the first occurrence of a specific object from the [ICollection<T>](#)

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Remove(  
    JToken item  
)
```

```
Public Function Remove ( _  
    item As JToken _  
) As Boolean
```

```
public:  
virtual bool Remove(  
    JToken^ item  
) sealed
```

## Parameters

### *item* (JToken)

The object to remove from the [ICollection<T>](#).

## Return Value

true if *item* was successfully removed from the [ICollection<T>](#); otherwise false. This method also returns false if *item* is not found in the original [ICollection<T>](#).

## Exceptions

Exception	Condition
<a href="#">NotSupportedException</a>	The <a href="#">ICollection&lt;T&gt;</a> is read-only.

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## (index)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **RemoveAt(Int32)**

Removes the [IList<T>](#) item at the specified index.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void RemoveAt(  
    int index  
)
```

```
Public Sub RemoveAt ( _  
    index As Integer _  
)
```

```
public:  
virtual void RemoveAt(  
    int index  
) sealed
```

### ▣ Parameters

#### *index* ([Int32](#))

The zero-based index of the item to remove.

### ▣ Exceptions

Exception	Condition
<a href="#">ArgumentOutOfRangeException</a>	<i>index</i> is not a valid index in the <a href="#">IList&lt;T&gt;</a> .
<a href="#">NotSupportedException</a>	The <a href="#">IList&lt;T&gt;</a> is read-only.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **Type** C#

Gets the node type for this [JToken](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override JTokenType Type { get; }
```

```
Public Overrides ReadOnly Property Type As JTokenType  
Get
```

```
public:  
virtual property JTokenType Type {  
    JTokenType get () override;  
}
```

## ▣ Value

The type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## WriteTo Method (writer, converters)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ►

### WriteTo(JsonWriter, JsonSerializer[])

Writes this token to a [JsonWriter](#).

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteTo(  
    JsonWriter writer,  
    params JsonConverter[] converters  
)
```

```
Public Overrides Sub WriteTo ( _  
    writer As JsonWriter, _  
    ParamArray converters As JsonConverter() _  
)
```

```
public:  
virtual void WriteTo(  
    JsonWriter^ writer,  
    ... array<JsonConverter^>^ converters  
) override
```

#### ▣ Parameters

##### **writer** ([JsonWriter](#))

A [JsonWriter](#) into which this method will write.

##### **converters** ([JsonConverter](#)[])

A collection of [JsonConverter](#) which will be used when writing the token.

(4.5.6.14930)

# JConstructor Class

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► **JConstructor** C#

Represents a JSON constructor.

## Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class JConstructor : JContainer
```

```
Public Class JConstructor _
    Inherits JContainer
```

```
public ref class JConstructor : public JContainer
```

## Members

All Members	Constructors	Methods	Properties	Events
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance		<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static		<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JConstructor()</a>	Initializes the JConstructor.
	<a href="#">JConstructor(JConstructor)</a>	Initializes the JConstructor from an existing JConstructor object.
	<a href="#">JConstructor(String, Object[])</a>	Initializes the JConstructor with the specified string and object array.

		cont
	<a href="#">JConstructor(String, Object)</a>	Initia the J the s cont
	<a href="#">JConstructor(String)</a>	Initia the J the s
	<a href="#">Add(Object)</a>	Adds as ch  (Inhe
	<a href="#">AddAfterSelf(Object)</a>	Adds imme  (Inhe
	<a href="#">AddBeforeSelf(Object)</a>	Adds imme token  (Inhe
	<a href="#">AddFirst(Object)</a>	Adds as th <a href="#">JTok</a>  (Inhe
	<a href="#">AddingNew</a>	Occu adde

		(Inhe
	<a href="#">AfterSelf()</a>	Retu sibir token  (Inhe
	<a href="#">Ancestors()</a>	Retu ance token  (Inhe
	<a href="#">BeforeSelf()</a>	Retu sibir token  (Inhe
	<a href="#">Children()</a>	Retu child docu  (Inhe
	<a href="#">ChildrenTokens</a>	Gets token  (Ove <a href="#">JCor</a>
	<a href="#">CollectionChanged</a>	Occu of th chan rese  (Inhe
	<a href="#">Count</a>	Gets JSOI

		(Inhe
	<a href="#">CreateReader()</a>	Crea this t  (Inhe
	<a href="#">CreateWriter()</a>	Crea can l to the  (Inhe
	<a href="#">DeepClone()</a>	Crea the <a href="#">J</a> are r  (Inhe
	<a href="#">Descendants()</a>	Retu desc token  (Inhe
	<a href="#">Equals(Object)</a>	Dete spec the c  (Inhe
	<a href="#">Finalize()</a>	Allov to fre perf oper is rec colle  (Inhe
	<a href="#">First</a>	Get t this t

		(Inhe
	<a href="#">GetHashCode()</a>	Serv for a  (Inhe
	<a href="#">GetMetaObject(Expression)</a>	Retu <a href="#">Dyna</a> respo oper this c  (Inhe
	<a href="#">GetType()</a>	Gets insta  (Inhe
	<a href="#">HasValues</a>	Gets whet child  (Inhe
	<a href="#">IsValid(JToken, JsonSchema)</a>	Dete <a href="#">JTok</a>  (Inhe
	<a href="#">IsValid(JToken, JsonSchema, IList&lt;String&gt;)</a>	Dete <a href="#">JTok</a>  (Inhe
	<a href="#">Item[Object]</a>	Gets spec  (Ove <a href="#">JTok</a>

	<a href="#">Last</a>	Get t this t  (Inhe
	<a href="#">ListChanged</a>	Occu chan list c  (Inhe
	<a href="#">Load(JsonReader)</a>	Loac from
	<a href="#">MemberwiseClone()</a>	Crea the c  (Inhe
	<a href="#">Name</a>	Gets this c
	<a href="#">Next</a>	Gets of thi  (Inhe
	<a href="#">OnAddingNew(AddingNewEventArgs)</a>	Rais even  (Inhe
	<a href="#">OnCollectionChanged(NotifyCollectionChangedEventArgs)</a>	Rais <a href="#">Colle</a>  (Inhe
	<a href="#">OnListChanged(ListChangedEventArgs)</a>	Rais even

		(Inhe
	<a href="#">Parent</a>	Gets (Inhe
	<a href="#">Previous</a>	Gets token (Inhe
	<a href="#">Remove()</a>	Rem pare (Inhe
	<a href="#">RemoveAll()</a>	Rem from (Inhe
	<a href="#">Replace(JToken)</a>	Repl spec (Inhe
	<a href="#">ReplaceAll(Object)</a>	Repl of thi spec (Inhe
	<a href="#">Root</a>	Gets <a href="#">JTok</a> (Inhe
	<a href="#">SelectToken(String)</a>	Sele matc (Inhe
	<a href="#">SelectToken(String, Boolean)</a>	Sele

		matc (Inhe
	<a href="#">ToObject&lt;T&gt;()</a>	Cre type (Inhe
	<a href="#">ToObject&lt;T&gt;(JsonSerializer)</a>	Cre type the s (Inhe
	<a href="#">ToString()</a>	Retu for th (Inhe
	<a href="#">ToString(Formatting, JsonSerializer[])</a>	Retu token form (Inhe
	<a href="#">Type</a>	Gets <a href="#">JTok</a> (Ove
	<a href="#">Validate(JToken, JsonSchema)</a>	Valid <a href="#">JTok</a> (Inhe
	<a href="#">Validate(JToken, JsonSchema, ValidationEventHandler)</a>	Valid <a href="#">JTok</a> (Inhe
	<a href="#">Value&lt;T&gt;(Object)</a>	Gets spec

		the s
		(Inhe
	<a href="#">Values&lt;T&gt;()</a>	Retu child docu
		(Inhe
	<a href="#">WriteTo(JsonWriter, JsonConverter[])</a>	Write <a href="#">Json</a>
		(Ove <a href="#">JTok</a> <a href="#">Json</a>

## [-] Inheritance Hierarchy

[Object](#)

└─ [JToken](#)

└─ [JContainer](#)

└─ **JConstructor**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ►

### JConstructor()

#### Members

Icon	Member	Description
	<a href="#">JConstructor()</a>	Initializes a new instance of the <a href="#">JConstructor</a> class.
	<a href="#">JConstructor(JConstructor)</a>	Initializes a new instance of the <a href="#">JConstructor</a> class from another <a href="#">JConstructor</a> object.
	<a href="#">JConstructor(String, Object[])</a>	Initializes a new instance of the <a href="#">JConstructor</a> class with the specified name and content.
	<a href="#">JConstructor(String, Object)</a>	Initializes a new instance of the <a href="#">JConstructor</a> class with the specified name and content.
	<a href="#">JConstructor(String)</a>	Initializes a new instance of the <a href="#">JConstructor</a> class with the specified name.

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ►

### JConstructor()

Initializes a new instance of the [JConstructor](#) class.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JConstructor()
```

```
Public Sub New
```

```
public:  
JConstructor()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor (other)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ►

### JConstructor(JConstructor)

Initializes a new instance of the [JConstructor](#) class from another  [JConstructor](#) object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JConstructor(  
    JConstructor other  
)
```

```
Public Sub New ( _  
    other As JConstructor _  
)
```

```
public:  
JConstructor(  
    JConstructor^ other  
)
```

#### ▣ Parameters

##### ***other*** ([JConstructor](#))

A [JConstructor](#) object to copy from.

## Constructor (name)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ►

### JConstructor(String)

Initializes a new instance of the [JConstructor](#) class with the specified name. C# ▼

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JConstructor(  
    string name  
)
```

```
Public Sub New ( _  
    name As String _  
)
```

```
public:  
JConstructor(  
    String^ name  
)
```

#### ▣ Parameters

##### *name* ([String](#))

The constructor name.

## Constructor (name, content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ►

### JConstructor(String, Object)

Initializes a new instance of the [JConstructor](#) class with the specified name and content.

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JConstructor(     string name,     Object content )</pre>		
	<pre>Public Sub New ( _     name As String, _     content As Object _ )</pre>	
		<pre>public: JConstructor(     String^ name,     Object^ content )</pre>

#### Parameters

##### **name** ([String](#))

The constructor name.

##### **content** ([Object](#))

The contents of the constructor.

(4.5.6.14930)

# JConstructor

## Constructor (name, content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ►

### JConstructor(String, Object[])

Initializes a new instance of the [JConstructor](#) class with the specified name and content.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JConstructor(  
    string name,  
    params Object[] content  
)
```

```
Public Sub New ( _  
    name As String, _  
    ParamArray content As Object() _  
)
```

```
public:  
JConstructor(  
    String^ name,  
    ... array<Object^>^ content  
)
```

#### Parameters

##### **name (String)**

The constructor name.

##### **content (Object[])**

The contents of the constructor.

(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ►

## ChildrenTokens

C#

Gets the container's children tokens.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected override IList<JToken> ChildrenTokens { ge
```

```
Protected Overrides ReadOnly Property ChildrenTokens  
Get
```

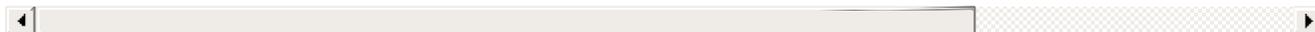
```
protected:  
virtual property IList<JToken> ChildrenTokens {  
    IList<JToken> get () override;  
}
```

### Value

The container's children tokens.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



Gets the [JToken](#) with the specified key.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override JToken this[
    Object key
] { get; set; }
```

```
Public Overrides Default Property Item ( _
    key As Object _
) As JToken
    Get
    Set
```

```
public:
virtual property JToken^ default[Object^ key] {
    JToken^ get (Object^ key) override;
    void set (Object^ key, JToken^ value) overri
}
```

## ▣ Parameters

**key** ([Object](#))

## ▣ Value

The [JToken](#) with the specified key.

## Load(JsonReader)

Loads an [JConstructor](#) from a [JsonReader](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public static JConstructor Load(
    JsonReader reader
)
```

```
Public Shared Function Load ( _
    reader As JsonReader _
) As JConstructor
```

```
public:
static JConstructor^ Load(
    JsonReader^ reader
)
```

### ▣ Parameters

#### ***reader*** ([JsonReader](#))

A [JsonReader](#) that will be read for the content of the [JConstructor](#).

### ▣ Return Value

A [JConstructor](#) that contains the JSON that was read from the specified [JsonReader](#).

Gets or sets the name of this constructor.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public string Name { get; set; }
```

```
Public Property Name As String  
    Get  
    Set
```

```
public:  
property String^ Name {  
    String^ get ();  
    void set (String^ value);  
}
```

## ▣ Value

The constructor name.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ► **Type**

C#

Gets the node type for this [JToken](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override JTokenType Type { get; }
```

```
Public Overrides ReadOnly Property Type As JTokenType  
Get
```

```
public:  
virtual property JTokenType Type {  
    JTokenType get () override;  
}
```

## ▣ Value

The type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## WriteTo Method (writer, converters)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ►

### WriteTo(JsonWriter, JsonConverter[])

Writes this token to a [JsonWriter](#).

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteTo(  
    JsonWriter writer,  
    params JsonConverter[] converters  
)
```

```
Public Overrides Sub WriteTo ( _  
    writer As JsonWriter, _  
    ParamArray converters As JsonConverter() _  
)
```

```
public:  
virtual void WriteTo(  
    JsonWriter^ writer,  
    ... array<JsonConverter^>^ converters  
) override
```

#### ▣ Parameters

##### **writer** ([JsonWriter](#))

A [JsonWriter](#) into which this method will write.

##### **converters** ([JsonConverter](#)[])

A collection of [JsonConverter](#) which will be used when writing the token.

(4.5.6.14930)

# JContainer Class

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► **JContainer**

Represents a token that can contain other tokens.

## Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public abstract class JContainer : <a href="#">JToken</a>,     <a href="#">IList&lt;JToken&gt;</a>, <a href="#">ICollection&lt;JToken&gt;</a>, <a href="#">IEnumera</a> <a href="#">ITypedList</a>, <a href="#">IBindingList</a>, <a href="#">IList</a>, <a href="#">ICollection</a> <a href="#">INotifyCollectionChanged</a></pre>		

```
Public MustInherit Class JContainer _
    Inherits JToken _
    Implements IList\(Of JToken\), ICollection\(Of
IEnumerable\(Of JToken\), ITypedList, IBinding
ICollection, IEnumerable, INotifyCollectionC
```

```
public ref class JContainer abstract : public JToken
    IList<JToken^>, ICollection<JToken^>, IEnume
ITypedList, IBindingList, IList, ICollection
INotifyCollectionChanged
```

## Members

All Members	Methods	Properties	Events
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declare <input checked="" type="checkbox"/> Inherited
Icon	Member	Description	
	<a href="#">Add(Object)</a>	Adds content to this container.	

	<a href="#">AddAfterSelf(Object)</a>	Adds content after  (Inherits from <a href="#">JTok</a> )
	<a href="#">AddBeforeSelf(Object)</a>	Adds content before  (Inherits from <a href="#">JTok</a> )
	<a href="#">AddFirst(Object)</a>	Adds content to the first child  <a href="#">JTok</a>
	<a href="#">AddingNew</a>	Occurs when a new item is added to the collection
	<a href="#">AfterSelf()</a>	Returns the content after the document  (Inherits from <a href="#">JTok</a> )
	<a href="#">Ancestors()</a>	Returns the content of the token

		(Inhe <a href="#">JTok</a>
	<a href="#">Ancestors&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Retu of tol cont: ance token colle  (Inhe <a href="#">Exte</a>
	<a href="#">AsJEnumerable(IEnumerable&lt;JToken&gt;)</a>	Retu type <a href="#">JEn</a>  (Inhe <a href="#">Exte</a>
	<a href="#">AsJEnumerable&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Retu type <a href="#">JEn</a>  (Inhe <a href="#">Exte</a>
	<a href="#">BeforeSelf()</a>	Retu of th befor docu  (Inhe <a href="#">JTok</a>
	<a href="#">Children()</a>	Retu of th this t docu  (Ove

		<a href="#">JTok</a>
	<a href="#">Children&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Retu of ch ever sour  (Inhe <a href="#">Exte</a>
	<a href="#">Children&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Retu of co token in the colle  (Inhe <a href="#">Exte</a>
	<a href="#">ChildrenTokens</a>	Gets child
	<a href="#">CollectionChanged</a>	Occu items colle chan colle
	<a href="#">Count</a>	Gets child
	<a href="#">CreateReader()</a>	Crea <a href="#">Json</a> token

		(Inhe <a href="#">JTok</a>
	<a href="#">CreateWriter()</a>	Crea <a href="#">Json</a> be u: token
	<a href="#">DeepClone()</a>	Crea insta <a href="#">JTok</a> token recu  (Inhe <a href="#">JTok</a>
	<a href="#">Descendants()</a>	Retu of th token in do
	<a href="#">Equals(Object)</a>	Dete the s is eq curre  (Inhe <a href="#">Obje</a>
	<a href="#">Finalize()</a>	Allov atten reso perf clear befor

		recla garb  (Inhe <a href="#">Obj</a>
	<a href="#">First</a>	Get t token  (Ove <a href="#">JTok</a>
	<a href="#">GetHashCode()</a>	Serv funct parti  (Inhe <a href="#">Obj</a>
	<a href="#">GetMetaObject(Expression)</a>	Retu <a href="#">Dyna</a> respi bindi perfc obje  (Inhe <a href="#">JTok</a>
	<a href="#">GetType()</a>	Gets curre  (Inhe <a href="#">Obj</a>
	<a href="#">HasValues</a>	Gets indic this t child  (Ove

		<a href="#">JTok</a>
	<a href="#">IsValid(JToken, JsonSchema)</a>	Dete the <a href="#">J</a>  (Inhe <a href="#">Exte</a>
	<a href="#">IsValid(JToken, JsonSchema, IList&lt;String&gt;)</a>	Dete the <a href="#">J</a>  (Inhe <a href="#">Exte</a>
	<a href="#">Item[Object]</a>	Gets the s  (Inhe <a href="#">JTok</a>
	<a href="#">Last</a>	Get t token  (Ove <a href="#">JTok</a>
	<a href="#">ListChanged</a>	Occu chan in the
	<a href="#">MemberwiseClone()</a>	Crea copy <a href="#">Obje</a>  (Inhe <a href="#">Obje</a>
	<a href="#">Next</a>	Gets token

		(Inhe <a href="#">JTok</a> )
	<a href="#">OnAddingNew(AddingNewEventArgs)</a>	Rais <a href="#">Addi</a>
	<a href="#">OnCollectionChanged(NotifyCollectionChangedEventArgs)</a>	Rais <a href="#">Colle</a> even
	<a href="#">OnListChanged(ListChangedEventArgs)</a>	Rais <a href="#">ListC</a>
	<a href="#">Parent</a>	Gets pare  (Inhe <a href="#">JTok</a> )
	<a href="#">Previous</a>	Gets sibir node  (Inhe <a href="#">JTok</a> )
	<a href="#">Remove()</a>	Rem from  (Inhe <a href="#">JTok</a> )
	<a href="#">RemoveAll()</a>	Rem node token

	<a href="#">Replace(JToken)</a>	Repl with token  (Inhe <a href="#">JTok</a> )
	<a href="#">ReplaceAll(Object)</a>	Repl child token spec
	<a href="#">Root</a>	Gets of thi  (Inhe <a href="#">JTok</a> )
	<a href="#">SelectToken(String)</a>	Sele that i objec  (Inhe <a href="#">JTok</a> )
	<a href="#">SelectToken(String, Boolean)</a>	Sele that i objec  (Inhe <a href="#">JTok</a> )
	<a href="#">ToObject&lt;T&gt;()</a>	Cre .NET <a href="#">JTok</a>  (Inhe

		<a href="#">JTok</a>
	<a href="#">ToObject&lt;T&gt;(JsonSerializer)</a>	<p>Cre .NET <a href="#">JTok</a> spec <a href="#">Json</a></p> <p>(Inhe <a href="#">JTok</a></p>
	<a href="#">ToString()</a>	<p>Retu JSOI</p> <p>(Inhe <a href="#">JTok</a></p>
	<a href="#">ToString(Formatting, JsonSerializer[])</a>	<p>Retu for th the g and</p> <p>(Inhe <a href="#">JTok</a></p>
	<a href="#">Type</a>	<p>Gets for th</p> <p>(Inhe <a href="#">JTok</a></p>
	<a href="#">Validate(JToken, JsonSchema)</a>	<p>Valid spec</p> <p>(Inhe <a href="#">Exte</a></p>
	<a href="#">Validate(JToken, JsonSchema, ValidationEventHandler)</a>	<p>Valid spec</p> <p>(Inhe</p>

		<a href="#">Exte</a>
	<a href="#">Value&lt;T&gt;(Object)</a>	Gets the s conv spec  (Inhe <a href="#">JTok</a>
	<a href="#">Value&lt;U&gt;(IEnumerable&lt;JToken&gt;)</a>	Conv  (Inhe <a href="#">Exte</a>
	<a href="#">Value&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Conv  (Inhe <a href="#">Exte</a>
	<a href="#">Values&lt;T&gt;()</a>	Retu of th this t docu  (Ove <a href="#">JTok</a>
	<a href="#">Values(IEnumerable&lt;JToken&gt;, Object)</a>	Retu of ch ever sour the g  (Inhe <a href="#">Exte</a>
	<a href="#">Values(IEnumerable&lt;JToken&gt;)</a>	Retu of ch ever sour

		(Inhe <a href="#">Exte</a>
	<a href="#">Values&lt;U&gt;(IEnumerable&lt;JToken&gt;, Object)</a>	Retu of co valu obje colle giver  (Inhe <a href="#">Exte</a>
	<a href="#">WriteTo(JsonWriter, JsonConverter[])</a>	Write <a href="#">Json</a>  (Inhe <a href="#">JTok</a>

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JToken](#)

└─ **JContainer**

└─ [JArray](#)

└─ [JConstructor](#)

└─ [JObject](#)

└─ [JProperty](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ► **Add(Object)**

Adds the specified content as children of this [JToken](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public virtual void Add(     <a href="#">Object</a> content )</pre>		
	<pre>Public Overridable Sub Add (      content As <a href="#">Object</a>  )</pre>	
		<pre>public: virtual void Add(     <a href="#">Object</a>^ content )</pre>

### ▣ Parameters

#### **content (Object)**

The content to be added.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(content)**[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ►**AddFirst(Object)**

Adds the specified content as the first children of this [JToken](#).

C#

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void AddFirst(  
    Object content  
)
```

```
Public Sub AddFirst ( _  
    content As Object _  
)
```

```
public:  
void AddFirst(  
    Object^ content  
)
```

**▣ Parameters*****content* ([Object](#))**

The content to be added.

Occurs before an item is added to the collection.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<code>public event <a href="#">AddingNewEventHandler</a> AddingNew</code>		
	<code>Public Event AddingNew As <a href="#">AddingNewEventHandler</a></code>	
		<pre>public:     event <a href="#">AddingNewEventHandler</a>^ AddingNew {         void add (<a href="#">AddingNewEventHandler</a>^ value);         void remove (<a href="#">AddingNewEventHandler</a>^ value);     }</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ► **Children()**

C#

Returns a collection of the child tokens of this token, in document order.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override IEnumerable<JToken> Children()
```

```
Public Overrides Function Children As IEnumerable(Of
```

```
public:  
virtual IEnumerable<JToken^> Children() override
```

### ▣ Return Value

An [IEnumerable<T>](#) of [JToken](#) containing the child tokens of this [JToken](#) in document order.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ►

## ChildrenTokens

Gets the container's children tokens.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected abstract IList<JToken> ChildrenTokens { ge
```

```
Protected MustOverride ReadOnly Property ChildrenTok  
Get
```

```
protected:  
virtual property IList<JToken> ChildrenTokens {  
    IList<JToken> get () abstract;  
}
```

### Value

The container's children tokens.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ►

## CollectionChanged

Occurs when the items list of the collection has changed, or the collection is reset.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public event NotifyCollectionChangedEventHandler Col
```

```
Public Event CollectionChanged As NotifyCollectionCh
```

```
public:  
virtual event NotifyCollectionChangedEventHandler ^  
    void add (NotifyCollectionChangedEventHandle  
    void remove (NotifyCollectionChangedEventHan  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ► **Count**

C# ▼

Gets the count of child JSON tokens.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public int Count { get; }
```

```
Public ReadOnly Property Count As Integer  
    Get
```

```
public:  
virtual property int Count {  
    int get () sealed;  
}
```

## ▣ Value

The count of child JSON tokens

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ► **CreateWriter()**

Creates an [JsonWriter](#) that can be used to add tokens to the [JToken](#).

C#

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonWriter CreateWriter()
```

```
Public Function CreateWriter As JsonWriter
```

```
public:  
JsonWriter^ CreateWriter()
```

## ▣ Return Value

An [JsonWriter](#) that is ready to have content written to it.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ► **Descendants()**

Returns a collection of the descendant tokens for this token in document order.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IEnumerable<JToken> Descendants()
```

```
Public Function Descendants As IEnumerable(Of JToken
```

```
public:  
IEnumerable<JToken> Descendants()
```

## ▣ Return Value

An [IEnumerable<T>](#) containing the descendant tokens of the [JToken](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ► **First**

C#

Get the first child token of this token.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override JToken First { get; }
```

```
Public Overrides ReadOnly Property First As JToken  
Get
```

```
public:  
virtual property JToken^ First {  
    JToken^ get () override;  
}
```

## ▣ Value

A [JToken](#) containing the first child token of the [JToken](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets a value indicating whether this token has children tokens.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override <a href="#">bool</a> HasValues { <a href="#">get</a>; }</pre>		
	<pre>Public Overrides ReadOnly Property HasValues As <a href="#">Bool</a>     Get</pre>	
		<pre>public: virtual property <a href="#">bool</a> HasValues {     <a href="#">bool</a> get () override; }</pre>

## ▣ Value

**true** if this token has child values; otherwise, **false**.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ► **Last**

C# ▼

Get the last child token of this token.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override JToken Last { get; }
```

```
Public Overrides ReadOnly Property Last As JToken  
Get
```

```
public:  
virtual property JToken^ Last {  
    JToken^ get () override;  
}
```

## ▣ Value

A [JToken](#) containing the last child token of the [JToken](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ► **ListChanged**

C#

Occurs when the list changes or an item in the list changes.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public event ListChangedEventHandler ListChanged
```

```
Public Event ListChanged As ListChangedEventHandler
```

```
public:  
virtual event ListChangedEventHandler^ ListChanged  
    void add (ListChangedEventHandler^ value);  
    void remove (ListChangedEventHandler^ value)  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ►

## OnAddingNew(AddingNewEventArgs)

C#

Raises the [AddingNew](#) event.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual void OnAddingNew(  
    AddingNewEventArgs e  
)
```

```
Protected Overridable Sub OnAddingNew ( _  
    e As AddingNewEventArgs _  
)
```

```
protected:  
virtual void OnAddingNew(  
    AddingNewEventArgs^ e  
)
```

### Parameters

**e** ([AddingNewEventArgs](#))

The [AddingNewEventArgs](#) instance containing the event data.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# OnCollectionChanged Method (e)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ►

## OnCollectionChanged(NotifyCollectionChangedEventArgs)

Raises the [CollectionChanged](#) event.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual void OnCollectionChanged(  
    NotifyCollectionChangedEventArgs e  
)
```

```
Protected Overridable Sub OnCollectionChanged ( _  
    e As NotifyCollectionChangedEventArgs _  
)
```

```
protected:  
virtual void OnCollectionChanged(  
    NotifyCollectionChangedEventArgs^ e  
)
```

### Parameters

#### e ([NotifyCollectionChangedEventArgs](#))

The [NotifyCollectionChangedEventArgs](#) instance containing the event data.

**(e)**[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ►

## OnListChanged(ListChangedEventArgs)

Raises the [ListChanged](#) event.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual void OnListChanged(  
    ListChangedEventArgs e  
)
```

```
Protected Overridable Sub OnListChanged ( _  
    e As ListChangedEventArgs _  
)
```

```
protected:  
virtual void OnListChanged(  
    ListChangedEventArgs^ e  
)
```

### ▣ Parameters

#### **e** ([ListChangedEventArgs](#))

The [ListChangedEventArgs](#) instance containing the event data.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ► **RemoveAll()**

C#

Removes the child nodes from this token.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void RemoveAll()
```

```
Public Sub RemoveAll
```

```
public:  
void RemoveAll()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ►

### ReplaceAll(Object)

Replaces the children nodes of this token with the specified content. C# ▼

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void ReplaceAll(  
    Object content  
)
```

```
Public Sub ReplaceAll ( _  
    content As Object _  
)
```

```
public:  
void ReplaceAll(  
    Object^ content  
)
```

#### ▣ Parameters

##### **content (Object)**

The content.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ► **Values<T>()**

C#

Returns a collection of the child values of this token, in document order.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override IEnumerable<T> Values<T>()
```

```
Public Overrides Function Values(Of T) As IEnumerable
```

```
public:  
generic<typename T>  
virtual IEnumerable<T>^ Values() override
```

### ▣ Generic Template Parameters

*T*

The type to convert the values to.

### ▣ Return Value

A [IEnumerable<T>](#) containing the child values of this [JToken](#), in document order.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Structure

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► **JEnumerable<T>**

Represents a collection of [JToken](#) objects.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public struct JEnumerable<T> : IEnumerable<T>,
    IEnumerable<T>, IEnumerable
where T : JToken
```

```
Public Structure JEnumerable(Of T As JToken) _
    Implements IEnumerable(Of T), IEnumerable(Of
    IEnumerable
```

```
generic<typename T>
where T : JToken
public value class JEnumerable : IEnumerable<T>,
    IEnumerable<T>, IEnumerable
```

### Generic Template Parameters

*T*

The type of token

### Members

All Members		Constructors	Methods	Properties	Fields	
<input checked="" type="checkbox"/>	Public		<input checked="" type="checkbox"/>	Instance	<input checked="" type="checkbox"/>	Declared
<input checked="" type="checkbox"/>	Protected		<input checked="" type="checkbox"/>	Static	<input checked="" type="checkbox"/>	Inherited
Icon	Member	Description				
	<a href="#">JEnumerable&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Initializes a new instance the <b>JEnumerable&lt;T&gt;</b>				

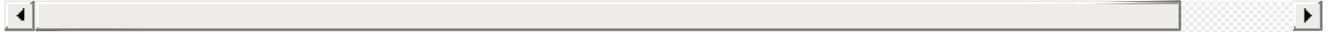
		struct.
	<a href="#">Ancestors&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns a collection of tokens that contains the ancestors of every token in the source collection.  (Inherited from <a href="#">Extension</a> )
	<a href="#">AsJEnumerable&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns the input typed as a <a href="#">JEnumerable&lt;T&gt;</a> .  (Inherited from <a href="#">Extension</a> )
	<a href="#">Children&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns a collection of child tokens of every array in the source collection.  (Inherited from <a href="#">Extension</a> )
	<a href="#">Children&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Returns a collection of converted child tokens of every array in the source collection.  (Inherited from <a href="#">Extension</a> )
	<a href="#">Empty</a>	An empty collection of <a href="#">JToken</a> objects.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to this instance.  (Overrides <a href="#">ValueType.Equals(Object)</a> )
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt

		to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetEnumerator()</a>	Returns an enumerator that iterates through the collection.
	<a href="#">GetHashCode()</a>	Returns a hash code for this instance.  (Overrides <a href="#">ValueType.GetHashCode</a> )
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">Item[Object]</a>	Gets the <a href="#">IEnumerable&lt;T&gt;</a> with the specified key.
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ToString()</a>	Returns the fully qualified type name of this instance.  (Inherited from <a href="#">ValueType</a> )
	<a href="#">Value&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Converts the value.

(Inherited from [Extension](#))

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



# JEnumerable<T> Constructor (enumerable)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JEnumerable<T>](#) ►

## JEnumerable<T>(IEnumerable<T>)

Initializes a new instance of the [JEnumerable<T>](#) struct.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JEnumerable(  
    IEnumerable<T> enumerable  
)
```

```
Public Sub New ( _  
    enumerable As IEnumerable(Of T) _  
)
```

```
public:  
JEnumerable(  
    IEnumerable<T>^ enumerable  
)
```

### ▣ Parameters

#### ***enumerable*** ([IEnumerable<T>](#))

The enumerable.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JEnumerable<T>](#) ► **Empty**

C#

An empty collection of [JToken](#) objects.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static readonly JEnumerable<T> Empty
```

```
Public Shared ReadOnly Empty As JEnumerable(Of T)
```

```
public:  
static initonly JEnumerable<T> Empty
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JEnumerable<T>](#) ►

## Equals(Object)

Determines whether the specified [Object](#) is equal to this instance.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override <a href="#">bool</a> Equals(     <a href="#">Object</a> <i>obj</i> )</pre>		
	<pre>Public Overrides Function Equals (      <i>obj</i> As <a href="#">Object</a>  ) As <a href="#">Boolean</a></pre>	
		<pre>public: virtual <a href="#">bool</a> Equals(     <a href="#">Object</a>^ <i>obj</i> ) override</pre>

### ▣ Parameters

#### *obj* ([Object](#))

The [Object](#) to compare with this instance.

### ▣ Return Value

`true` if the specified [Object](#) is equal to this instance; otherwise, `false`

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JEnumerable<T>](#) ►

## GetEnumerator()

C#

Returns an enumerator that iterates through the collection.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IEnumerator<T> GetEnumerator()
```

```
Public Function GetEnumerator As IEnumerator(Of T)
```

```
public:  
virtual IEnumerator<T>^ GetEnumerator() sealed
```

### ▣ Return Value

A [IEnumerator<T>](#) that can be used to iterate through the collection.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JEnumerable<T>](#) ►

## GetHashCode()

Returns a hash code for this instance.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

```
public:  
virtual int GetHashCode() override
```

### ▣ Return Value

A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Item[Object]

Gets the [JEnumerable<T>](#) with the specified key.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public IEnumerable<JToken> this[
    Object key
] { get; }
```

```
Public ReadOnly Default Property Item ( _
    key As Object _
) As IEnumerable\(Of JToken\)
    Get
```

```
public:
virtual property IEnumerable<JToken> default[Object
    IEnumerable<JToken> get (Object key) sea
}
```

### ▣ Parameters

**key** ([Object](#))

### ▣ Value

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



Represents a JSON object.

## Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public class JObject : JContainer, IDictionary&lt;string, JToken&gt;, ICollection&lt;KeyValuePair&lt;string, JToken&gt;&gt;, IEnumerable, INotifyPropertyChanged, ICustom</pre>		

```
Public Class JObject _
    Inherits JContainer _
    Implements IDictionary(Of String, JToken),
    ICollection(Of KeyValuePair(Of String, JToken)),
    IEnumerable, INotifyPropertyChanged, ICustom
```

```
public ref class JObject : public JContainer,
    IDictionary<String^, JToken^>, ICollection<K
    IEnumerable<KeyValuePair<String^, JToken^>>,
    INotifyPropertyChanged, ICustomTypeDescripto
```

## Members

All Members	Constructors	Methods	Properties	Events
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance		<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static		<input checked="" type="checkbox"/> Inherited
Icon	Member	Description		
	<a href="#">JObject()</a>	Initializes a new instance of the JObject class.		
	<a href="#">JObject(JObject)</a>	Initializes a new instance of the JObject class, copying the content of the specified JObject instance.		

		objec
	<a href="#">JObject(Object[])</a>	Initia <b>JObj</b> cont
	<a href="#">JObject(Object)</a>	Initia <b>JObj</b> cont
	<a href="#">Add(String, JToken)</a>	Adds
	<a href="#">Add(Object)</a>	Adds child  (Inhe
	<a href="#">AddAfterSelf(Object)</a>	Adds imme  (Inhe
	<a href="#">AddBeforeSelf(Object)</a>	Adds imme  (Inhe
	<a href="#">AddFirst(Object)</a>	Adds first c  (Inhe
	<a href="#">AddingNew</a>	Occu the c

		(Inhe
	<a href="#">AfterSelf()</a>	Retu token orde  (Inhe
	<a href="#">Ancestors()</a>	Retu token  (Inhe
	<a href="#">BeforeSelf()</a>	Retu token docu  (Inhe
	<a href="#">Children()</a>	Retu token orde  (Inhe
	<a href="#">ChildrenTokens</a>	Gets  (Ove <a href="#">JCor</a>
	<a href="#">CollectionChanged</a>	Occu colle colle  (Inhe
	<a href="#">Count</a>	Gets  (Inhe
	<a href="#">CreateReader()</a>	Crea token

		(Inhe
	<a href="#">CreateWriter()</a>	Crea used  (Inhe
	<a href="#">DeepClone()</a>	Crea <a href="#">JTok</a> recu  (Inhe
	<a href="#">Descendants()</a>	Retu desc docu  (Inhe
	<a href="#">Equals(Object)</a>	Dete <a href="#">Obje</a>  (Inhe
	<a href="#">Finalize()</a>	Allov resol oper recla  (Inhe
	<a href="#">First</a>	Get t  (Inhe
	<a href="#">FromObject(Object)</a>	Crea
	<a href="#">FromObject(Object, JsonSerializer)</a>	Crea
		

	<a href="#">GetEnumerator()</a>	Retu throu
	<a href="#">GetHashCode()</a>	Serv partic  (Inhe
	<a href="#">GetMetaObject(Expression)</a>	Retu respo perfo  (Ove <a href="#">JTok</a>
	<a href="#">GetType()</a>	Gets insta  (Inhe
	<a href="#">HasValues</a>	Gets token  (Inhe
	<a href="#">IsValid(JToken, JsonSchema)</a>	Dete valid  (Inhe
	<a href="#">IsValid(JToken, JsonSchema, IList&lt;String&gt;)</a>	Dete valid  (Inhe
	<a href="#">Item[Object]</a>	Gets key.  (Ove
	<a href="#">Item[String]</a>	Gets

		spec
	<a href="#">Last</a>	Get t (Inhe
	<a href="#">ListChanged</a>	Occu item (Inhe
 	<a href="#">Load(JsonReader)</a>	Loac <a href="#">Json</a>
 	<a href="#">MemberwiseClone()</a>	Crea curre (Inhe
	<a href="#">Next</a>	Gets node (Inhe
 	<a href="#">OnAddingNew(AddingNewEventArgs)</a>	Rais (Inhe
 	<a href="#">OnCollectionChanged(NotifyCollectionChangedEventArgs)</a>	Rais even (Inhe
 	<a href="#">OnListChanged(ListChangedEventArgs)</a>	Rais (Inhe
 	<a href="#">OnPropertyChanged(String)</a>	Rais with

	<a href="#">OnPropertyChanging(String)</a>	Rais with
	<a href="#">Parent</a>	Gets  (Inhe
	<a href="#">Parse(String)</a>	Load cont
	<a href="#">Previous</a>	Gets this r  (Inhe
	<a href="#">Properties()</a>	Gets objec
	<a href="#">Property(String)</a>	Gets nam
	<a href="#">PropertyChanged</a>	Occu chan
	<a href="#">PropertyChanging</a>	Occu chan
	<a href="#">PropertyValues()</a>	Gets objec

	<a href="#">Remove(String)</a>	Rem spec
	<a href="#">Remove()</a>	Rem  (Inhe
	<a href="#">RemoveAll()</a>	Rem token  (Inhe
	<a href="#">Replace(JToken)</a>	Repl spec  (Inhe
	<a href="#">ReplaceAll(Object)</a>	Repl token  (Inhe
	<a href="#">Root</a>	Gets  (Inhe
	<a href="#">SelectToken(String)</a>	Sele obje  (Inhe
	<a href="#">SelectToken(String, Boolean)</a>	Sele obje  (Inhe
	<a href="#">ToObject&lt;T&gt;()</a>	Crea the <a href="#">J</a>

		(Inhe
	<a href="#">ToObject&lt;T&gt;(JsonSerializer)</a>	Cre the J Json
		(Inhe
	<a href="#">ToString()</a>	Retu token
		(Inhe
	<a href="#">ToString(Formatting, JsonSerializer[])</a>	Retu using conv
		(Inhe
	<a href="#">TryGetValue(String, JToken)</a>	Tries
	<a href="#">Type</a>	Gets
		(Ove
	<a href="#">Validate(JToken, JsonSchema)</a>	Valid
		(Inhe
	<a href="#">Validate(JToken, JsonSchema, ValidationEventHandler)</a>	Valid
		(Inhe
	<a href="#">Value&lt;T&gt;(Object)</a>	Gets key c
		(Inhe
	<a href="#">Values&lt;T&gt;()</a>	Retu value orde

		(Inhe
	<a href="#">WriteTo(JsonWriter, JsonSerializer[])</a>	Write  (Ove <a href="#">JTok</a> <a href="#">Json</a>

## [-] Inheritance Hierarchy

[Object](#)

└─ [JToken](#)

└─ [JContainer](#)

└─ **JObject**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Members

Icon	Member	Description
	<a href="#">JObject()</a>	Initializes a new instance of the <a href="#">JObject</a> class.
	<a href="#">JObject(JObject)</a>	Initializes a new instance of the <a href="#">JObject</a> class from another <a href="#">JObject</a> object.
	<a href="#">JObject(Object[])</a>	Initializes a new instance of the <a href="#">JObject</a> class with the specified content.
	<a href="#">JObject(Object)</a>	Initializes a new instance of the <a href="#">JObject</a> class with the specified content.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **JObject()**

C#

Initializes a new instance of the [JObject](#) class.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JObject()
```

```
Public Sub New
```

```
public:  
JObject()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JObject Constructor (other)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **JObject(JObject)**

Initializes a new instance of the [JObject](#) class from another [JObject](#) object.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JObject(     <a href="#">JObject</a> other )</pre>	<pre>Public Sub New (      other As <a href="#">JObject</a>  )</pre>	<pre>public: JObject(     <a href="#">JObject</a>^ other )</pre>

## ▣ Parameters

### ***other*** ([JObject](#))

A [JObject](#) object to copy from.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JObject Constructor (content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **JObject(Object)**

Initializes a new instance of the [JObject](#) class with the specified content

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JObject(  
    Object content  
)
```

```
Public Sub New ( _  
    content As Object _  
)
```

```
public:  
JObject(  
    Object^ content  
)
```

## Parameters

### **content (Object)**

The contents of the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JObject Constructor (content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **JObject(Object[])**

Initializes a new instance of the [JObject](#) class with the specified content

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JObject(  
    params Object[] content  
)
```

```
Public Sub New ( _  
    ParamArray content As Object() _  
)
```

```
public:  
JObject(  
    ... array<Object>^ content  
)
```

## Parameters

### *content* ([Object](#)[])

The contents of the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Add Method (propertyName, value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **Add(String, JToken)**

Adds the specified property name.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Add(  
    string propertyName,  
    JToken value  
)
```

```
Public Sub Add ( _  
    propertyName As String, _  
    value As JToken _  
)
```

```
public:  
virtual void Add(  
    String^ propertyName,  
    JToken^ value  
) sealed
```

### ▣ Parameters

**propertyName** ([String](#))

Name of the property.

**value** ([JToken](#))

The value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **ChildrenTokens**

C#

Gets the container's children tokens.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
protected override IList<JToken> ChildrenTokens { ge
```

```
Protected Overrides ReadOnly Property ChildrenTokens  
Get
```

```
protected:  
virtual property IList<JToken> ChildrenTokens {  
    IList<JToken> get () override;  
}
```

## Value

The container's children tokens.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **FromObject()**

C# ▼

## Members

Icon	Member	Description
	<a href="#">FromObject(Object)</a>	Creates a <a href="#">JObject</a> from an object.
	<a href="#">FromObject(Object, JsonSerializer)</a>	Creates a <a href="#">JArray</a> from an object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ►

## FromObject(Object)

Creates a [JObject](#) from an object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JObject FromObject(  
    Object o  
)
```

```
Public Shared Function FromObject ( _  
    o As Object _  
) As JObject
```

```
public:  
static JObject^ FromObject(  
    Object^ o  
)
```

### ▣ Parameters

#### o ([Object](#))

The object that will be used to create [JObject](#).

### ▣ Return Value

A [JObject](#) with the values of the specified object

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# FromObject Method (o, jsonSerializer)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ►

## FromObject(Object, JsonSerializer)

C#

Creates a [JArray](#) from an object.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JObject FromObject(  
    Object o,  
    JsonSerializer jsonSerializer  
)
```

```
Public Shared Function FromObject ( _  
    o As Object, _  
    jsonSerializer As JsonSerializer _  
) As JObject
```

```
public:  
static JObject^ FromObject(  
    Object^ o,  
    JsonSerializer^ jsonSerializer  
)
```

### Parameters

#### o ([Object](#))

The object that will be used to create [JArray](#).

#### jsonSerializer ([JsonSerializer](#))

The [JsonSerializer](#) that will be used to read the object.

### Return Value

A [JArray](#) with the values of the specified object

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **GetEnumerator()**

C#

Returns an enumerator that iterates through the collection.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public IEnumerator<KeyValuePair<string, JToken>> Get
```

```
Public Function GetEnumerator As IEnumerator(Of KeyV
```

```
public:  
virtual IEnumerator<KeyValuePair<String^, JToken^>>^
```

## Return Value

A [IEnumerator<T>](#) that can be used to iterate through the collection.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## (parameter)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ►

### GetMetaObject(Expression)

Returns the [DynamicMetaObject](#) responsible for binding operations performed on this object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected override DynamicMetaObject GetMetaObject(  
    Expression parameter  
)
```

```
Protected Overrides Function GetMetaObject ( _  
    parameter As Expression _  
) As DynamicMetaObject
```

```
protected:  
virtual DynamicMetaObject^ GetMetaObject(  
    Expression^ parameter  
) override
```

#### ▣ Parameters

##### *parameter* ([Expression](#))

The expression tree representation of the runtime value.

#### ▣ Return Value

The [DynamicMetaObject](#) to bind this object.

[-] **Members**

Icon	Member	Description
	<a href="#">Item[Object]</a>	Gets the <a href="#">JToken</a> with the specified key  (Overrides <a href="#">JToken.Item[Object]</a> .)
	<a href="#">Item[String]</a>	Gets or sets the <a href="#">JToken</a> with the specified property name.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets the [JToken](#) with the specified key.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override JToken this[
    Object key
] { get; set; }
```

```
Public Overrides Default Property Item ( _
    key As Object _
) As JToken
    Get
    Set
```

```
public:
virtual property JToken^ default[Object^ key] {
    JToken^ get (Object^ key) override;
    void set (Object^ key, JToken^ value) overri
}
```

### ▣ Parameters

**key** ([Object](#))

### ▣ Value

The [JToken](#) with the specified key.

## (propertyName)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► [Item\[String\]](#)

C#

Gets or sets the [JToken](#) with the specified property name.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JToken this[
    string propertyName
] { get; set; }
```

```
Public Default Property Item ( _
    propertyName As String _
) As JToken
    Get
    Set
```

```
public:
virtual property JToken^ default[String^ propertyName]
    JToken^ get (String^ propertyName) sealed;
    void set (String^ propertyName, JToken^ value);
}
```

### ▣ Parameters

*propertyName* ([String](#))

### ▣ Value

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ►

## Load(JsonReader)

Loads an [JObject](#) from a [JsonReader](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JObject Load(  
    JsonReader reader  
)
```

```
Public Shared Function Load ( _  
    reader As JsonReader _  
) As JObject
```

```
public:  
static JObject^ Load(  
    JsonReader^ reader  
)
```

### ▣ Parameters

#### **reader** ([JsonReader](#))

A [JsonReader](#) that will be read for the content of the [JObject](#).

### ▣ Return Value

A [JObject](#) that contains the JSON that was read from the specified [JsonReader](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# OnPropertyChanged Method (propertyName)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ►

## OnPropertyChanged(String)

Raises the [PropertyChanged](#) event with the provided arguments.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual void OnPropertyChanged(  
    string propertyName  
)
```

```
Protected Overridable Sub OnPropertyChanged ( _  
    propertyName As String _  
)
```

```
protected:  
virtual void OnPropertyChanged(  
    String^ propertyName  
)
```

### Parameters

**propertyName** ([String](#))

Name of the property.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# OnPropertyChanging Method (propertyName)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ►

## OnPropertyChanging(String)

Raises the [PropertyChanging](#) event with the provided arguments.

### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>protected virtual void OnPropertyChanging(     string propertyName )</pre>	<pre>Protected Overridable Sub OnPropertyChanging ( _     propertyName As String _ )</pre>	<pre>protected: virtual void OnPropertyChanging(     String^ propertyName )</pre>

### Parameters

**propertyName** ([String](#))

Name of the property.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **Parse(String)**

C#

Load a [JObject](#) from a string that contains JSON.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JObject Parse(  
    string json  
)
```

```
Public Shared Function Parse ( _  
    json As String _  
) As JObject
```

```
public:  
static JObject^ Parse(  
    String^ json  
)
```

## ▣ Parameters

**json (String)**

A [String](#) that contains JSON.

## ▣ Return Value

A [JObject](#) populated from the string that contains JSON.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **Properties()**

C# ▼

Gets an [IEnumerable<T>](#) of this object's properties.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IEnumerable<JProperty> Properties()
```

```
Public Function Properties As IEnumerable(Of JProper
```

```
public:  
IEnumerable<JProperty^>^ Properties()
```

## ▣ Return Value

An [IEnumerable<T>](#) of this object's properties.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **Property(String)**

C# ▼

Gets a [JProperty](#) the specified name.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JProperty Property(  
    string name  
)
```

```
Public Function Property ( _  
    name As String _  
) As JProperty
```

```
public:  
JProperty^ Property(  
    String^ name  
)
```

## ▣ Parameters

***name*** ([String](#))

The property name.

## ▣ Return Value

A [JProperty](#) with the specified name or null.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **PropertyChanged**

C# ▼

Occurs when a property value changes.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public event PropertyChangedEventHandler PropertyChanged
```

```
Public Event PropertyChanged As PropertyChangedEvent
```

```
public:  
virtual event PropertyChangedEventHandler^ Property  
void add (PropertyChangedEventHandler^ value  
void remove (PropertyChangedEventHandler^ va  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ►

## PropertyChanging

C#

Occurs when a property value is changing.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public event PropertyChangingEventHandler PropertyCh
```

```
Public Event PropertyChanging As PropertyChangingEve
```

```
public:  
virtual event PropertyChangingEventHandler^ Propert  
    void add (PropertyChangingEventHandler^ valu  
    void remove (PropertyChangingEventHandler^ v  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **PropertyValues()**

Gets an [IEnumerable<T>](#) of this object's property values.

C# ▼

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IEnumerable<JToken> PropertyValues()
```

```
Public Function PropertyValues As IEnumerable(Of JTo
```

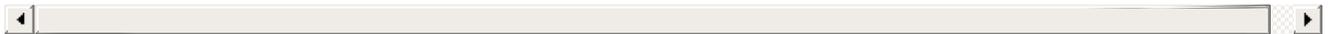
```
public:  
IEnumerable<JToken^> PropertyValues()
```

### ▣ Return Value

An [IEnumerable<T>](#) of this object's property values.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



# Remove Method (propertyName)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **Remove(String)**

C#

Removes the property with the specified name.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Remove(  
    string propertyName  
)
```

```
Public Function Remove ( _  
    propertyName As String _  
) As Boolean
```

```
public:  
virtual bool Remove(  
    String^ propertyName  
) sealed
```

## ▣ Parameters

***propertyName*** ([String](#))

Name of the property.

## ▣ Return Value

true if item was successfully removed; otherwise, false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## (propertyName, value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ►

### TryGetValue(String, JToken)

Tries the get value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool TryGetValue(  
    string propertyName,  
    out JToken value  
)
```

```
Public Function TryGetValue ( _  
    propertyName As String, _  
    <OutAttribute> ByRef value As JToken _  
) As Boolean
```

```
public:  
virtual bool TryGetValue(  
    String^ propertyName,  
    [OutAttribute] JToken^% value  
) sealed
```

#### ▣ Parameters

**propertyName** ([String](#))

Name of the property.

**value** ([JToken](#))

The value.

#### ▣ Return Value

true if a value was successfully retrieved; otherwise, false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **Type**

C# ▼

Gets the node type for this [JToken](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override JTokenType Type { get; }
```

```
Public Overrides ReadOnly Property Type As JTokenType  
Get
```

```
public:  
virtual property JTokenType Type {  
    JTokenType get () override;  
}
```

## ▣ Value

The type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## WriteTo Method (writer, converters)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ►

### WriteTo(JsonWriter, JsonSerializer[])

Writes this token to a [JsonWriter](#).

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteTo(  
    JsonWriter writer,  
    params JsonConverter[] converters  
)
```

```
Public Overrides Sub WriteTo ( _  
    writer As JsonWriter, _  
    ParamArray converters As JsonConverter() _  
)
```

```
public:  
virtual void WriteTo(  
    JsonWriter^ writer,  
    ... array<JsonConverter^>^ converters  
) override
```

#### ▣ Parameters

##### **writer** ([JsonWriter](#))

A [JsonWriter](#) into which this method will write.

##### **converters** ([JsonConverter](#)[])

A collection of [JsonConverter](#) which will be used when writing the token.

(4.5.6.14930)

# JProperty Class

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► **JProperty**

C# ▼

Represents a JSON property.

## Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class JProperty : JContainer
```

```
Public Class JProperty _
    Inherits JContainer
```

```
public ref class JProperty : public JContainer
```

## Members

All Members	Constructors	Methods	Properties	Events
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance		<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static		<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JProperty(JProperty)</a>	Initializes the JProperty class and sets the annotations.
	<a href="#">JProperty(String, Object[])</a>	Initializes the JProperty class with a name and a collection of objects.
	<a href="#">JProperty(String, Object)</a>	Initializes the JProperty class with a name and a single object.

	<a href="#">Add(Object)</a>	Adds as cl  (Inhe
	<a href="#">AddAfterSelf(Object)</a>	Adds imme  (Inhe
	<a href="#">AddBeforeSelf(Object)</a>	Adds imme token  (Inhe
	<a href="#">AddFirst(Object)</a>	Adds as th <a href="#">JTok</a>  (Inhe
	<a href="#">AddingNew</a>	Occu adde  (Inhe
	<a href="#">AfterSelf()</a>	Retu sibir token  (Inhe
	<a href="#">Ancestors()</a>	Retu ance token  (Inhe
	<a href="#">BeforeSelf()</a>	Retu sibir token

		(Inhe
	<a href="#">Children()</a>	Retu child docu  (Inhe
	<a href="#">ChildrenTokens</a>	Gets token  (Ove <a href="#">JCor</a>
	<a href="#">CollectionChanged</a>	Occu of the chan reset  (Inhe
	<a href="#">Count</a>	Gets JSON  (Inhe
	<a href="#">CreateReader()</a>	Crea this t  (Inhe
	<a href="#">CreateWriter()</a>	Crea can l to the  (Inhe
	<a href="#">DeepClone()</a>	Crea the <a href="#">J</a> are r  (Inhe

	<a href="#">Descendants()</a>	Retu desc token  (Inhe
	<a href="#">Equals(Object)</a>	Dete spec the c  (Inhe
	<a href="#">Finalize()</a>	Allow to fre perf oper is rec colle  (Inhe
	<a href="#">First</a>	Get t this t  (Inhe
	<a href="#">GetHashCode()</a>	Serv for a  (Inhe
	<a href="#">GetMetaObject(Expression)</a>	Retu <a href="#">Dyna</a> respo oper this c  (Inhe
	<a href="#">GetType()</a>	Gets insta

		(Inhe
	<a href="#">HasValues</a>	Gets whet child  (Inhe
	<a href="#">IsValid(JToken, JsonSchema)</a>	Dete <a href="#">JTok</a>  (Inhe
	<a href="#">IsValid(JToken, JsonSchema, IList&lt;String&gt;)</a>	Dete <a href="#">JTok</a>  (Inhe
	<a href="#">Item[Object]</a>	Gets spec  (Inhe
	<a href="#">Last</a>	Get t this t  (Inhe
	<a href="#">ListChanged</a>	Occu chan list c  (Inhe
	<a href="#">Load(JsonReader)</a>	Loac <a href="#">Json</a>
	<a href="#">MemberwiseClone()</a>	Cre the c

		(Inhe
	<a href="#">Name</a>	Gets
	<a href="#">Next</a>	Gets of thi  (Inhe
	<a href="#">OnAddingNew(AddingNewEventArgs)</a>	Rais even  (Inhe
	<a href="#">OnCollectionChanged(NotifyCollectionChangedEventArgs)</a>	Rais <a href="#">Colle</a>  (Inhe
	<a href="#">OnListChanged(ListChangedEventArgs)</a>	Rais even  (Inhe
	<a href="#">Parent</a>	Gets  (Inhe
	<a href="#">Previous</a>	Gets token  (Inhe
	<a href="#">Remove()</a>	Rem pare  (Inhe
	<a href="#">RemoveAll()</a>	Rem from

		(Inhe
	<a href="#">Replace(JToken)</a>	Repl spec  (Inhe
	<a href="#">ReplaceAll(Object)</a>	Repl of thi spec  (Inhe
	<a href="#">Root</a>	Gets <a href="#">JTok</a>  (Inhe
	<a href="#">SelectToken(String)</a>	Sele matc  (Inhe
	<a href="#">SelectToken(String, Boolean)</a>	Sele matc  (Inhe
	<a href="#">ToObject&lt;T&gt;()</a>	Crea type  (Inhe
	<a href="#">ToObject&lt;T&gt;(JsonSerializer)</a>	Crea type the s  (Inhe
	<a href="#">ToString()</a>	Retu for th  (Inhe

	<a href="#">ToString(Formatting, JsonConverter[])</a>	Retu token form  (Inhe
	<a href="#">Type</a>	Gets <a href="#">JTok</a>  (Ove
	<a href="#">Validate(JToken, JsonSchema)</a>	Valid <a href="#">JTok</a>  (Inhe
	<a href="#">Validate(JToken, JsonSchema, ValidationEventHandler)</a>	Valid <a href="#">JTok</a>  (Inhe
	<a href="#">Value</a>	Gets value
	<a href="#">Value&lt;T&gt;(Object)</a>	Gets spec the s  (Inhe
	<a href="#">Values&lt;T&gt;()</a>	Retu child docu  (Inhe
	<a href="#">WriteTo(JsonWriter, JsonConverter[])</a>	Write <a href="#">Json</a>  (Ove

## [-] Inheritance Hierarchy

[Object](#)

└─ [JToken](#)

└─ [JContainer](#)

└─ **JProperty**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



C#

## Members

Icon	Member	Description
	<a href="#">JProperty(JProperty)</a>	Initializes a new instance of the <a href="#">JProperty</a> class from another <a href="#">JProperty</a> object.
	<a href="#">JProperty(String, Object[])</a>	Initializes a new instance of the <a href="#">JProperty</a> class.
	<a href="#">JProperty(String, Object)</a>	Initializes a new instance of the <a href="#">JProperty</a> class.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JProperty Constructor (other)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JProperty](#) ►

## JProperty(JProperty)

Initializes a new instance of the [JProperty](#) class from another [JProperty](#) object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JProperty(  
    JProperty other  
)
```

```
Public Sub New ( _  
    other As JProperty _  
)
```

```
public:  
JProperty(  
    JProperty^ other  
)
```

### ▣ Parameters

#### ***other*** ([JProperty](#))

A [JProperty](#) object to copy from.

# JProperty Constructor

## (name, content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JProperty](#) ►

### JProperty(String, Object)

Initializes a new instance of the [JProperty](#) class.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JProperty(     <a href="#">string</a> name,     <a href="#">Object</a> content )</pre>		
	<pre>Public Sub New (      name As <a href="#">String</a>,      content As <a href="#">Object</a>  )</pre>	
		<pre>public: JProperty(     <a href="#">String</a>^ name,     <a href="#">Object</a>^ content )</pre>

#### ▣ Parameters

##### **name** ([String](#))

The property name.

##### **content** ([Object](#))

The property content.

## (name, content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JProperty](#) ►

### JProperty(String, Object[])

Initializes a new instance of the [JProperty](#) class.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JProperty(     string name,     params Object[] content )</pre>		
	<pre>Public Sub New ( _     name As String, _     ParamArray content As Object() _ )</pre>	
		<pre>public: JProperty(     String^ name,     ... array&lt;Object&gt;^ content )</pre>

#### ▣ Parameters

##### **name (String)**

The property name.

##### **content (Object[])**

The property content.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JProperty](#) ► **ChildrenTokens**

C#

Gets the container's children tokens.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
protected override IList<JToken> ChildrenTokens { ge
```

```
Protected Overrides ReadOnly Property ChildrenTokens  
Get
```

```
protected:  
virtual property IList<JToken> ChildrenTokens {  
    IList<JToken> get () override;  
}
```

## Value

The container's children tokens.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JProperty](#) ►

## Load(JsonReader)

Loads an [JProperty](#) from a [JsonReader](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JProperty Load(  
    JsonReader reader  
)
```

```
Public Shared Function Load ( _  
    reader As JsonReader _  
) As JProperty
```

```
public:  
static JProperty^ Load(  
    JsonReader^ reader  
)
```

### ▣ Parameters

#### ***reader*** ([JsonReader](#))

A [JsonReader](#) that will be read for the content of the [JProperty](#).

### ▣ Return Value

A [JProperty](#) that contains the JSON that was read from the specified [JsonReader](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JProperty](#) ► **Name**

C# ▼

Gets the property name.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Name { get; }
```

```
Public ReadOnly Property Name As String  
    Get
```

```
public:  
property String^ Name {  
    String^ get ();  
}
```

## ▣ Value

The property name.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JProperty](#) ► **Type**

Gets the node type for this [JToken](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override JTokenType Type { get; }
```

```
Public Overrides ReadOnly Property Type As JTokenType  
Get
```

```
public:  
virtual property JTokenType Type {  
    JTokenType get () override;  
}
```

## ▣ Value

The type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets the property value.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public JToken Value { get; set; }
```

```
Public Property Value As JToken  
    Get  
    Set
```

```
public:  
property JToken^ Value {  
    JToken^ get ();  
    void set (JToken^ value);  
}
```

## ▣ Value

The property value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## WriteTo Method (writer, converters)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JProperty](#) ►

### WriteTo(JsonWriter, JsonSerializer[])

C#

Writes this token to a [JsonWriter](#).

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteTo(  
    JsonWriter writer,  
    params JsonConverter[] converters  
)
```

```
Public Overrides Sub WriteTo ( _  
    writer As JsonWriter, _  
    ParamArray converters As JsonConverter() _  
)
```

```
public:  
virtual void WriteTo(  
    JsonWriter^ writer,  
    ... array<JsonConverter^>^ converters  
) override
```

#### ▣ Parameters

##### **writer** ([JsonWriter](#))

A [JsonWriter](#) into which this method will write.

##### **converters** ([JsonConverter](#)[])

A collection of [JsonConverter](#) which will be used when writing the token.

(4.5.6.14930)

# JPropertyDescriptor

## Class

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► **JPropertyDescriptor**

Represents a view of a [JProperty](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JPropertyDescriptor : PropertyDescriptor
```

```
Public Class JPropertyDescriptor _  
    Inherits PropertyDescriptor
```

```
public ref class JPropertyDescriptor : public Proper
```

### Members

All Members

Constructors

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declare
- Inherited

Icon	Member	Description
	<a href="#">JPropertyDescriptor(String, Type)</a>	Initializes a new instance of the <b>JPropertyDescriptor</b> class.
	<a href="#">AddValueChanged(Object, EventHandler)</a>	Enables other objects to be notified of property changes.  (Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">AttributeArray</a>	Gets or sets an array of attributes.  (Inherited from <a href="#">MemberDescriptor</a> )

	<a href="#">Attributes</a>	Gets the collection of attribute  (Inherited from <a href="#">MemberDescr</a>
	<a href="#">CanResetValue(Object)</a>	When overridden in a derived whether resetting an object cl  (Overrides <a href="#">PropertyDescriptor.CanReset</a>
	<a href="#">Category</a>	Gets the name of the categor member belongs, as specifie <a href="#">CategoryAttribute</a> .  (Inherited from <a href="#">MemberDescr</a>
	<a href="#">ComponentType</a>	When overridden in a derived of the component this propert  (Overrides <a href="#">PropertyDescriptor</a>
	<a href="#">Converter</a>	Gets the type converter for th  (Inherited from <a href="#">PropertyDesc</a>
	<a href="#">CreateAttributeCollection()</a>	Creates a collection of attribu of attributes passed to the co  (Inherited from <a href="#">MemberDescr</a>
	<a href="#">CreateInstance(Type)</a>	Creates an instance of the sp  (Inherited from <a href="#">PropertyDesc</a>
	<a href="#">Description</a>	Gets the description of the m in the <a href="#">DescriptionAttribute</a> .  (Inherited from <a href="#">MemberDescr</a>
	<a href="#">DesignTimeOnly</a>	Gets whether this member sh design time, as specified in th <a href="#">DesignOnlyAttribute</a> .

		(Inherited from <a href="#">MemberDescr</a> )
	<a href="#">DisplayName</a>	Gets the name that can be di such as a Properties window.  (Inherited from <a href="#">MemberDescr</a> )
	<a href="#">Equals(Object)</a>	Compares this to another obj equivalent.  (Inherited from <a href="#">PropertyDesc</a> )
	<a href="#">FillAttributes(IList)</a>	Adds the attributes of the <a href="#">Pro</a> the specified list of attributes  (Inherited from <a href="#">PropertyDesc</a> )
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to perform other cleanup operat <a href="#">Object</a> is reclaimed by garbage  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetChildProperties()</a>	Returns the default <a href="#">PropertyL</a>  (Inherited from <a href="#">PropertyDesc</a> )
	<a href="#">GetChildProperties(Attribute[])</a>	Returns a <a href="#">PropertyDescriptor</a> specified array of attributes a  (Inherited from <a href="#">PropertyDesc</a> )
	<a href="#">GetChildProperties(Object)</a>	Returns a <a href="#">PropertyDescriptor</a> given object.  (Inherited from <a href="#">PropertyDesc</a> )
	<a href="#">GetChildProperties(Object, Attribute[])</a>	Returns a <a href="#">PropertyDescriptor</a> given object using a specified as a filter.  (Inherited from <a href="#">PropertyDesc</a> )
		

	<a href="#">GetEditor(Type)</a>	Gets an editor of the specified  (Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">GetHashCode()</a>	Returns the hash code for this  (Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">GetInvocationTarget(Type, Object)</a>	This method returns the object used during invocation of method  (Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetTypeFromName(String)</a>	Returns a type using its name  (Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">GetValue(Object)</a>	When overridden in a derived class, returns the current value of the property  (Overrides <a href="#">PropertyDescriptor.GetValue</a> )
	<a href="#">GetValueChangedHandler(Object)</a>	Retrieves the current set of ValueChanged handlers for a specific component  (Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">IsBrowsable</a>	Gets a value indicating whether the property is browsable, as specified in the <a href="#">BrowsableAttribute</a>  (Inherited from <a href="#">MemberDescriptor</a> )
	<a href="#">IsLocalizable</a>	Gets a value indicating whether the property should be localized, as specified in the <a href="#">LocalizableAttribute</a>  (Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">IsReadOnly</a>	When overridden in a derived class, returns a value indicating whether the property is read-only

		indicating whether this property is serializable. (Overrides <a href="#">PropertyDescriptor.IsSerializable</a> )
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the object. (Inherited from <a href="#">Object</a> .)
	<a href="#">Name</a>	Gets the name of the member. (Inherited from <a href="#">MemberDescriptor</a> .)
	<a href="#">NameHashCode</a>	Gets the hash code for the name. (Overrides <a href="#">MemberDescriptor.NameHashCode</a> )
	<a href="#">OnValueChanged(Object, EventArgs)</a>	Raises the ValueChanged event when the value is implemented. (Inherited from <a href="#">PropertyDescriptor</a> .)
	<a href="#">PropertyType</a>	When overridden in a derived class, gets the type of the property. (Overrides <a href="#">PropertyDescriptor.PropertyType</a> )
	<a href="#">RemoveValueChanged(Object, EventHandler)</a>	Enables other objects to be notified of property changes. (Inherited from <a href="#">PropertyDescriptor</a> .)
	<a href="#">ResetValue(Object)</a>	When overridden in a derived class, sets the current value for this property of the object to the default value. (Overrides <a href="#">PropertyDescriptor.ResetValue(Object)</a> )
	<a href="#">SerializationVisibility</a>	Gets a value indicating whether the property should be serialized, as specified by <a href="#">DesignerSerializationVisibilityAttribute</a> . (Inherited from <a href="#">PropertyDescriptor</a> .)

	<a href="#">SetValue(Object, Object)</a>	When overridden in a derived value of the component to a c  (Overrides <a href="#">PropertyDescriptor.Object</a> .)
	<a href="#">ShouldSerializeValue(Object)</a>	When overridden in a derived value indicating whether the v needs to be persisted.  (Overrides <a href="#">PropertyDescriptor.ShouldSe</a>
	<a href="#">SupportsChangeEvents</a>	Gets a value indicating wheth notifications for this property outside the property descripto  (Inherited from <a href="#">PropertyDesc</a>
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represe <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

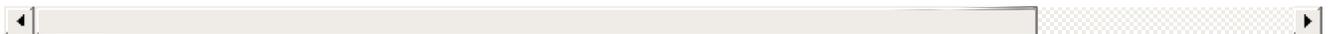
└─ [MemberDescriptor](#)

└─ [PropertyDescriptor](#)

└─ **JPropertyDescriptor**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# JPropertyDescriptor

## Constructor (name, propertyType)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JPropertyDescriptor](#) ►

### JPropertyDescriptor(String, Type)

Initializes a new instance of the [JPropertyDescriptor](#) class.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JPropertyDescriptor(     <a href="#">string</a> name,     <a href="#">Type</a> propertyType )</pre>		
	<pre>Public Sub New ( _     name As <a href="#">String</a>, _     propertyType As <a href="#">Type</a> _ )</pre>	
		<pre>public: JPropertyDescriptor(     <a href="#">String</a>^ name,     <a href="#">Type</a>^ propertyType )</pre>

#### ▣ Parameters

##### **name** ([String](#))

The name.

##### **propertyType** ([Type](#))

Type of the property.

## (component)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JPropertyDescriptor](#) ►

### CanResetValue(Object)

When overridden in a derived class, returns whether resetting an object changes its value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override bool CanResetValue(  
    Object component  
)
```

```
Public Overrides Function CanResetValue ( _  
    component As Object _  
) As Boolean
```

```
public:  
virtual bool CanResetValue(  
    Object^ component  
) override
```

#### ▣ Parameters

##### **component** ([Object](#))

The component to test for reset capability.

#### ▣ Return Value

true if resetting the component changes its value; otherwise, false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JPropertyDescriptor](#) ►

## ComponentType

When overridden in a derived class, gets the type of the component this property is bound to. C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Type ComponentType { get; }
```

```
Public Overrides ReadOnly Property ComponentType As  
    Get
```

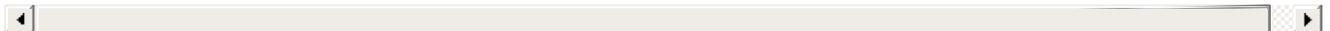
```
public:  
virtual property Type^ ComponentType {  
    Type^ get () override;  
}
```

### Return Value

A [Type](#) that represents the type of component this property is bound to. When the [GetValue\(Object\)](#) or [SetValue\(Object, Object\)](#) methods are invoked, the object specified might be an instance of this type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## (component)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JPropertyDescriptor](#) ►

### GetValue(Object)

When overridden in a derived class, gets the current value of the property on a component. C#

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Object GetValue(  
    Object component  
)
```

```
Public Overrides Function GetValue ( _  
    component As Object _  
) As Object
```

```
public:  
virtual Object^ GetValue(  
    Object^ component  
) override
```

#### Parameters

##### **component** ([Object](#))

The component with the property for which to retrieve the value.

#### Return Value

The value of a property for a given component.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JPropertyDescriptor](#) ►

## IsReadOnly

When overridden in a derived class, gets a value indicating whether this property is read-only.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public override bool IsReadOnly { get; }
```

```
Public Overrides ReadOnly Property IsReadOnly As Bool  
    Get
```

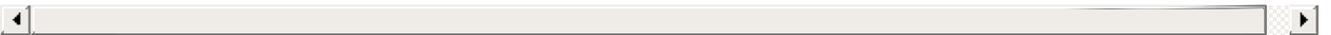
```
public:  
virtual property bool IsReadOnly {  
    bool get () override;  
}
```

### Return Value

true if the property is read-only; otherwise, false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JPropertyDescriptor](#) ►

## NameHashCode

C#

Gets the hash code for the name of the member.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected override int NameHashCode { get; }
```

```
Protected Overrides ReadOnly Property NameHashCode A  
Get
```

```
protected:  
virtual property int NameHashCode {  
    int get () override;  
}
```

### Value

### Return Value

The hash code for the name of the member.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JPropertyDescriptor](#) ►

## PropertyType

When overridden in a derived class, gets the type of the property.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Type PropertyType { get; }
```

```
Public Overrides ReadOnly Property PropertyType As I  
Get
```

```
public:  
virtual property Type^ PropertyType {  
    Type^ get () override;  
}
```

### ▣ Return Value

A [Type](#) that represents the type of the property.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## (component)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JPropertyDescriptor](#) ►

### ResetValue(Object)

When overridden in a derived class, resets the value for this property of the component to the default value.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public override void ResetValue(     <a href="#">Object</a> component )</pre>	<pre>Public Overrides Sub ResetValue (      component As <a href="#">Object</a>  )</pre>	<pre>public: virtual void ResetValue(     <a href="#">Object</a>^ component ) override</pre>

#### ▣ Parameters

##### **component** ([Object](#))

The component with the property value that is to be reset to the default value.

**(component, value)**[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JPropertyDescriptor](#) ►**SetValue(Object, Object)**

C#

When overridden in a derived class, sets the value of the component to different value.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void SetValue(  
    Object component,  
    Object value  
)
```

```
Public Overrides Sub SetValue ( _  
    component As Object, _  
    value As Object _  
)
```

```
public:  
virtual void SetValue(  
    Object^ component,  
    Object^ value  
) override
```

**Parameters****component ([Object](#))**

The component with the property value that is to be set.

**value ([Object](#))**

The new value.

(4.5.6.14930)

## Method (component)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JPropertyDescriptor](#) ►

### ShouldSerializeValue(Object)

When overridden in a derived class, determines a value indicating whether the value of this property needs to be persisted.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public override bool ShouldSerializeValue(  
    Object component  
)
```

```
Public Overrides Function ShouldSerializeValue ( _  
    component As Object _  
) As Boolean
```

```
public:  
virtual bool ShouldSerializeValue(  
    Object^ component  
) override
```

#### Parameters

##### **component** ([Object](#))

The component with the property to be examined for persistence.

#### Return Value

true if the property should be persisted; otherwise, false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Represents a raw JSON string.

## Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class JRaw : JValue
```

```
Public Class JRaw _
    Inherits JValue
```

```
public ref class JRaw : public JValue
```

## Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
			<input checked="" type="checkbox"/> Declare
			<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JRaw(JRaw)</a>	Initializes a new instance of the <b>JRaw</b> class from another <b>JRaw</b> object.
	<a href="#">JRaw(Object)</a>	Initializes a new instance of the <b>JRaw</b> class.
	<a href="#">AddAfterSelf(Object)</a>	Adds the specified content immediately after this token.

		(Inherited from <a href="#">JToken</a> .)
	<a href="#">AddBeforeSelf(Object)</a>	Adds the specified content immediately before this token.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">AfterSelf()</a>	Returns a collection of the sibling tokens after this token, in document order.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Ancestors()</a>	Returns a collection of the ancestor tokens of this token.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">BeforeSelf()</a>	Returns a collection of the sibling tokens before this token, in document order.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Children()</a>	Returns a collection of the child tokens of this token, in document order.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">CompareTo(JValue)</a>	Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.  (Inherited from <a href="#">JValue</a> .)
	<a href="#">Create(JsonReader)</a>	Creates an instance of <b>JRaw</b> with the content of the reader's current

		token.
	<a href="#">CreateReader()</a>	Creates an <a href="#">JsonReader</a> for this token.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">DeepClone()</a>	Creates a new instance of the <a href="#">JToken</a> . All child tokens are recursively cloned.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Equals(JValue)</a>	Indicates whether the current object is equal to another object of the same type.  (Inherited from <a href="#">JValue</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">JValue</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">First</a>	Get the first child token of this token.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a

		particular type.  (Inherited from <a href="#">JValue</a> .)
	<a href="#">GetMetaObject(Expression)</a>	Returns the <a href="#">DynamicMetaObject</a> responsible for binding operations performed on this object.  (Inherited from <a href="#">JValue</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">HasValues</a>	Gets a value indicating whether this token has children tokens.  (Inherited from <a href="#">JValue</a> .)
	<a href="#">IsValid(JToken, JsonSchema)</a>	Determines whether the <a href="#">JToken</a> is valid.  (Inherited from <a href="#">Extensions</a> .)
	<a href="#">IsValid(JToken, JsonSchema, IList&lt;String&gt;)</a>	Determines whether the <a href="#">JToken</a> is valid.  (Inherited from <a href="#">Extensions</a> .)
	<a href="#">Item[Object]</a>	Gets the <a href="#">JToken</a> with the specified key.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Last</a>	Get the last child token of this token.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .

		(Inherited from <a href="#">Object</a> .)
	<a href="#">Next</a>	Gets the next sibling token of this node.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Parent</a>	Gets or sets the parent.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Previous</a>	Gets the previous sibling token of this node.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Remove()</a>	Removes this token from its parent  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Replace(JToken)</a>	Replaces this token with the specified token.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Root</a>	Gets the root <a href="#">JToken</a> of this <a href="#">JToken</a>  (Inherited from <a href="#">JToken</a> .)
	<a href="#">SelectToken(String)</a>	Selects the token that matches the object path.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">SelectToken(String, Boolean)</a>	Selects the token that matches the object path.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">ToObject&lt;T&gt;()</a>	Creates the specified .NET type from the <a href="#">JToken</a> .  (Inherited from <a href="#">JToken</a> .)

	<a href="#">ToObject&lt;T&gt;(JsonSerializer)</a>	Creates the specified .NET type from the <a href="#">JToken</a> using the specified <a href="#">JsonSerializer</a> .  (Inherited from <a href="#">JToken</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents this instance.  (Inherited from <a href="#">JValue</a> .)
	<a href="#">ToString(String)</a>	Returns a <a href="#">String</a> that represents this instance.  (Inherited from <a href="#">JValue</a> .)
	<a href="#">ToString(IFormatProvider)</a>	Returns a <a href="#">String</a> that represents this instance.  (Inherited from <a href="#">JValue</a> .)
	<a href="#">ToString(String, IFormatProvider)</a>	Returns a <a href="#">String</a> that represents this instance.  (Inherited from <a href="#">JValue</a> .)
	<a href="#">ToString(Formatting, JsonSerializer[])</a>	Returns the JSON for this token using the given formatting and converters.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Type</a>	Gets the node type for this <a href="#">JToken</a> .  (Inherited from <a href="#">JValue</a> .)
	<a href="#">Validate(JToken, JsonSchema)</a>	Validates the specified <a href="#">JToken</a> .  (Inherited from <a href="#">Extensions</a> .)
	<a href="#">Validate(JToken, JsonSchema, ValidationEventHandler)</a>	Validates the specified <a href="#">JToken</a> .  (Inherited from <a href="#">Extensions</a> .)

	<a href="#">Value</a>	Gets or sets the underlying token value.  (Inherited from <a href="#">JValue</a> .)
	<a href="#">Value&lt;T&gt;(Object)</a>	Gets the <a href="#">JToken</a> with the specified key converted to the specified type  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Values&lt;T&gt;()</a>	Returns a collection of the child values of this token, in document order.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">WriteTo(JsonWriter, JsonSerializer[])</a>	Writes this token to a <a href="#">JsonWriter</a> .  (Inherited from <a href="#">JValue</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JToken](#)

└─ [JValue](#)

└─ **JRaw**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JRaw](#) ► **JRaw()**

C# ▼

[-] **Members**

Icon	Member	Description
	<a href="#">JRaw(JRaw)</a>	Initializes a new instance of the <a href="#">JRaw</a> class from another <a href="#">JRaw</a> object.
	<a href="#">JRaw(Object)</a>	Initializes a new instance of the <a href="#">JRaw</a> class.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JRaw](#) ► [JRaw\(JRaw\)](#)

Initializes a new instance of the [JRaw](#) class from another [JRaw](#) object.

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JRaw(  
    JRaw other  
)
```

```
Public Sub New ( _  
    other As JRaw _  
)
```

```
public:  
JRaw(  
    JRaw^ other  
)
```

## Parameters

### ***other*** ([JRaw](#))

A [JRaw](#) object to copy from.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JRaw Constructor (rawJson)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JRaw](#) ► **JRaw(Object)**

C#

Initializes a new instance of the [JRaw](#) class.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JRaw(  
    Object rawJson  
)
```

```
Public Sub New (   
    rawJson As Object   
)
```

```
public:  
JRaw(  
    Object^ rawJson  
)
```

## ▣ Parameters

***rawJson*** ([Object](#))

The raw json.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JRaw](#) ►

## Create(JsonReader)

Creates an instance of [JRaw](#) with the content of the reader's current token.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static <a href="#">JRaw</a> Create(     <a href="#">JsonReader</a> reader )</pre>		
	<pre>Public Shared Function Create ( _     reader As <a href="#">JsonReader</a> _ ) As <a href="#">JRaw</a></pre>	
		<pre>public: static <a href="#">JRaw</a>^ Create(     <a href="#">JsonReader</a>^ reader )</pre>

### ▣ Parameters

***reader*** ([JsonReader](#))

The reader.

### ▣ Return Value

An instance of [JRaw](#) with the content of the reader's current token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JToken Class

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► **JToken**

Represents an abstract JSON token.

## Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public abstract class JToken : <a href="#">IEnumerable&lt;JToken&gt;</a>,     <a href="#">IEnumerable&lt;JToken&gt;</a>, <a href="#">IEnumerable</a>, <a href="#">IJsonLineInfo</a>,     <a href="#">IDynamicMetaObjectProvider</a></pre>		

```
Public MustInherit Class JToken _
    Implements IEnumerable\(Of JToken\), IEnumerable,
    IEnumerable, IJsonLineInfo, ICloneable, IDyn
```

```
public ref class JToken abstract : IEnumerable<JTok,
    IEnumerable<JToken^>, IEnumerable, IJsonLine,
    IDynamicMetaObjectProvider
```

## Members

All Members	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static

Icon	Member	Description
	<a href="#">AddAfterSelf(Object)</a>	Adds the specified content immediately after this token.
	<a href="#">AddBeforeSelf(Object)</a>	Adds the specified content immediately

		before this token.
	<a href="#">AfterSelf()</a>	Returns a collection of the sibling tokens after this token, in document order.
	<a href="#">Ancestors()</a>	Returns a collection of the ancestor tokens of this token.
	<a href="#">Ancestors&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns a collection of tokens that contains the ancestors of every token in the source collection.  (Inherited from <a href="#">Extensions.</a> )
	<a href="#">AsJEnumerable(IEnumerable&lt;JToken&gt;)</a>	Returns the input typed as <a href="#">IEnumerable&lt;T&gt;</a> .  (Inherited from <a href="#">Extensions.</a> )
	<a href="#">AsJEnumerable&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns the input typed as <a href="#">IEnumerable&lt;T&gt;</a> .  (Inherited from <a href="#">Extensions.</a> )
	<a href="#">BeforeSelf()</a>	Returns a collection of the sibling tokens before this token, in document order.

	<a href="#">Children()</a>	Returns a collection of the child tokens of this token, in document order.
	<a href="#">Children&lt;T&gt;()</a>	Returns a collection of the child tokens of this token, in document order, filtered by the specified type.
	<a href="#">Children&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Returns a collection of converted child tokens of every array in the source collection.  (Inherited from <a href="#">Extensions.</a> )
	<a href="#">CreateReader()</a>	Creates an <a href="#">JsonReader</a> for this token.
	<a href="#">DeepClone()</a>	Creates a new instance of the <b>JToken</b> . All child tokens are recursively cloned.
 	<a href="#">DeepEquals(JToken, JToken)</a>	Compares the values of two tokens, including the values of all descendant tokens.

	<a href="#">EqualityComparer</a>	Gets a comparer that can compare two tokens for value equality.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Explicit(JToken to Boolean)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Boolean</a> .
	<a href="#">Explicit(JToken to DateTimeOffset)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">DateTimeOffset</a> .
	<a href="#">Explicit(JToken to Nullable&lt;Boolean&gt;)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Int64)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Int64</a> .
	<a href="#">Explicit(JToken to Nullable&lt;DateTime&gt;)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Nullable&lt;T&gt;</a> .

	<a href="#">Explicit(JToken to Nullable&lt;DateTimeOffset&gt;)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Nullable&lt;Decimal&gt;)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Nullable&lt;Double&gt;)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Int32)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Int32</a> .
	<a href="#">Explicit(JToken to Int16)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Int16</a> .
	<a href="#">Explicit(JToken to UInt16)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">UInt16</a> .
	<a href="#">Explicit(JToken to Nullable&lt;Int32&gt;)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Nullable&lt;T&gt;</a> .

	<a href="#">Explicit(JToken to Nullable&lt;Int16&gt;)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Nullable&lt;UInt16&gt;)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to DateTime)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">DateTime</a> .
	<a href="#">Explicit(JToken to Nullable&lt;Int64&gt;)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Nullable&lt;Single&gt;)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Decimal)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Decimal</a> .
	<a href="#">Explicit(JToken to Nullable&lt;UInt32&gt;)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Nullable&lt;T&gt;</a> .

	<a href="#">Explicit(JToken to Nullable&lt;UInt64&gt;)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Double)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Double</a> .
	<a href="#">Explicit(JToken to Single)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">Single</a> .
	<a href="#">Explicit(JToken to String)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">String</a> .
	<a href="#">Explicit(JToken to UInt32)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">UInt32</a> .
	<a href="#">Explicit(JToken to UInt64)</a>	Performs an explicit conversion from <b>JToken</b> to <a href="#">UInt64</a> .
	<a href="#">Explicit(JToken to Byte[])</a>	Performs an explicit conversion from <b>JToken</b> to <b>Byte[]</b> .

	<a href="#">Finalize()</a>	<p>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operation before the <a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">First</a>	Get the first child token of this token.
	<a href="#">FromObject(Object)</a>	Creates a <b>JToken</b> from an object.
	<a href="#">FromObject(Object, JsonSerializer)</a>	Creates a <b>JToken</b> from an object using the specified <a href="#">JsonSerializer</a>
	<a href="#">GetHashCode()</a>	<p>Serves as a hash function for a particular type.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetMetaObject(Expression)</a>	Returns the <a href="#">DynamicMetaObject</a> responsible for binding operations performed on this object.

	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">HasValues</a>	Gets a value indicating whether this token has children tokens.
	<a href="#">Implicit(Boolean to JToken)</a>	Performs an implicit conversion from <a href="#">Boolean</a> to <b>JToken</b> .
	<a href="#">Implicit(DateTimeOffset to JToken)</a>	Performs an implicit conversion from <a href="#">DateTimeOffset</a> to <b>JToken</b> .
	<a href="#">Implicit(Nullable&lt;Boolean&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .
	<a href="#">Implicit(Int64 to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .
	<a href="#">Implicit(Nullable&lt;DateTime&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .

 <a href="#">Implicit(Nullable&lt;DateTimeOffset&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .
 <a href="#">Implicit(Nullable&lt;Decimal&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .
 <a href="#">Implicit(Nullable&lt;Double&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .
 <a href="#">Implicit(Int16 to JToken)</a>	Performs an implicit conversion from <a href="#">Int16</a> to <b>JToken</b> .
 <a href="#">Implicit(UInt16 to JToken)</a>	Performs an implicit conversion from <a href="#">UInt16</a> to <b>JToken</b> .
 <a href="#">Implicit(Int32 to JToken)</a>	Performs an implicit conversion from <a href="#">Int32</a> to <b>JToken</b> .
 <a href="#">Implicit(Nullable&lt;Int32&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .

	<a href="#">Implicit(DateTime to JToken)</a>	Performs an implicit conversion from <a href="#">DateTime</a> to <b>JToken</b> .
	<a href="#">Implicit(Nullable&lt;Int64&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .
	<a href="#">Implicit(Nullable&lt;Single&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .
	<a href="#">Implicit(Decimal to JToken)</a>	Performs an implicit conversion from <a href="#">Decimal</a> to <b>JToken</b> .
	<a href="#">Implicit(Nullable&lt;Int16&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .
	<a href="#">Implicit(Nullable&lt;UInt16&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .
	<a href="#">Implicit(Nullable&lt;UInt32&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .

	<a href="#">Implicit(Nullable&lt;UInt64&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .
	<a href="#">Implicit(Double to JToken)</a>	Performs an implicit conversion from <a href="#">Double</a> to <b>JToken</b> .
	<a href="#">Implicit(Single to JToken)</a>	Performs an implicit conversion from <a href="#">Single</a> to <b>JToken</b> .
	<a href="#">Implicit(String to JToken)</a>	Performs an implicit conversion from <a href="#">String</a> to <b>JToken</b> .
	<a href="#">Implicit(UInt32 to JToken)</a>	Performs an implicit conversion from <a href="#">UInt32</a> to <b>JToken</b> .
	<a href="#">Implicit(UInt64 to JToken)</a>	Performs an implicit conversion from <a href="#">UInt64</a> to <b>JToken</b> .
	<a href="#">Implicit(Byte[] to JToken)</a>	Performs an implicit conversion from <b>Byte[]</b> to <b>JToken</b> .

	<a href="#">IsValid(JToken, JsonSchema)</a>	Determines whether the <b>JToken</b> is valid.  (Inherited from <a href="#">Extensions.</a> )
	<a href="#">IsValid(JToken, JsonSchema, IList&lt;String&gt;)</a>	Determines whether the <b>JToken</b> is valid.  (Inherited from <a href="#">Extensions.</a> )
	<a href="#">Item[Object]</a>	Gets the <b>JToken</b> with the specified key.
	<a href="#">Last</a>	Get the last child token of this token.
 	<a href="#">Load(JsonReader)</a>	Creates a <b>JToken</b> from a <a href="#">JsonReader</a> .
 	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">Next</a>	Gets the next sibling token of this node.
	<a href="#">Parent</a>	Gets or sets the parent.
 	<a href="#">Parse(String)</a>	Load a <b>JToken</b> from a

		string that contains JSON.
	<a href="#">Previous</a>	Gets the previous sibling token of this node.
	<a href="#">ReadFrom(JsonReader)</a>	Creates a <b>JToken</b> from a <a href="#">JsonReader</a> .
	<a href="#">Remove()</a>	Removes this token from its parent.
	<a href="#">Replace(JToken)</a>	Replaces this token with the specified token.
	<a href="#">Root</a>	Gets the root <b>JToken</b> of this <b>JToken</b> .
	<a href="#">SelectToken(String)</a>	Selects the token that matches the object path
	<a href="#">SelectToken(String, Boolean)</a>	Selects the token that matches the object path
	<a href="#">ToObject&lt;T&gt;()</a>	Creates the specified .NET type from the

		<b>JToken.</b>
	<a href="#">ToObject&lt;T&gt;(JsonSerializer)</a>	Creates the specified .NET type from the <b>JToken</b> using the specified <a href="#">JsonSerializer</a>
	<a href="#">ToString()</a>	Returns the indented JSON for this token.  (Overrides <a href="#">Object.ToString()</a> .)
	<a href="#">ToString(Formatting, JsonSerializer[])</a>	Returns the JSON for this token using the given formatting and converters.
	<a href="#">Type</a>	Gets the node type for this <b>JToken</b> .
	<a href="#">Validate(JToken, JsonSchema)</a>	Validates the specified <b>JToken</b> .  (Inherited from <a href="#">Extensions</a> .)
	<a href="#">Validate(JToken, JsonSchema, ValidationEventHandler)</a>	Validates the specified <b>JToken</b> .  (Inherited from <a href="#">Extensions</a> .)
	<a href="#">Value&lt;T&gt;(Object)</a>	Gets the <b>JToken</b> with

		the specified key converted to the specified type.
	<a href="#">Value&lt;U&gt;(IEnumerable&lt;JToken&gt;)</a>	Converts the value.  (Inherited from <a href="#">Extensions.</a> )
	<a href="#">Value&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Converts the value.  (Inherited from <a href="#">Extensions.</a> )
	<a href="#">Values&lt;T&gt;()</a>	Returns a collection of the child values of this token, in document order.
	<a href="#">Values(IEnumerable&lt;JToken&gt;, Object)</a>	Returns a collection of child values of every object in the source collection with the given key.  (Inherited from <a href="#">Extensions.</a> )
	<a href="#">Values(IEnumerable&lt;JToken&gt;)</a>	Returns a collection of child values of every object in the source collection.  (Inherited from <a href="#">Extensions.</a> )
	<a href="#">Values&lt;U&gt;(IEnumerable&lt;JToken&gt;,</a>	Returns a collection of

	<a href="#">Object</a> )	converted child values of every object in the source collection with the given key.  (Inherited from <a href="#">Extensions</a> .)
	<a href="#">WriteTo(JsonWriter, JsonSerializer[])</a>	Writes this token to a <a href="#">JsonWriter</a> .

## ▣ Inheritance Hierarchy

[Object](#)

└─ **JToken**

└─ [JContainer](#)

└─ [JValue](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



**(content)**[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►**AddAfterSelf(Object)**

Adds the specified content immediately after this token.

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void AddAfterSelf(  
    Object content  
)
```

```
Public Sub AddAfterSelf ( _  
    content As Object _  
)
```

```
public:  
void AddAfterSelf(  
    Object^ content  
)
```

**▣ Parameters*****content* ([Object](#))**

A content object that contains simple content or a collection of content objects to be added after this token.

**(content)**[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►**AddBeforeSelf(Object)**

Adds the specified content immediately before this token.

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public void AddBeforeSelf(  
    Object content  
)
```

```
Public Sub AddBeforeSelf ( _  
    content As Object _  
)
```

```
public:  
void AddBeforeSelf(  
    Object^ content  
)
```

**▣ Parameters*****content* ([Object](#))**

A content object that contains simple content or a collection of content objects to be added before this token.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **AfterSelf()**

C#

Returns a collection of the sibling tokens after this token, in document order.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IEnumerable<JToken> AfterSelf()
```

```
Public Function AfterSelf As IEnumerable(Of JToken)
```

```
public:  
IEnumerable<JToken>^ AfterSelf()
```

### ▣ Return Value

A collection of the sibling tokens after this tokens, in document order.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Ancestors()**

C#

Returns a collection of the ancestor tokens of this token.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IEnumerable<JToken> Ancestors()
```

```
Public Function Ancestors As IEnumerable(Of JToken)
```

```
public:  
IEnumerable<JToken> Ancestors()
```

## ▣ Return Value

A collection of the ancestor tokens of this token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **BeforeSelf()**

C#

Returns a collection of the sibling tokens before this token, in document order.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IEnumerable<JToken> BeforeSelf()
```

```
Public Function BeforeSelf As IEnumerable(Of JToken)
```

```
public:  
IEnumerable<JToken> BeforeSelf()
```

## ▣ Return Value

A collection of the sibling tokens before this token, in document order.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Members

Icon	Member	Description
	<a href="#">Children()</a>	Returns a collection of the child tokens of this token, in document order.
	<a href="#">Children&lt;T&gt;()</a>	Returns a collection of the child tokens of this token, in document order, filtered by the specified type.
	<a href="#">Children&lt;T, U&gt; (IEnumerable&lt;T&gt;)</a>	Returns a collection of converted child tokens of every array in the source collection.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Children()**

C#

Returns a collection of the child tokens of this token, in document order.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual IEnumerable<JToken> Children()
```

```
Public Overridable Function Children As IEnumerable(<
```

```
public:  
virtual IEnumerable<JToken^> Children()
```

## Return Value

An [IEnumerable<T>](#) of [JToken](#) containing the child tokens of this [JToken](#) in document order.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Children<T>()**

C#

Returns a collection of the child tokens of this token, in document order, filtered by the specified type.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JEnumerable<T> Children<T>()  
where T : JToken
```

```
Public Function Children(Of T As JToken) As JEnumera
```

```
public:  
generic<typename T>  
where T : JToken  
JEnumerable<T> Children()
```

### ▣ Generic Template Parameters

***T***

The type to filter the child tokens on.

### ▣ Return Value

A [JEnumerable<T>](#) containing the child tokens of this [JToken](#), in document order.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **CreateReader()**

C#

Creates an [JsonReader](#) for this token.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonReader CreateReader()
```

```
Public Function CreateReader As JsonReader
```

```
public:  
JsonReader^ CreateReader()
```

## ▣ Return Value

An [JsonReader](#) that can be used to read this token and its descendants

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **DeepClone()**

Creates a new instance of the [JToken](#). All child tokens are recursively cloned.

C#

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JToken DeepClone()
```

```
Public Function DeepClone As JToken
```

```
public:  
JToken^ DeepClone()
```

## ▣ Return Value

A new instance of the [JToken](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## DeepEquals Method (t1, t2)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

### DeepEquals(JToken, JToken)

Compares the values of two tokens, including the values of all descendant tokens.

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static bool DeepEquals(     JToken t1,     JToken t2 )</pre>		
	<pre>Public Shared Function DeepEquals ( _     t1 As JToken, _     t2 As JToken _ ) As Boolean</pre>	
		<pre>public: static bool DeepEquals(     JToken^ t1,     JToken^ t2 )</pre>

#### Parameters

##### **t1** ([JToken](#))

The first [JToken](#) to compare.

##### **t2** ([JToken](#))

The second [JToken](#) to compare.

#### Return Value

true if the tokens are equal; otherwise false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **EqualityComparer**

Gets a comparer that can compare two tokens for value equality. C# ▼

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JTokenEqualityComparer EqualityCompare
```

```
Public Shared ReadOnly Property EqualityComparer As  
    Get
```

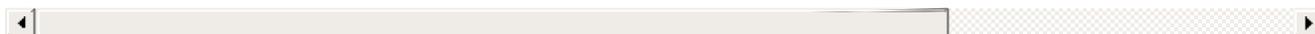
```
public:  
static property JTokenEqualityComparer^ EqualityComp  
    JTokenEqualityComparer^ get ();  
}
```

### Value

A [JTokenEqualityComparer](#) that can compare two nodes for value equality.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)



[-] **Members**

Icon	Member	Description
	<a href="#">Explicit(JToken to Boolean)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Boolean</a> .
	<a href="#">Explicit(JToken to DateTimeOffset)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">DateTimeOffset</a> .
	<a href="#">Explicit(JToken to Nullable&lt;Boolean&gt;)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Int64)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Int64</a> .
	<a href="#">Explicit(JToken to Nullable&lt;DateTime&gt;)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Nullable&lt;DateTimeOffset&gt;)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Nullable&lt;Decimal&gt;)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to</a>	Performs an explicit conversion from

	<a href="#">Nullable&lt;Double&gt;</a> )	<a href="#">JToken</a> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Int32)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Int32</a> .
	<a href="#">Explicit(JToken to Int16)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Int16</a> .
	<a href="#">Explicit(JToken to UInt16)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">UInt16</a> .
	<a href="#">Explicit(JToken to Nullable&lt;Int32&gt;)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Nullable&lt;Int16&gt;)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Nullable&lt;UInt16&gt;)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to DateTime)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">DateTime</a> .
	<a href="#">Explicit(JToken to Nullable&lt;Int64&gt;)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Nullable&lt;T&gt;</a> .

	<a href="#">Explicit(JToken to Nullable&lt;Single&gt;)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Decimal)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Decimal</a> .
	<a href="#">Explicit(JToken to Nullable&lt;UInt32&gt;)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Nullable&lt;UInt64&gt;)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Nullable&lt;T&gt;</a> .
	<a href="#">Explicit(JToken to Double)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Double</a> .
	<a href="#">Explicit(JToken to Single)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Single</a> .
	<a href="#">Explicit(JToken to String)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">String</a> .
	<a href="#">Explicit(JToken to UInt32)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">UInt32</a> .
	<a href="#">Explicit(JToken to UInt64)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">UInt64</a> .

	<a href="#">Explicit(JToken to Byte[])</a>	Performs an explicit conversion from <a href="#">JToken</a> to <b>Byte[]</b> .
---	--	--

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Decimal)**

Performs an explicit conversion from [JToken](#) to [Decimal](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator decimal (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Decimal
```

```
static explicit operator Decimal (  
    JToken^ value  
)
```

## ▣ Parameters

**value ([JToken](#))**

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<UInt32>)**

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static explicit operator Nullable&lt;uint&gt; (     JToken value )</pre>		
	<pre>Public Shared Narrowing Operator CType ( _     value As JToken _ ) As Nullable(Of UInteger)</pre>	
		<pre>static explicit operator Nullable&lt;unsigned int&gt; (     JToken^ value )</pre>

## ▣ Parameters

***value*** ([JToken](#))

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<UInt64>)**

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator Nullable<ulong> (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Nullable(Of ULong)
```

```
static explicit operator Nullable<unsigned long long  
    JToken^ value  
)
```

## Parameters

**value (JToken)**

The value.

## Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to DateTime)**

Performs an explicit conversion from [JToken](#) to [DateTime](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static explicit operator DateTime (     JToken value )</pre>		
	<pre>Public Shared Narrowing Operator CType ( _     value As JToken _ ) As DateTime</pre>	
		<pre>static explicit operator DateTime (     JToken^ value )</pre>

## ▣ Parameters

***value*** ([JToken](#))

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<Int64>)**

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static explicit operator Nullable&lt;long&gt; (     JToken value )</pre>		
	<pre>Public Shared Narrowing Operator CType ( _     value As JToken _ ) As Nullable(Of Long)</pre>	
		<pre>static explicit operator Nullable&lt;long long&gt; (     JToken^ value )</pre>

## ▣ Parameters

***value*** ([JToken](#))

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<Single>)**

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static explicit operator Nullable&lt;float&gt; (     JToken value )</pre>		
	<pre>Public Shared Narrowing Operator CType ( _     value As JToken _ ) As Nullable(Of Single)</pre>	
		<pre>static explicit operator Nullable&lt;float&gt; (     JToken^ value )</pre>

## ▣ Parameters

***value*** ([JToken](#))

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to UInt32)**

Performs an explicit conversion from [JToken](#) to [UInt32](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static explicit operator uint (     JToken value )</pre>		
	<pre>Public Shared Narrowing Operator CType ( _     value As JToken _ ) As UIntInteger</pre>	
		<pre>static explicit operator unsigned int (     JToken^ value )</pre>

## ▣ Parameters

***value*** ([JToken](#))

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to UInt64)**

Performs an explicit conversion from [JToken](#) to [UInt64](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator ulong (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As ULong
```

```
static explicit operator unsigned long long (  
    JToken^ value  
)
```

## ▣ Parameters

**value ([JToken](#))**

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Byte[])**

Performs an explicit conversion from [JToken](#) to **Byte[]**.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static explicit operator byte[] (     <a href="#">JToken</a> value )</pre>		
<pre>Public Shared Narrowing Operator CType ( _     value As <a href="#">JToken</a> _ ) As <a href="#">Byte</a>()</pre>		
<pre>static explicit operator array&lt;<a href="#">unsigned char</a>&gt;^ (     <a href="#">JToken</a>^ value )</pre>		

## ▣ Parameters

**value ([JToken](#))**

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Double)**

Performs an explicit conversion from [JToken](#) to [Double](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator double (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Double
```

```
static explicit operator double (  
    JToken^ value  
)
```

## ▣ Parameters

**value ([JToken](#))**

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Single)**

Performs an explicit conversion from [JToken](#) to [Single](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator float (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Single
```

```
static explicit operator float (  
    JToken^ value  
)
```

## ▣ Parameters

***value*** ([JToken](#))

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to String)**

Performs an explicit conversion from [JToken](#) to [String](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator string (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As String
```

```
static explicit operator String^ (  
    JToken^ value  
)
```

## ▣ Parameters

**value ([JToken](#))**

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<UInt16>)**

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static explicit operator Nullable&lt;ushort&gt; (     JToken value )</pre>		
	<pre>Public Shared Narrowing Operator CType ( _     value As JToken _ ) As Nullable(Of UShort)</pre>	
		<pre>static explicit operator Nullable&lt;unsigned short&gt; (     JToken^ value )</pre>

## ▣ Parameters

***value*** ([JToken](#))

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Int64)**

Performs an explicit conversion from [JToken](#) to [Int64](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator long (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Long
```

```
static explicit operator long long (  
    JToken^ value  
)
```

## ▣ Parameters

***value*** ([JToken](#))

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<DateTime>)**

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator Nullable<DateTime> (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Nullable(Of DateTime)
```

```
static explicit operator Nullable<DateTime> (  
    JToken^ value  
)
```

## ▣ Parameters

**value ([JToken](#))**

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<DateTimeOffset>)**

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static explicit operator <a href="#">Nullable&lt;DateTimeOff</a>     <a href="#">JToken</a> <i>value</i> )</pre>		
<pre>Public Shared Narrowing Operator CType ( _     <i>value</i> As <a href="#">JToken</a> _ ) As <a href="#">Nullable</a>(Of <a href="#">DateTimeOffset</a>)</pre>		
<pre>static explicit operator <a href="#">Nullable&lt;DateTimeOffset&gt;</a> (      <a href="#">JToken</a>^ <i>value</i> )</pre>		

## ▣ Parameters

***value*** ([JToken](#))

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Boolean)**

Performs an explicit conversion from [JToken](#) to [Boolean](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static explicit operator bool (     JToken value )</pre>		
	<pre>Public Shared Narrowing Operator CType ( _     value As JToken _ ) As Boolean</pre>	
		<pre>static explicit operator bool (     JToken^ value )</pre>

## ▣ Parameters

***value*** ([JToken](#))

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to DateTimeOffset)**

Performs an explicit conversion from [JToken](#) to [DateTimeOffset](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public static explicit operator DateTimeOffset (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As DateTimeOffset
```

```
static explicit operator DateTimeOffset (  
    JToken^ value  
)
```

## ▣ Parameters

**value ([JToken](#))**

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<Boolean>)**

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator Nullable<bool> (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Nullable(Of Boolean)
```

```
static explicit operator Nullable<bool> (  
    JToken^ value  
)
```

## ▣ Parameters

**value** ([JToken](#))

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<Decimal>)**

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator Nullable<decimal> (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Nullable(Of Decimal)
```

```
static explicit operator Nullable<Decimal> (  
    JToken^ value  
)
```

## ▣ Parameters

**value** ([JToken](#))

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to UInt16)**

Performs an explicit conversion from [JToken](#) to [UInt16](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static explicit operator ushort (     JToken value )</pre>		
	<pre>Public Shared Narrowing Operator CType ( _     value As JToken _ ) As UShort</pre>	
		<pre>static explicit operator unsigned short (     JToken^ value )</pre>

## ▣ Parameters

***value*** ([JToken](#))

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<Int32>)**

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static explicit operator Nullable&lt;int&gt; (     JToken value )</pre>		
	<pre>Public Shared Narrowing Operator CType ( _     value As JToken _ ) As Nullable(Of Integer)</pre>	
		<pre>static explicit operator Nullable&lt;int&gt; (     JToken^ value )</pre>

## ▣ Parameters

***value*** ([JToken](#))

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<Int16>)**

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

C#

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator Nullable<short> (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Nullable(Of Short)
```

```
static explicit operator Nullable<short> (  
    JToken^ value  
)
```

## ▣ Parameters

***value*** ([JToken](#))

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<Double>)**

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator Nullable<double> (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Nullable(Of Double)
```

```
static explicit operator Nullable<double> (  
    JToken^ value  
)
```

## Parameters

**value** ([JToken](#))

The value.

## Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Int32)**

Performs an explicit conversion from [JToken](#) to [Int32](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator int (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Integer
```

```
static explicit operator int (  
    JToken^ value  
)
```

## ▣ Parameters

**value ([JToken](#))**

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Int16)**

Performs an explicit conversion from [JToken](#) to [Int16](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator short (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Short
```

```
static explicit operator short (  
    JToken^ value  
)
```

## ▣ Parameters

**value ([JToken](#))**

The value.

## ▣ Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Get the first child token of this token.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual JToken First { get; }
```

```
Public Overridable ReadOnly Property First As JToken  
    Get
```

```
public:  
virtual property JToken^ First {  
    JToken^ get ();  
}
```

## ▣ Value

A [JToken](#) containing the first child token of the [JToken](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Members

Icon	Member	Description
	<a href="#">FromObject(Object)</a>	Creates a <a href="#">JToken</a> from an object.
	<a href="#">FromObject(Object, JsonSerializer)</a>	Creates a <a href="#">JToken</a> from an object using the specified <a href="#">JsonSerializer</a> .

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►**FromObject(Object)**

Creates a [JToken](#) from an object.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public static JToken FromObject(  
    Object o  
)
```

```
Public Shared Function FromObject ( _  
    o As Object _  
) As JToken
```

```
public:  
static JToken^ FromObject(  
    Object^ o  
)
```

**Parameters****o ([Object](#))**

The object that will be used to create [JToken](#).

**Return Value**

A [JToken](#) with the value of the specified object

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## FromObject Method (o, JsonSerializer)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

### FromObject(Object, JsonSerializer)

Creates a [JToken](#) from an object using the specified [JsonSerializer](#).

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JToken FromObject(  
    Object o,  
    JsonSerializer jsonSerializer  
)
```

```
Public Shared Function FromObject ( _  
    o As Object, _  
    jsonSerializer As JsonSerializer _  
) As JToken
```

```
public:  
static JToken^ FromObject(  
    Object^ o,  
    JsonSerializer^ jsonSerializer  
)
```

#### ▣ Parameters

##### o ([Object](#))

The object that will be used to create [JToken](#).

##### *jsonSerializer* ([JsonSerializer](#))

The [JsonSerializer](#) that will be used when reading the object.

#### ▣ Return Value

A [JToken](#) with the value of the specified object

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## (parameter)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

### GetMetaObject(Expression)

Returns the [DynamicMetaObject](#) responsible for binding operations performed on this object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual DynamicMetaObject GetMetaObject(  
    Expression parameter  
)
```

```
Protected Overridable Function GetMetaObject ( _  
    parameter As Expression _  
) As DynamicMetaObject
```

```
protected:  
virtual DynamicMetaObject^ GetMetaObject(  
    Expression^ parameter  
)
```

#### ▣ Parameters

##### *parameter* ([Expression](#))

The expression tree representation of the runtime value.

#### ▣ Return Value

The [DynamicMetaObject](#) to bind this object.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **HasValues**

C#

Gets a value indicating whether this token has children tokens.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract bool HasValues { get; }
```

```
Public MustOverride ReadOnly Property HasValues As B  
    Get
```

```
public:  
virtual property bool HasValues {  
    bool get () abstract;  
}
```

## Value

`true` if this token has child values; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[-] **Members**

Icon	Member	Description
	<a href="#">Implicit(Boolean to JToken)</a>	Performs an implicit conversion from <a href="#">Boolean</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(DateTimeOffset to JToken)</a>	Performs an implicit conversion from <a href="#">DateTimeOffset</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Nullable&lt;Boolean&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Int64 to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Nullable&lt;DateTime&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Nullable&lt;DateTimeOffset&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <a href="#">JToken</a> .

	<a href="#">Implicit(Nullable&lt;Decimal&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Nullable&lt;Double&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Int16 to JToken)</a>	Performs an implicit conversion from <a href="#">Int16</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(UInt16 to JToken)</a>	Performs an implicit conversion from <a href="#">UInt16</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Int32 to JToken)</a>	Performs an implicit conversion from <a href="#">Int32</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Nullable&lt;Int32&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(DateTime to JToken)</a>	Performs an implicit conversion from <a href="#">DateTime</a> to <a href="#">JToken</a> .

	<a href="#">Implicit(Nullable&lt;Int64&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Nullable&lt;Single&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Decimal to JToken)</a>	Performs an implicit conversion from <a href="#">Decimal</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Nullable&lt;Int16&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Nullable&lt;UInt16&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Nullable&lt;UInt32&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Nullable&lt;UInt64&gt; to JToken)</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <a href="#">JToken</a> .

	<a href="#">Implicit(Double to JToken)</a>	Performs an implicit conversion from <a href="#">Double</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Single to JToken)</a>	Performs an implicit conversion from <a href="#">Single</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(String to JToken)</a>	Performs an implicit conversion from <a href="#">String</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(UInt32 to JToken)</a>	Performs an implicit conversion from <a href="#">UInt32</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(UInt64 to JToken)</a>	Performs an implicit conversion from <a href="#">UInt64</a> to <a href="#">JToken</a> .
	<a href="#">Implicit(Byte[] to JToken)</a>	Performs an implicit conversion from <b>Byte[]</b> to <a href="#">JToken</a> .

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(Boolean to JToken)**

Performs an implicit conversion from [Boolean](#) to [JToken](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     bool value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As Boolean _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     bool value )</pre>

## ▣ Parameters

**value** ([Boolean](#))

The value to create a [JValue](#) from.

## ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(Byte[] to JToken)**

Performs an implicit conversion from **Byte[]** to [JToken](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     byte[] value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As Byte() _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     array&lt;unsigned char&gt;^ value )</pre>

## ▣ Parameters

**value (Byte[])**

The value to create a [JValue](#) from.

## ▣ Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(DateTime to JToken)**

Performs an implicit conversion from [DateTime](#) to [JToken](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     DateTime value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As DateTime _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     DateTime value )</pre>

## ▣ Parameters

### **value** ([DateTime](#))

The value to create a [JValue](#) from.

## ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(DateTimeOffset to JToken)

Performs an implicit conversion from [DateTimeOffset](#) to [JToken](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     DateTimeOffset value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As DateTimeOffset _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     DateTimeOffset value )</pre>

### ▣ Parameters

#### **value** ([DateTimeOffset](#))

The value to create a [JValue](#) from.

### ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(Decimal to JToken)**

Performs an implicit conversion from [Decimal](#) to [JToken](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static implicit operator JToken (  
    decimal value  
)
```

```
Public Shared Widening Operator CType ( _  
    value As Decimal _  
) As JToken
```

```
static implicit operator JToken^ (  
    Decimal value  
)
```

## ▣ Parameters

**value** ([Decimal](#))

The value to create a [JValue](#) from.

## ▣ Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(Double to JToken)**

Performs an implicit conversion from [Double](#) to [JToken](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     double value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As Double _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     double value )</pre>

## ▣ Parameters

**value** ([Double](#))

The value to create a [JValue](#) from.

## ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(Int16 to JToken)**

Performs an implicit conversion from [Int16](#) to [JToken](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static implicit operator JToken (  
    short value  
)
```

```
Public Shared Widening Operator CType ( _  
    value As Short _  
) As JToken
```

```
static implicit operator JToken^ (  
    short value  
)
```

## ▣ Parameters

### ***value*** ([Int16](#))

The value to create a [JValue](#) from.

## ▣ Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(Int32 to JToken)**

Performs an implicit conversion from [Int32](#) to [JToken](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator <a href="#">JToken</a> (     <a href="#">int</a> value )</pre>		
	<pre>Public Shared Widening Operator CType (      value As <a href="#">Integer</a>  ) As <a href="#">JToken</a></pre>	
		<pre>static implicit operator <a href="#">JToken</a>^ (     <a href="#">int</a> value )</pre>

## ▣ Parameters

### **value** ([Int32](#))

The value to create a [JValue](#) from.

## ▣ Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(Int64 to JToken)**

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

## Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     long value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As Long _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     long long value )</pre>

## Parameters

**value (Int64)**

The value to create a [JValue](#) from.

## Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(Nullable<Boolean> to JToken)

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     Nullable&lt;bool&gt; value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As Nullable(Of Boolean) _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     Nullable&lt;bool&gt; value )</pre>

### ▣ Parameters

**value** ([Nullable<Boolean>](#))

The value to create a [JValue](#) from.

### ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(Nullable<DateTime> to JToken)

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     Nullable&lt;DateTime&gt; value )</pre>		
	<pre>Public Shared Widening Operator CType (     value As Nullable(Of DateTime) _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     Nullable&lt;DateTime&gt; value )</pre>

### ▣ Parameters

**value** ([Nullable<DateTime>](#))

The value to create a [JValue](#) from.

### ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(Nullable<DateTimeOffset> to JToken)

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     Nullable&lt;DateTimeOffset&gt; value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As Nullable(Of DateTimeOffset) _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     Nullable&lt;DateTimeOffset&gt; value )</pre>

### ▣ Parameters

**value** ([Nullable<DateTimeOffset>](#))

The value to create a [JValue](#) from.

### ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(Nullable<Decimal> to JToken)

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     Nullable&lt;decimal&gt; value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As Nullable(Of Decimal) _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     Nullable&lt;Decimal&gt; value )</pre>

### ▣ Parameters

**value** ([Nullable<Decimal>](#))

The value to create a [JValue](#) from.

### ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(Nullable<Double> to JToken)

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     Nullable&lt;double&gt; value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As Nullable(Of Double) _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     Nullable&lt;double&gt; value )</pre>

### ▣ Parameters

**value (Nullable<Double>)**

The value to create a [JValue](#) from.

### ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(Nullable<Int16> to JToken)

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     Nullable&lt;short&gt; value )</pre>		
<pre>Public Shared Widening Operator CType ( _     value As Nullable(Of Short) _ ) As JToken</pre>		
<pre>static implicit operator JToken^ (     Nullable&lt;short&gt; value )</pre>		

### ▣ Parameters

**value** ([Nullable<Int16>](#))

The value to create a [JValue](#) from.

### ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(Nullable<Int32> to JToken)

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     Nullable&lt;int&gt; value )</pre>		
<pre>Public Shared Widening Operator CType ( _     value As Nullable(Of Integer) _ ) As JToken</pre>		
<pre>static implicit operator JToken^ (     Nullable&lt;int&gt; value )</pre>		

### ▣ Parameters

**value** ([Nullable<Int32>](#))

The value to create a [JValue](#) from.

### ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(Nullable<Int64> to JToken)

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     Nullable&lt;long&gt; value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As Nullable(Of Long) _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     Nullable&lt;long long&gt; value )</pre>

### ▣ Parameters

**value** ([Nullable<Int64>](#))

The value to create a [JValue](#) from.

### ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(Nullable<Single> to JToken)

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     Nullable&lt;float&gt; value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As Nullable(Of Single) _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     Nullable&lt;float&gt; value )</pre>

### ▣ Parameters

**value** ([Nullable<Single>](#))

The value to create a [JValue](#) from.

### ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(Nullable<UInt16> to JToken)

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     Nullable&lt;ushort&gt; value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As Nullable(Of UShort) _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     Nullable&lt;unsigned short&gt; value )</pre>

### ▣ Parameters

**value** ([Nullable<UInt16>](#))

The value to create a [JValue](#) from.

### ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(Nullable<UInt32> to JToken)

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     Nullable&lt;uint&gt; value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As Nullable(Of UIntInteger) _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     Nullable&lt;unsigned int&gt; value )</pre>

### ▣ Parameters

**value** ([Nullable<UInt32>](#))

The value to create a [JValue](#) from.

### ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(Nullable<UInt64> to JToken)

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     Nullable&lt;ulong&gt; value )</pre>		
<pre>Public Shared Widening Operator CType ( _     value As Nullable(Of ULong) _ ) As JToken</pre>		
<pre>static implicit operator JToken^ (     Nullable&lt;unsigned long long&gt; value )</pre>		

### ▣ Parameters

**value** ([Nullable<UInt64>](#))

The value to create a [JValue](#) from.

### ▣ Return Value

The [JValue](#) initialized with the specified value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(Single to JToken)**

Performs an implicit conversion from [Single](#) to [JToken](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static implicit operator JToken (  
    float value  
)
```

```
Public Shared Widening Operator CType ( _  
    value As Single _  
) As JToken
```

```
static implicit operator JToken^ (  
    float value  
)
```

## ▣ Parameters

**value** ([Single](#))

The value to create a [JValue](#) from.

## ▣ Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(String to JToken)**

Performs an implicit conversion from [String](#) to [JToken](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static implicit operator JToken (  
    string value  
)
```

```
Public Shared Widening Operator CType ( _  
    value As String _  
) As JToken
```

```
static implicit operator JToken^ (  
    String^ value  
)
```

## ▣ Parameters

### **value (String)**

The value to create a [JValue](#) from.

## ▣ Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(UInt16 to JToken)**

Performs an implicit conversion from [UInt16](#) to [JToken](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     ushort value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As UShort _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     unsigned short value )</pre>

## ▣ Parameters

**value (UInt16)**

The value to create a [JValue](#) from.

## ▣ Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(UInt32 to JToken)**

Performs an implicit conversion from [UInt32](#) to [JToken](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     uint value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As UInteger _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     unsigned int value )</pre>

## ▣ Parameters

**value (UInt32)**

The value to create a [JValue](#) from.

## ▣ Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(UInt64 to JToken)**

Performs an implicit conversion from [UInt64](#) to [JToken](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static implicit operator JToken (     ulong value )</pre>		
	<pre>Public Shared Widening Operator CType ( _     value As ULong _ ) As JToken</pre>	
		<pre>static implicit operator JToken^ (     unsigned long long value )</pre>

## ▣ Parameters

**value (UInt64)**

The value to create a [JValue](#) from.

## ▣ Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets the [JToken](#) with the specified key.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual JToken this[
    Object key
] { get; set; }
```

```
Public Overridable Default Property Item ( _
    key As Object _
) As JToken
    Get
    Set
```

```
public:
virtual property JToken^ default[Object^ key] {
    JToken^ get (Object^ key);
    void set (Object^ key, JToken^ value);
}
```

### ▣ Parameters

**key** ([Object](#))

### ▣ Value

The [JToken](#) with the specified key.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Last** C#

Get the last child token of this token.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual JToken Last { get; }
```

```
Public Overridable ReadOnly Property Last As JToken  
    Get
```

```
public:  
virtual property JToken^ Last {  
    JToken^ get ();  
}
```

## ▣ Value

A [JToken](#) containing the last child token of the [JToken](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Load(JsonReader)

Creates a [JToken](#) from a [JsonReader](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JToken Load(  
    JsonReader reader  
)
```

```
Public Shared Function Load ( _  
    reader As JsonReader _  
) As JToken
```

```
public:  
static JToken^ Load(  
    JsonReader^ reader  
)
```

### ▣ Parameters

#### ***reader*** ([JsonReader](#))

An [JsonReader](#) positioned at the token to read into this [JToken](#).

### ▣ Return Value

An [JToken](#) that contains the token and its descendant tokens that were read from the reader. The runtime type of the token is determined by the token type of the first token encountered in the reader.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Next** C#

Gets the next sibling token of this node.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JToken Next { get; internal set; }
```

```
Public Property Next As JToken  
    Get  
    Friend Set
```

```
public:  
property JToken^ Next {  
    JToken^ get ();  
    internal: void set (JToken^ value);  
}
```

## ▣ Value

The [JToken](#) that contains the next sibling token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Parent**

C# ▼

Gets or sets the parent.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JContainer Parent { get; internal set; }
```

```
Public Property Parent As JContainer  
    Get  
    Friend Set
```

```
public:  
property JContainer^ Parent {  
    JContainer^ get ();  
    internal: void set (JContainer^ value);  
}
```

## ▣ Value

The parent.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Parse(String)**

C#

Load a [JToken](#) from a string that contains JSON.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JToken Parse(  
    string json  
)
```

```
Public Shared Function Parse ( _  
    json As String _  
) As JToken
```

```
public:  
static JToken^ Parse(  
    String^ json  
)
```

## ▣ Parameters

**json (String)**

A [String](#) that contains JSON.

## ▣ Return Value

A [JToken](#) populated from the string that contains JSON.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Previous**

Gets the previous sibling token of this node.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JToken Previous { get; internal set; }
```

```
Public Property Previous As JToken  
    Get  
    Friend Set
```

```
public:  
property JToken^ Previous {  
    JToken^ get ();  
    internal: void set (JToken^ value);  
}
```

## ▣ Value

The [JToken](#) that contains the previous sibling token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (reader)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

### ReadFrom(JsonReader)

Creates a [JToken](#) from a [JsonReader](#).

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JToken ReadFrom(  
    JsonReader reader  
)
```

```
Public Shared Function ReadFrom ( _  
    reader As JsonReader _  
) As JToken
```

```
public:  
static JToken^ ReadFrom(  
    JsonReader^ reader  
)
```

#### ▣ Parameters

##### ***reader*** ([JsonReader](#))

An [JsonReader](#) positioned at the token to read into this [JToken](#).

#### ▣ Return Value

An [JToken](#) that contains the token and its descendant tokens that were read from the reader. The runtime type of the token is determined by the token type of the first token encountered in the reader.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Remove()**

C# ▼

Removes this token from its parent.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Remove()
```

```
Public Sub Remove
```

```
public:  
void Remove()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Replaces this token with the specified token.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public void Replace(     <a href="#">JToken</a> value )</pre>		
	<pre>Public Sub Replace ( _     value As <a href="#">JToken</a> _ )</pre>	
		<pre>public: void Replace(     <a href="#">JToken</a>^ value )</pre>

## ▣ Parameters

### **value (JToken)**

The value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Root**

C# ▼

Gets the root [JToken](#) of this [JToken](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JToken Root { get; }
```

```
Public ReadOnly Property Root As JToken  
    Get
```

```
public:  
property JToken^ Root {  
    JToken^ get ();  
}
```

## ▣ Value

The root [JToken](#) of this [JToken](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

C#

## Members

Icon	Member	Description
	<a href="#">SelectToken(String)</a>	Selects the token that matches the object path.
	<a href="#">SelectToken(String, Boolean)</a>	Selects the token that matches the object path.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (path)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

### SelectToken(String)

Selects the token that matches the object path.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JToken SelectToken(  
    string path  
)
```

```
Public Function SelectToken ( _  
    path As String _  
) As JToken
```

```
public:  
JToken^ SelectToken(  
    String^ path  
)
```

#### ▣ Parameters

##### *path* ([String](#))

The object path from the current [JToken](#) to the [JToken](#) to be returned. This must be a string of property names or array indexes separated by periods, such as

C#

 Cop

```
Tables[0].DefaultView[0].Price
```

in C# or

C#

 Cop

```
Tables(0).DefaultView(0).Price
```

in Visual Basic.

### ▣ **Return Value**

The [JToken](#) that matches the object path or a null reference if no matching token is found.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## SelectToken Method (path, errorWhenNoMatch)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

### SelectToken(String, Boolean)

Selects the token that matches the object path.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JToken SelectToken(  
    string path,  
    bool errorWhenNoMatch  
)
```

```
Public Function SelectToken ( _  
    path As String, _  
    errorWhenNoMatch As Boolean _  
) As JToken
```

```
public:  
JToken^ SelectToken(  
    String^ path,  
    bool errorWhenNoMatch  
)
```

#### ▣ Parameters

##### *path* ([String](#))

The object path from the current [JToken](#) to the [JToken](#) to be returned. This must be a string of property names or array indexes separated by periods, such as

C#

 Cop

```
Tables[0].DefaultView[0].Price
```

in C# or

**C#**

 Cop

```
Tables(0).DefaultView(0).Price
```

in Visual Basic.

***errorWhenNoMatch*** ([Boolean](#))

A flag to indicate whether an error should be thrown if no token is found.

▣ **Return Value**

The [JToken](#) that matches the object path.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **ToObject()**

C# ▼

## Members

Icon	Member	Description
	<a href="#">ToObject&lt;T&gt;()</a>	Creates the specified .NET type from the <a href="#">JToken</a> .
	<a href="#">ToObject&lt;T&gt;(JsonSerializer)</a>	Creates the specified .NET type from the <a href="#">JToken</a> using the specified <a href="#">JsonSerializer</a> .

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **ToObject<T>()**

Creates the specified .NET type from the [JToken](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public T ToObject<T>()
```

```
Public Function ToObject(Of T) As T
```

```
public:  
generic<typename T>  
T ToObject()
```

## ▣ Generic Template Parameters

*T*

**[Missing <typeparam name="T"/> documentation for "M:Newtonsoft.Json.Linq.JToken.ToObject`1"]**

## ▣ Return Value

The new object created from the JSON value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## ToObject<T> Method (JsonSerializer)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► [ToObject<T>](#)  
([JsonSerializer](#))

Creates the specified .NET type from the [JToken](#) using the specified [JsonSerializer](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public T ToObject<T>(
    JsonSerializer jsonSerializer
)
```

```
Public Function ToObject(Of T) ( _
    jsonSerializer As JsonSerializer _
) As T
```

```
public:
generic<typename T>
T ToObject(
    JsonSerializer^ jsonSerializer
)
```

### Generic Template Parameters

**T**

[Missing <typeparam name="T"/> documentation for "M:Newtonsoft.Json.Linq.JToken.ToObject`1(Newtonsoft.Json")

### Parameters

**jsonSerializer** ([JsonSerializer](#))

The [JsonSerializer](#) that will be used when creating the object.

## ▣ Return Value

The new object created from the JSON value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Members

Icon	Member	Description
	<a href="#">ToString()</a>	Returns the indented JSON for this token.  (Overrides <a href="#">Object.ToString()</a> .)
	<a href="#">ToString(Formatting, JsonSerializer[])</a>	Returns the JSON for this token using the given formatting and converters.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Tostring()**

C#

Returns the indented JSON for this token.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

```
public:  
virtual String^ ToString() override
```

## ▣ Return Value

The indented JSON for this token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## ToString Method (formatting, converters)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

### ToString(Formatting, JsonSerializer[])

Returns the JSON for this token using the given formatting and converters.

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public string ToString(     Formatting formatting,     params JsonSerializer[] converters )</pre>	<pre>Public Function ToString ( _     formatting As Formatting, _     ParamArray converters As JsonSerializer() _ ) As String</pre>	

#### Parameters

##### *formatting* ([Formatting](#))

Indicates how the output is formatted.

##### *converters* ([JsonSerializer\[\]](#))

A collection of [JsonSerializer](#) which will be used when writing the token.

#### Return Value

The JSON for this token using the given formatting and converters.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Type** C#

Gets the node type for this [JToken](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract JTokenType Type { get; }
```

```
Public MustOverride ReadOnly Property Type As JToken  
Get
```

```
public:  
virtual property JTokenType Type {  
    JTokenType get () abstract;  
}
```

## ▣ Value

The type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



Gets the [JToken](#) with the specified key converted to the specified type. C# ▼

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual T Value<T>(
    Object key
)
```

```
Public Overridable Function Value(Of T) ( _
    key As Object _
) As T
```

```
public:
generic<typename T>
virtual T Value(
    Object^ key
)
```

### ▣ Generic Template Parameters

***T***

The type to convert the token to.

### ▣ Parameters

**key (Object)**

The token key.

### ▣ Return Value

The converted token value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Values<T>()**

C#

Returns a collection of the child values of this token, in document order.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual IEnumerable<T> Values<T>()
```

```
Public Overridable Function Values(Of T) As IEnumerable
```

```
public:  
generic<typename T>  
virtual IEnumerable<T>^ Values()
```

### ▣ Generic Template Parameters

***T***

The type to convert the values to.

### ▣ Return Value

A [IEnumerable<T>](#) containing the child values of this [JToken](#), in document order.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## WriteTo Method (writer, converters)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

### WriteTo(JsonWriter, JsonConverter[])

Writes this token to a [JsonWriter](#).

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract void WriteTo(  
    JsonWriter writer,  
    params JsonConverter[] converters  
)
```

```
Public MustOverride Sub WriteTo ( _  
    writer As JsonWriter, _  
    ParamArray converters As JsonConverter() _  
)
```

```
public:  
virtual void WriteTo(  
    JsonWriter^ writer,  
    ... array<JsonConverter^>^ converters  
) abstract
```

#### ▣ Parameters

##### **writer** ([JsonWriter](#))

A [JsonWriter](#) into which this method will write.

##### **converters** ([JsonConverter](#)[])

A collection of [JsonConverter](#) which will be used when writing the token.

(4.5.6.14930)

# JTokenEqualityComparer Class

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► **JTokenEqualityComparer**

Compares tokens to determine whether they are equal.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JTokenEqualityComparer : IEqualityCompa
```

```
Public Class JTokenEqualityComparer _  
    Implements IEqualityComparer(Of JToken)
```

```
public ref class JTokenEqualityComparer : IEqualityC
```

## Members

All Members

Constructors

Methods

Public

Protected

Instance

Static

Declare

Inherite

Icon	Member	Description
	<a href="#">JTokenEqualityComparer()</a>	Initializes a new instance of the <b>JTokenEqualityComparer</b> class
	<a href="#">Equals(JToken, JToken)</a>	Determines whether the specified objects are equal.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>

		(Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode(JToken)</a>	Returns a hash code for the specific object.
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JTokenEqualityComparer](#)



# JTokenEqualityComparer

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenEqualityComparer](#) ► **JTokenEqualityComparer()**

Initializes a new instance of the [JTokenEqualityComparer](#) class

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JTokenEqualityComparer()
```

```
Public Sub New
```

```
public:  
JTokenEqualityComparer()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenEqualityComparer](#) ►

## Equals(JToken, JToken)

Determines whether the specified objects are equal.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Equals(  
    JToken x,  
    JToken y  
)
```

```
Public Function Equals ( _  
    x As JToken, _  
    y As JToken _  
) As Boolean
```

```
public:  
virtual bool Equals(  
    JToken^ x,  
    JToken^ y  
) sealed
```

### ▣ Parameters

#### **x (JToken)**

The first object of type [JToken](#) to compare.

#### **y (JToken)**

The second object of type [JToken](#) to compare.

### ▣ Return Value

true if the specified objects are equal; otherwise, false.

(4.5.6.14930)

**(obj)**[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenEqualityComparer](#) ►**GetHashCode(JToken)**

Returns a hash code for the specified object.

**Declaration Syntax**

C#	Visual Basic	Visual C++
----	--------------	------------

```
public int GetHashCode(  
    JToken obj  
)
```

```
Public Function GetHashCode ( _  
    obj As JToken _  
) As Integer
```

```
public:  
virtual int GetHashCode(  
    JToken^ obj  
) sealed
```

**Parameters*****obj* (JToken)**

The [Object](#) for which a hash code is to be returned.

**Return Value**

A hash code for the specified object.

**Exceptions**

Exception	Condition
<a href="#">ArgumentNullException</a>	The type of <i>obj</i> is a reference type and <i>obj</i> is null.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

# JTokenReader Class

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenReader](#) C#

Represents a reader that provides fast, non-cached, forward-only access to serialized Json data.

## Declaration Syntax

C#      Visual Basic      Visual C++

```
public class JTokenReader : JsonReader,
    IJsonLineInfo
```

```
Public Class JTokenReader _
    Inherits JsonReader _
    Implements IJsonLineInfo
```

```
public ref class JTokenReader : public JsonReader,
    IJsonLineInfo
```

## Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JTokenReader(JToken)</a>	Initializes a new instance of the <b>JTokenReader</b> class.
	<a href="#">Close()</a>	Changes the <a href="#">JsonReader.State</a> to Closed.  (Inherited from <a href="#">JsonReader</a> .)

	<a href="#">CloseInput</a>	<p>Gets or sets a value indicating whether the underlying stream or <a href="#">TextReader</a> should be closed when the reader is closed.</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
	<a href="#">Culture</a>	<p>Gets or sets the culture used when reading JSON. Defaults to <a href="#">InvariantCulture</a>.</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
	<a href="#">CurrentState</a>	<p>Gets the current reader state.</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
	<a href="#">DateParseHandling</a>	<p>Get or set how date formatted strings: e.g. "VDate(1198908717056)V" and "2012-03-21T05:40Z", are parsed when reading JSON.</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
	<a href="#">DateTimeZoneHandling</a>	<p>Get or set how <a href="#">DateTime</a> time zones are handling when reading JSON.</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
	<a href="#">Depth</a>	<p>Gets the depth of the current token in the JSON document.</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
	<a href="#">Dispose(Boolean)</a>	<p>Releases unmanaged and - optional managed resources</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
	<a href="#">Equals(Object)</a>	<p>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</p>

		(Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MaxDepth</a>	Gets or sets the maximum depth allowed when reading JSON. Reading past this depth will throw a <a href="#">JsonReaderException</a> .  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Path</a>	Gets the path of the current JSON token.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">QuoteChar</a>	Gets the quotation mark character used to enclose the value of a string.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">Read()</a>	Reads the next JSON token from the stream.

		(Overrides <a href="#">JsonReader.Read().</a> )
	<a href="#">ReadAsBytes()</a>	<p>Reads the next JSON token from the stream as a <b>Byte[]</b>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsBytes().</a>)</p>
	<a href="#">ReadAsDateTime()</a>	<p>Reads the next JSON token from the stream as a <a href="#">Nullable&lt;T&gt;</a>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsDateTime().</a>)</p>
	<a href="#">ReadAsDateTimeOffset()</a>	<p>Reads the next JSON token from the stream as a <a href="#">Nullable&lt;T&gt;</a>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsDateTimeOffset</a></p>
	<a href="#">ReadAsDecimal()</a>	<p>Reads the next JSON token from the stream as a <a href="#">Nullable&lt;T&gt;</a>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsDecimal().</a>)</p>
	<a href="#">ReadAsInt32()</a>	<p>Reads the next JSON token from the stream as a <a href="#">Nullable&lt;T&gt;</a>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsInt32().</a>)</p>
	<a href="#">ReadAsString()</a>	<p>Reads the next JSON token from the stream as a <a href="#">String</a>.</p> <p>(Overrides <a href="#">JsonReader.ReadAsString().</a>)</p>
	<a href="#">SetStateBasedOnCurrent()</a>	<p>Sets the state based on current token type.</p> <p>(Inherited from <a href="#">JsonReader.</a>)</p>

	<a href="#">SetToken(JsonToken)</a>	Sets the current token.  (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">SetToken(JsonToken, Object)</a>	Sets the current token and value.  (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">Skip()</a>	Skips the children of the current token.  (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">TokenType</a>	Gets the type of the current JSON token.  (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">Value</a>	Gets the text value of the current JSON token.  (Inherited from <a href="#">JsonReader.</a> )
	<a href="#">ValueType</a>	Gets The Common Language Runtime (CLR) type for the current JSON token.  (Inherited from <a href="#">JsonReader.</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonReader](#)

└─ **JTokenReader**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor (token)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenReader](#) ►

### JTokenReader(JToken)

Initializes a new instance of the [JTokenReader](#) class.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JTokenReader(  
    JToken token  
)
```

```
Public Sub New ( _  
    token As JToken _  
)
```

```
public:  
JTokenReader(  
    JToken^ token  
)
```

#### ▣ Parameters

##### ***token*** ([JToken](#))

The token to read from.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenReader](#) ► **Read()**

C#

Reads the next JSON token from the stream.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override bool Read()
```

```
Public Overrides Function Read As Boolean
```

```
public:  
virtual bool Read() override
```

## ▣ Return Value

true if the next token was read successfully; false if there are no more tokens to read.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenReader](#) ►

## ReadAsBytes()

Reads the next JSON token from the stream as a **Byte[]**.

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override byte[] ReadAsBytes()
```

```
Public Overrides Function ReadAsBytes As Byte()
```

```
public:  
virtual array<unsigned char>^ ReadAsBytes() override
```

### ▣ Return Value

A **Byte[]** or a null reference if the next JSON token is null. This method will return **null** at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenReader](#) ►

## ReadAsDateTime()

Reads the next JSON token from the stream as a [Nullable<T>](#).

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<DateTime> ReadAsDateTime()
```

```
Public Overrides Function ReadAsDateTime As Nullable
```

```
public:  
virtual Nullable<DateTime> ReadAsDateTime() override
```

### ▣ Return Value

A [String](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenReader](#) ►

### ReadAsDateTimeOffset()

Reads the next JSON token from the stream as a [Nullable<T>](#).

C#

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<DateTimeOffset> ReadAsDateT
```

```
Public Overrides Function ReadAsDateTimeOffset As Nu
```

```
public:  
virtual Nullable<DateTimeOffset> ReadAsDateTimeOffse
```

#### Return Value

A [Nullable<T>](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenReader](#) ►

## ReadAsDecimal()

Reads the next JSON token from the stream as a [Nullable<T>](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<decimal> ReadAsDecimal()
```

```
Public Overrides Function ReadAsDecimal As Nullable(
```

```
public:  
virtual Nullable<Decimal> ReadAsDecimal() override
```

### ▣ Return Value

A [Nullable<T>](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenReader](#) ►

## ReadAsInt32()

Reads the next JSON token from the stream as a [Nullable<T>](#).

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<int> ReadAsInt32()
```

```
Public Overrides Function ReadAsInt32 As Nullable\(Of
```

```
public:  
virtual Nullable<int> ReadAsInt32() override
```

### ▣ Return Value

A [Nullable<T>](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenReader](#) ►

## ReadAsString()

Reads the next JSON token from the stream as a [String](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override string ReadAsString()
```

```
Public Overrides Function ReadAsString As String
```

```
public:  
virtual String^ ReadAsString() override
```

### ▣ Return Value

A [String](#). This method will return `null` at the end of an array.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Specifies the type of token.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum JTokenType
```

```
Public Enumeration JTokenType
```

```
public enum class JTokenType
```

## ▣ Members

Member	Description
<b>None</b>	No token type has been set.
<b>Object</b>	A JSON object.
<b>Array</b>	A JSON array.
<b>Constructor</b>	A JSON constructor.
<b>Property</b>	A JSON object property.
<b>Comment</b>	A comment.
<b>Integer</b>	An integer value.
<b>Float</b>	A float value.
<b>String</b>	A string value.
<b>Boolean</b>	A boolean value.
<b>Null</b>	A null value.
<b>Undefined</b>	An undefined value.
<b>Date</b>	A date value.

<b>Raw</b>	A raw JSON value.
<b>Bytes</b>	A collection of bytes value.
<b>Guid</b>	A Guid value.
<b>Uri</b>	A Uri value.
<b>TimeSpan</b>	A TimeSpan value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# JTokenWriter Class

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) C#

Represents a writer that provides a fast, non-cached, forward-only way generating Json data.

## Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class JTokenWriter : JsonWriter
```

```
Public Class JTokenWriter _
    Inherits JsonWriter
```

```
public ref class JTokenWriter : public JsonWriter
```

## Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
			<input checked="" type="checkbox"/> Declare
			<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JTokenWriter(JContainer)</a>	Initializes a new instance <b>JTokenWriter</b> class with <a href="#">JContainer</a> .
	<a href="#">JTokenWriter()</a>	Initializes a new instance <b>JTokenWriter</b> class.
	<a href="#">Close()</a>	Closes this stream and the stream.

		(Overrides <a href="#">JsonWriter.Cl</a>
	<a href="#">CloseOutput</a>	Gets or sets a value indicating whether the underlying stream or stream writer should be closed when the <a href="#">JsonWriter</a> is closed.  (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">DateFormatHandling</a>	Get or set how dates are formatted in JSON text.  (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">DateTimeZoneHandling</a>	Get or set how <a href="#">DateTime</a> values are handled when writing JSON text.  (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">Equals(Object)</a>	Determines whether the current <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is garbage collected.  (Inherited from <a href="#">Object</a> .)
	<a href="#">Flush()</a>	Flushes whatever is in the buffer to the underlying streams and the underlying stream.  (Overrides <a href="#">JsonWriter.Fli</a>
	<a href="#">Formatting</a>	Indicates how JSON text is formatted.  (Inherited from <a href="#">JsonWriter</a> )
		

	<a href="#">GetHashCode()</a>	Serves as a hash function type. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">Path</a>	Gets the path of the writer. (Inherited from <a href="#">JsonWriter</a> .)
	<a href="#">Token</a>	Gets the token being written.
	<a href="#">Top</a>	Gets the top. (Inherited from <a href="#">JsonWriter</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">WriteComment(String)</a>	Writes out a comment. <b>C#</b> <pre>/* ... */</pre> containing the specified text. (Overrides <a href="#">JsonWriter.WriteComment</a> .)
	<a href="#">WriteEnd(JsonToken)</a>	Writes the end.

		(Overrides <a href="#">JsonWriter.WriteEnd(JsonWriter)</a> )
	<a href="#">WriteEnd()</a>	Writes the end of the current object or array.  (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteEndArray()</a>	Writes the end of an array.  (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteEndConstructor()</a>	Writes the end of a constructor.  (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteEndObject()</a>	Writes the end of a JSON object.  (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteIndent()</a>	Writes indent characters.  (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteIndentSpace()</a>	Writes an indent space.  (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteNull()</a>	Writes a null value.  (Overrides <a href="#">JsonWriter.WriteNull()</a> )
	<a href="#">WritePropertyName(String)</a>	Writes the property name and value pair on a JSON object.  (Overrides <a href="#">JsonWriter.WritePropertyName(String)</a> )
	<a href="#">WriteRaw(String)</a>	Writes raw JSON.  (Overrides <a href="#">JsonWriter.WriteRaw(String)</a> )
	<a href="#">WriteRawValue(String)</a>	Writes raw JSON where expected and updates the current state.

		(Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteStartArray()</a>	Writes the beginning of a  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteStartConstructor(String)</a>	Writes the start of a cons given name.  (Overrides <a href="#">JsonWriter.WriteStartCon</a>
	<a href="#">WriteStartObject()</a>	Writes the beginning of a  (Overrides <a href="#">JsonWriter.WriteStartObj</a>
	<a href="#">WriteState</a>	Gets the state of the writ  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteToken(JsonReader)</a>	Writes the current <a href="#">JsonR</a>  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteUndefined()</a>	Writes an undefined valu  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(String)</a>	Writes a <a href="#">String</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(S</a>
	<a href="#">WriteValue(Int32)</a>	Writes a <a href="#">Int32</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(UInt32)</a>	Writes a <a href="#">UInt32</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(U</a>
		

	<a href="#">WriteValue(Int64)</a>	Writes a <a href="#">Int64</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(UInt64)</a>	Writes a <a href="#">UInt64</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(U</a>
	<a href="#">WriteValue(Single)</a>	Writes a <a href="#">Single</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(S</a>
	<a href="#">WriteValue(Double)</a>	Writes a <a href="#">Double</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(Boolean)</a>	Writes a <a href="#">Boolean</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(B</a>
	<a href="#">WriteValue(Int16)</a>	Writes a <a href="#">Int16</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(UInt16)</a>	Writes a <a href="#">UInt16</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(U</a>
	<a href="#">WriteValue(Char)</a>	Writes a <a href="#">Char</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(Byte)</a>	Writes a <a href="#">Byte</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(SByte)</a>	Writes a <a href="#">SByte</a> value.  (Overrides

		<a href="#">JsonWriter.WriteValue(S</a>
	<a href="#">WriteValue(Decimal)</a>	Writes a <a href="#">Decimal</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(DateTime)</a>	Writes a <a href="#">DateTime</a> value  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(DateTimeOffset)</a>	Writes a <a href="#">DateTimeOffset</a>  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(Byte[])</a>	Writes a <b>Byte[]</b> value.  (Overrides <a href="#">JsonWriter.WriteValue(B</a>
	<a href="#">WriteValue(TimeSpan)</a>	Writes a <a href="#">TimeSpan</a> value  (Overrides <a href="#">JsonWriter.WriteValue(Ti</a>
	<a href="#">WriteValue(Guid)</a>	Writes a <a href="#">Guid</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(Uri)</a>	Writes a <a href="#">Uri</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(Nullable&lt;Int32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;UInt32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Int64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val

		(Inherited from <a href="#">JsonWrite</a> )
	<a href="#">WriteValue(Nullable&lt;UInt64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a> )
	<a href="#">WriteValue(Nullable&lt;Single&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a> )
	<a href="#">WriteValue(Nullable&lt;Double&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a> )
	<a href="#">WriteValue(Nullable&lt;Boolean&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a> )
	<a href="#">WriteValue(Nullable&lt;Int16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a> )
	<a href="#">WriteValue(Nullable&lt;UInt16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a> )
	<a href="#">WriteValue(Nullable&lt;Char&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a> )
	<a href="#">WriteValue(Nullable&lt;Byte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a> )
	<a href="#">WriteValue(Nullable&lt;SByte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a> )
	<a href="#">WriteValue(Nullable&lt;Decimal&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a> )
	<a href="#">WriteValue(Nullable&lt;DateTime&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val

		(Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;DateTimeOffset&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;Guid&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Nullable&lt;TimeSpan&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValue(Object)</a>	Writes a <a href="#">Object</a> value. An if the value cannot be wr JSON token. (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteValueDelimiter()</a>	Writes the JSON value d (Inherited from <a href="#">JsonWrite</a>
	<a href="#">WriteWhitespace(String)</a>	Writes out the given whit (Inherited from <a href="#">JsonWrite</a>

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonWriter](#)

└─ **JTokenWriter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

**JTokenWriter()**

C#

### Members

Icon	Member	Description
	<a href="#">JTokenWriter(JContainer)</a>	Initializes a new instance of the <a href="#">JTokenWriter</a> class writing to the given <a href="#">JContainer</a> .
	<a href="#">JTokenWriter()</a>	Initializes a new instance of the <a href="#">JTokenWriter</a> class.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### JTokenWriter()

Initializes a new instance of the [JTokenWriter](#) class.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JTokenWriter()
```

```
Public Sub New
```

```
public:  
JTokenWriter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Constructor (container)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### JTokenWriter(JContainer)

Initializes a new instance of the [JTokenWriter](#) class writing to the given [JContainer](#).

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JTokenWriter(     <a href="#">JContainer</a> container )</pre>		
	<pre>Public Sub New ( _     container As <a href="#">JContainer</a> _ )</pre>	
		<pre>public: JTokenWriter(     <a href="#">JContainer</a>^ container )</pre>

#### Parameters

##### **container** ([JContainer](#))

The container being written to.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **Close()**

C#

Closes this stream and the underlying stream.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void Close()
```

```
Public Overrides Sub Close
```

```
public:  
virtual void Close() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **Flush()**

C#

Flushes whatever is in the buffer to the underlying streams and also flushes the underlying stream.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void Flush()
```

```
Public Overrides Sub Flush
```

```
public:  
virtual void Flush() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets the token being written.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JToken Token { get; }
```

```
Public ReadOnly Property Token As JToken  
    Get
```

```
public:  
property JToken^ Token {  
    JToken^ get ();  
}
```

## ▣ Value

The token being written.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# WriteComment Method (text)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

## WriteComment(String)

Writes out a comment

C#

C#

 Cop

```
/* ... */
```

containing the specified text.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteComment(  
    string text  
)
```

```
Public Overrides Sub WriteComment ( _  
    text As String _  
)
```

```
public:  
virtual void WriteComment(  
    String^ text  
) override
```

### ▣ Parameters

**text** ([String](#))

Text to place inside the comment.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

## WriteEnd(JsonToken)

Writes the end.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected override void WriteEnd(  
    JsonToken token  
)
```

```
Protected Overrides Sub WriteEnd ( _  
    token As JsonToken _  
)
```

```
protected:  
virtual void WriteEnd(  
    JsonToken token  
) override
```

### ▣ Parameters

***token*** ([JsonToken](#))

The token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **WriteNull()**

C#

Writes a null value.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteNull()
```

```
Public Overrides Sub WriteNull
```

```
public:  
virtual void WriteNull() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Method (name)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WritePropertyName(String)

Writes the property name of a name/value pair on a Json object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WritePropertyName(  
    string name  
)
```

```
Public Overrides Sub WritePropertyName ( _  
    name As String _  
)
```

```
public:  
virtual void WritePropertyName(  
    String^ name  
) override
```

#### ▣ Parameters

**name** ([String](#))

The name of the property.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

## WriteRaw(String)

Writes raw JSON.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteRaw(  
    string json  
)
```

```
Public Overrides Sub WriteRaw ( _  
    json As String _  
)
```

```
public:  
virtual void WriteRaw(  
    String^ json  
) override
```

### ▣ Parameters

*json* (**String**)

The raw JSON to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

## WriteStartArray()

Writes the beginning of a Json array.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteStartArray()
```

```
Public Overrides Sub WriteStartArray
```

```
public:  
virtual void WriteStartArray() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**Method (name)**[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►**WriteStartConstructor(String)**

Writes the start of a constructor with the given name.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteStartConstructor(  
    string name  
)
```

```
Public Overrides Sub WriteStartConstructor ( _  
    name As String _  
)
```

```
public:  
virtual void WriteStartConstructor(  
    String^ name  
) override
```

**Parameters*****name* ([String](#))**

The name of the constructor.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

## WriteStartObject()

Writes the beginning of a Json object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteStartObject()
```

```
Public Overrides Sub WriteStartObject
```

```
public:  
virtual void WriteStartObject() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►**WriteUndefined()**

Writes an undefined value.

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteUndefined()
```

```
Public Overrides Sub WriteUndefined
```

```
public:  
virtual void WriteUndefined() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# WriteValue Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **WriteValue()**

C# ▼

## Members

Icon	Member	Description
	<a href="#">WriteValue(String)</a>	Writes a <a href="#">String</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(S</a>
	<a href="#">WriteValue(Int32)</a>	Writes a <a href="#">Int32</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(In</a>
	<a href="#">WriteValue(UInt32)</a>	Writes a <a href="#">UInt32</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(U</a>
	<a href="#">WriteValue(Int64)</a>	Writes a <a href="#">Int64</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(In</a>
	<a href="#">WriteValue(UInt64)</a>	Writes a <a href="#">UInt64</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(U</a>
	<a href="#">WriteValue(Single)</a>	Writes a <a href="#">Single</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(S</a>
	<a href="#">WriteValue(Double)</a>	Writes a <a href="#">Double</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(Boolean)</a>	Writes a <a href="#">Boolean</a> value.

		(Overrides <a href="#">JsonWriter.WriteValue(B</a>
	<a href="#">WriteValue(Int16)</a>	Writes a <a href="#">Int16</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(In</a>
	<a href="#">WriteValue(UInt16)</a>	Writes a <a href="#">UInt16</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(U</a>
	<a href="#">WriteValue(Char)</a>	Writes a <a href="#">Char</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(C</a>
	<a href="#">WriteValue(Byte)</a>	Writes a <a href="#">Byte</a> value.  (Overrides <a href="#">JsonWriter.W</a>
	<a href="#">WriteValue(SByte)</a>	Writes a <a href="#">SByte</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(S</a>
	<a href="#">WriteValue(Decimal)</a>	Writes a <a href="#">Decimal</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(DateTime)</a>	Writes a <a href="#">DateTime</a> value  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
	<a href="#">WriteValue(DateTimeOffset)</a>	Writes a <a href="#">DateTimeOffset</a>  (Overrides <a href="#">JsonWriter.WriteValue(D</a>
		

	<a href="#">WriteValue(Byte[])</a>	Writes a <b>Byte[]</b> value.  (Overrides <a href="#">JsonWriter.WriteValue(B</a>
≡	<a href="#">WriteValue(TimeSpan)</a>	Writes a <a href="#">TimeSpan</a> value  (Overrides <a href="#">JsonWriter.WriteValue(Ti</a>
≡	<a href="#">WriteValue(Guid)</a>	Writes a <a href="#">Guid</a> value.  (Overrides <a href="#">JsonWriter.WriteValue(G</a>
≡	<a href="#">WriteValue(Uri)</a>	Writes a <a href="#">Uri</a> value.  (Overrides <a href="#">JsonWriter.W</a>
≡	<a href="#">WriteValue(Nullable&lt;Int32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">WriteValue(Nullable&lt;UInt32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">WriteValue(Nullable&lt;Int64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">WriteValue(Nullable&lt;UInt64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">WriteValue(Nullable&lt;Single&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">WriteValue(Nullable&lt;Double&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">WriteValue(Nullable&lt;Boolean&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val

	<a href="#">WriteValue(Nullable&lt;Int16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;UInt16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Char&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Byte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;SByte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Decimal&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;DateTime&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;DateTimeOffset&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;Guid&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Nullable&lt;TimeSpan&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
	<a href="#">WriteValue(Object)</a>	Writes a <a href="#">Object</a> value. An exception is raised if the value cannot be written.

single JSON token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(Boolean)

Writes a [Boolean](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    bool value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Boolean _  
)
```

```
public:  
virtual void WriteValue(  
    bool value  
) override
```

#### ▣ Parameters

**value** ([Boolean](#))

The [Boolean](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(Byte)

Writes a [Byte](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    byte value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Byte _  
)
```

```
public:  
virtual void WriteValue(  
    unsigned char value  
) override
```

#### ▣ Parameters

##### **value** ([Byte](#))

The [Byte](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►**WriteValue(Byte[])**Writes a **Byte[]** value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    byte[] value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Byte() _  
)
```

```
public:  
virtual void WriteValue(  
    array<unsigned char>^ value  
) override
```

▣ **Parameters****value** ([Byte\[\]](#))The **Byte[]** value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(Char)

Writes a [Char](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    char value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Char _  
)
```

```
public:  
virtual void WriteValue(  
    wchar\_t value  
) override
```

#### ▣ Parameters

##### **value** ([Char](#))

The [Char](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(DateTime)

Writes a [DateTime](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    DateTime value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As DateTime _  
)
```

```
public:  
virtual void WriteValue(  
    DateTime value  
) override
```

#### ▣ Parameters

##### **value** ([DateTime](#))

The [DateTime](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(DateTimeOffset)

C#

Writes a [DateTimeOffset](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    DateTimeOffset value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As DateTimeOffset _  
)
```

```
public:  
virtual void WriteValue(  
    DateTimeOffset value  
) override
```

#### ▣ Parameters

**value** ([DateTimeOffset](#))

The [DateTimeOffset](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►**WriteValue(Decimal)**

Writes a [Decimal](#) value.

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    decimal value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Decimal _  
)
```

```
public:  
virtual void WriteValue(  
    Decimal value  
) override
```

**▣ Parameters****value ([Decimal](#))**

The [Decimal](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(Double)

Writes a [Double](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    double value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Double _  
)
```

```
public:  
virtual void WriteValue(  
    double value  
) override
```

#### ▣ Parameters

##### **value** ([Double](#))

The [Double](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►**WriteValue(Guid)**Writes a [Guid](#) value.▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    Guid value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Guid _  
)
```

```
public:  
virtual void WriteValue(  
    Guid value  
) override
```

▣ **Parameters****value ([Guid](#))**The [Guid](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(Int16)

Writes a [Int16](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    short value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Short _  
)
```

```
public:  
virtual void WriteValue(  
    short value  
) override
```

#### ▣ Parameters

##### **value** ([Int16](#))

The [Int16](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(Int32)

C#

Writes a [Int32](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    int value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Integer _  
)
```

```
public:  
virtual void WriteValue(  
    int value  
) override
```

#### ▣ Parameters

##### **value** ([Int32](#))

The [Int32](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►**WriteValue(Int64)**

Writes a [Int64](#) value.

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    long value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Long _  
)
```

```
public:  
virtual void WriteValue(  
    long long value  
) override
```

**▣ Parameters****value ([Int64](#))**

The [Int64](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(SByte)

Writes a [SByte](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    sbyte value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As SByte _  
)
```

```
public:  
virtual void WriteValue(  
    signed char value  
) override
```

#### ▣ Parameters

##### **value (SByte)**

The [SByte](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(Single)

Writes a [Single](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    float value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Single _  
)
```

```
public:  
virtual void WriteValue(  
    float value  
) override
```

#### ▣ Parameters

**value** ([Single](#))

The [Single](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(String)

Writes a [String](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    string value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As String _  
)
```

```
public:  
virtual void WriteValue(  
    String^ value  
) override
```

#### ▣ Parameters

##### **value** ([String](#))

The [String](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(TimeSpan)

Writes a [TimeSpan](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    TimeSpan value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As TimeSpan _  
)
```

```
public:  
virtual void WriteValue(  
    TimeSpan value  
) override
```

#### ▣ Parameters

##### **value** ([TimeSpan](#))

The [TimeSpan](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(UInt16)

C#

Writes a [UInt16](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    ushort value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As UShort _  
)
```

```
public:  
virtual void WriteValue(  
    unsigned short value  
) override
```

#### ▣ Parameters

##### **value** ([UInt16](#))

The [UInt16](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(UInt32)

Writes a [UInt32](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    uint value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As UInteger _  
)
```

```
public:  
virtual void WriteValue(  
    unsigned int value  
) override
```

#### ▣ Parameters

##### **value** ([UInt32](#))

The [UInt32](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(UInt64)

Writes a [UInt64](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    ulong value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As ULong _  
)
```

```
public:  
virtual void WriteValue(  
    unsigned long long value  
) override
```

#### ▣ Parameters

##### **value** ([UInt64](#))

The [UInt64](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WriteValue(Uri)

Writes a [Uri](#) value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteValue(  
    Uri value  
)
```

```
Public Overrides Sub WriteValue ( _  
    value As Uri _  
)
```

```
public:  
virtual void WriteValue(  
    Uri^ value  
) override
```

#### ▣ Parameters

##### **value ([Uri](#))**

The [Uri](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JValue Class

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► **JValue**

Represents a value in JSON (string, integer, date, etc).

## Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public class JValue : JToken, IEquatable&lt;JValue&gt;,     IFormattable, IComparable, IComparable&lt;JValue&gt;</pre>		

```
Public Class JValue _
    Inherits JToken _
    Implements IEquatable(Of JValue), IFormattable,
    IComparable, IComparable(Of JValue)
```

```
public ref class JValue : public JToken,
    IEquatable<JValue^>, IFormattable, IComparable
```

## Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declare <input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JValue(JValue)</a>	Initializes a new instance of the <b>JValue</b> class from another <b>JValue</b> object.
	<a href="#">JValue(Int64)</a>	Initializes a new instance of the <b>JValue</b> class with the given value.

	<a href="#">JValue(UInt64)</a>	Initializes a new instance of the <b>JValue</b> class with the given value.
	<a href="#">JValue(Double)</a>	Initializes a new instance of the <b>JValue</b> class with the given value.
	<a href="#">JValue(DateTime)</a>	Initializes a new instance of the <b>JValue</b> class with the given value.
	<a href="#">JValue(Boolean)</a>	Initializes a new instance of the <b>JValue</b> class with the given value.
	<a href="#">JValue(String)</a>	Initializes a new instance of the <b>JValue</b> class with the given value.
	<a href="#">JValue(Guid)</a>	Initializes a new instance of the <b>JValue</b> class with the given value.
	<a href="#">JValue(Uri)</a>	Initializes a new instance of the <b>JValue</b> class with the given value.
	<a href="#">JValue(TimeSpan)</a>	Initializes a new instance of the <b>JValue</b> class with the given value.
	<a href="#">JValue(Object)</a>	Initializes a new instance of the

		<b>JValue</b> class with the given value.
	<a href="#">AddAfterSelf(Object)</a>	Adds the specified content immediately after this token.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">AddBeforeSelf(Object)</a>	Adds the specified content immediately before this token.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">AfterSelf()</a>	Returns a collection of the sibling tokens after this token, in document order.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Ancestors()</a>	Returns a collection of the ancestor tokens of this token.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">BeforeSelf()</a>	Returns a collection of the sibling tokens before this token, in document order.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Children()</a>	Returns a collection of the child tokens of this token, in document order.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">CompareTo(JValue)</a>	Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the

		same position in the sort order as the other object.
	<a href="#">CreateComment(String)</a>	Creates a <b>JValue</b> comment with the given value.
	<a href="#">CreateReader()</a>	Creates an <a href="#">JsonReader</a> for this token.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">CreateString(String)</a>	Creates a <b>JValue</b> string with the given value.
	<a href="#">DeepClone()</a>	Creates a new instance of the <a href="#">JToken</a> . All child tokens are recursively cloned.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Equals(JValue)</a>	Indicates whether the current object is equal to another object of the same type.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Overrides <a href="#">Object.Equals(Object)</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.

		(Inherited from <a href="#">Object</a> .)
	<a href="#">First</a>	Get the first child token of this token.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Overrides <a href="#">Object.GetHashCode()</a> .)
	<a href="#">GetMetaObject(Expression)</a>	Returns the <a href="#">DynamicMetaObject</a> responsible for binding operations performed on this object.  (Overrides <a href="#">JToken.GetMetaObject(Expression)</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">HasValues</a>	Gets a value indicating whether this token has children tokens.  (Overrides <a href="#">JToken.HasValues</a> .)
	<a href="#">IsValid(JToken, JsonSchema)</a>	Determines whether the <a href="#">JToken</a> is valid.  (Inherited from <a href="#">Extensions</a> .)
	<a href="#">IsValid(JToken, JsonSchema, IList&lt;String&gt;)</a>	Determines whether the <a href="#">JToken</a> is valid.  (Inherited from <a href="#">Extensions</a> .)
	<a href="#">Item[Object]</a>	Gets the <a href="#">JToken</a> with the specified key.  (Inherited from <a href="#">JToken</a> .)
		

	<a href="#">Last</a>	Get the last child token of this token  (Inherited from <a href="#">JToken</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Next</a>	Gets the next sibling token of this node.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Parent</a>	Gets or sets the parent.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Previous</a>	Gets the previous sibling token of this node.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Remove()</a>	Removes this token from its parent  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Replace(JToken)</a>	Replaces this token with the specified token.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Root</a>	Gets the root <a href="#">JToken</a> of this <a href="#">JToken</a>  (Inherited from <a href="#">JToken</a> .)
	<a href="#">SelectToken(String)</a>	Selects the token that matches the object path.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">SelectToken(String, Boolean)</a>	Selects the token that matches the object path.

		(Inherited from <a href="#">JToken</a> .)
	<a href="#">ToObject&lt;T&gt;()</a>	Creates the specified .NET type from the <a href="#">JToken</a> .  (Inherited from <a href="#">JToken</a> .)
	<a href="#">ToObject&lt;T&gt;(JsonSerializer)</a>	Creates the specified .NET type from the <a href="#">JToken</a> using the specified <a href="#">JsonSerializer</a> .  (Inherited from <a href="#">JToken</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents this instance.  (Overrides <a href="#">JToken.ToString()</a> .)
	<a href="#">ToString(String)</a>	Returns a <a href="#">String</a> that represents this instance.
	<a href="#">ToString(IFormatProvider)</a>	Returns a <a href="#">String</a> that represents this instance.
	<a href="#">ToString(String, IFormatProvider)</a>	Returns a <a href="#">String</a> that represents this instance.
	<a href="#">ToString(Formatting, JsonConverters[])</a>	Returns the JSON for this token using the given formatting and converters.  (Inherited from <a href="#">JToken</a> .)
	<a href="#">Type</a>	Gets the node type for this <a href="#">JToken</a> .  (Overrides <a href="#">JToken.Type</a> .)
		

	<a href="#">Validate(JToken, JsonSchema)</a>	Validates the specified <a href="#">JToken</a> . (Inherited from <a href="#">Extensions</a> .)
	<a href="#">Validate(JToken, JsonSchema, ValidationEventHandler)</a>	Validates the specified <a href="#">JToken</a> . (Inherited from <a href="#">Extensions</a> .)
	<a href="#">Value</a>	Gets or sets the underlying token value.
	<a href="#">Value&lt;T&gt;(Object)</a>	Gets the <a href="#">JToken</a> with the specified key converted to the specified type (Inherited from <a href="#">JToken</a> .)
	<a href="#">Values&lt;T&gt;()</a>	Returns a collection of the child values of this token, in document order. (Inherited from <a href="#">JToken</a> .)
	<a href="#">WriteTo(JsonWriter, JsonConverter[])</a>	Writes this token to a <a href="#">JsonWriter</a> . (Overrides <a href="#">JToken.WriteTo(JsonWriter, JsonConverter[])</a> .)

## Inheritance Hierarchy

[Object](#)

└─ [JToken](#)

└─ **JValue**

└─ [JRaw](#)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(\)](#)

C#

## Members

Icon	Member	Description
	<a href="#">JValue(JValue)</a>	Initializes a new instance of the <a href="#">JValue</a> class from another <a href="#">JValue</a> object.
	<a href="#">JValue(Int64)</a>	Initializes a new instance of the <a href="#">JValue</a> class with the given value.
	<a href="#">JValue(UInt64)</a>	Initializes a new instance of the <a href="#">JValue</a> class with the given value.
	<a href="#">JValue(Double)</a>	Initializes a new instance of the <a href="#">JValue</a> class with the given value.
	<a href="#">JValue(DateTime)</a>	Initializes a new instance of the <a href="#">JValue</a> class with the given value.
	<a href="#">JValue(Boolean)</a>	Initializes a new instance of the <a href="#">JValue</a> class with the given value.
	<a href="#">JValue(String)</a>	Initializes a new instance of the <a href="#">JValue</a> class with the given value.

 <a href="#">JValue(Guid)</a>	Initializes a new instance of the <a href="#">JValue</a> class with the given value.
 <a href="#">JValue(Uri)</a>	Initializes a new instance of the <a href="#">JValue</a> class with the given value.
 <a href="#">JValue(TimeSpan)</a>	Initializes a new instance of the <a href="#">JValue</a> class with the given value.
 <a href="#">JValue(Object)</a>	Initializes a new instance of the <a href="#">JValue</a> class with the given value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# JValue Constructor

## (other)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(JValue\)](#)

Initializes a new instance of the [JValue](#) class from another [JValue](#) object

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JValue(  
    JValue other  
)
```

```
Public Sub New (   
    other As JValue   
)
```

```
public:  
JValue(  
    JValue^ other  
)
```

### ▣ Parameters

#### ***other*** ([JValue](#))

A [JValue](#) object to copy from.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JValue Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► **JValue(Boolean)**

Initializes a new instance of the [JValue](#) class with the given value.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JValue(  
    bool value  
)
```

```
Public Sub New ( _  
    value As Boolean _  
)
```

```
public:  
JValue(  
    bool value  
)
```

## Parameters

**value** ([Boolean](#))

The value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JValue Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(DateTime\)](#)

Initializes a new instance of the [JValue](#) class with the given value.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JValue(  
    DateTime value  
)
```

```
Public Sub New ( _  
    value As DateTime _  
)
```

```
public:  
JValue(  
    DateTime value  
)
```

## Parameters

**value** ([DateTime](#))

The value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JValue Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(Double\)](#)

Initializes a new instance of the [JValue](#) class with the given value.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JValue(  
    double value  
)
```

```
Public Sub New ( _  
    value As Double _  
)
```

```
public:  
JValue(  
    double value  
)
```

## Parameters

**value (Double)**

The value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JValue Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(Guid\)](#)

Initializes a new instance of the [JValue](#) class with the given value.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JValue(  
    Guid value  
)
```

```
Public Sub New ( _  
    value As Guid _  
)
```

```
public:  
JValue(  
    Guid value  
)
```

## Parameters

**value ([Guid](#))**

The value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# JValue Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(Int64\)](#)

Initializes a new instance of the [JValue](#) class with the given value.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JValue(  
    long value  
)
```

```
Public Sub New ( _  
    value As Long _  
)
```

```
public:  
JValue(  
    long long value  
)
```

## Parameters

**value (Int64)**

The value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JValue Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(Object\)](#)

Initializes a new instance of the [JValue](#) class with the given value.

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JValue(  
    Object value  
)
```

```
Public Sub New ( _  
    value As Object _  
)
```

```
public:  
JValue(  
    Object^ value  
)
```

## Parameters

**value (Object)**

The value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JValue Constructor

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(String\)](#)

Initializes a new instance of the [JValue](#) class with the given value.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JValue(  
    string value  
)
```

```
Public Sub New (   
    value As String   
)
```

```
public:  
JValue(  
    String^ value  
)
```

### Parameters

#### **value (String)**

The value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JValue Constructor

## (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(TimeSpan\)](#)

Initializes a new instance of the [JValue](#) class with the given value.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JValue(  
    TimeSpan value  
)
```

```
Public Sub New ( _  
    value As TimeSpan _  
)
```

```
public:  
JValue(  
    TimeSpan value  
)
```

### Parameters

**value** ([TimeSpan](#))

The value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JValue Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(UInt64\)](#)

Initializes a new instance of the [JValue](#) class with the given value.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JValue(  
    ulong value  
)
```

```
Public Sub New (   
    value As ULong   
)
```

```
public:  
JValue(  
    unsigned long long value  
)
```

## Parameters

**value** ([UInt64](#))

The value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JValue Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(Uri\)](#)

Initializes a new instance of the [JValue](#) class with the given value.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JValue(  
    Uri value  
)
```

```
Public Sub New ( _  
    value As Uri _  
)
```

```
public:  
JValue(  
    Uri^ value  
)
```

## Parameters

**value (Uri)**

The value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ►

## CompareTo(JValue)

C#

Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public int CompareTo(  
    JValue obj  
)
```

```
Public Function CompareTo ( _  
    obj As JValue _  
) As Integer
```

```
public:  
virtual int CompareTo(  
    JValue^ obj  
) sealed
```

### Parameters

**obj (JValue)**

An object to compare with this instance.

### Return Value

A 32-bit signed integer that indicates the relative order of the objects being compared. The return value has these meanings: Value Meaning  
Less than zero This instance is less than *obj*.  
Zero This instance is equal to *obj*.  
Greater than zero This instance is greater than *obj*.

## ▣ Exceptions

Exception	Condition
<a href="#">ArgumentException</a>	<i>obj</i> is not the same type as this instance.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ►**CreateComment(String)**

Creates a [JValue](#) comment with the given value.

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public static JValue CreateComment(  
    string value  
)
```

```
Public Shared Function CreateComment ( _  
    value As String _  
) As JValue
```

```
public:  
static JValue^ CreateComment(  
    String^ value  
)
```

**▣ Parameters****value ([String](#))**

The value.

**▣ Return Value**

A [JValue](#) comment with the given value.

**(value)**[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ►**CreateString(String)**

Creates a [JValue](#) string with the given value.

**▣ Declaration Syntax**

C#

Visual Basic

Visual C++

```
public static JValue CreateString(  
    string value  
)
```

```
Public Shared Function CreateString ( _  
    value As String _  
) As JValue
```

```
public:  
static JValue^ CreateString(  
    String^ value  
)
```

**▣ Parameters****value ([String](#))**

The value.

**▣ Return Value**

A [JValue](#) string with the given value.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► **Equals()**

C# ▼

## Members

Icon	Member	Description
	<a href="#">Equals(JValue)</a>	Indicates whether the current object is equal to another object of the same type.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Overrides <a href="#">Object.Equals(Object)</a> .)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► **Equals(JValue)**

C#

Indicates whether the current object is equal to another object of the same type.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Equals(  
    JValue other  
)
```

```
Public Function Equals ( _  
    other As JValue _  
) As Boolean
```

```
public:  
virtual bool Equals(  
    JValue^ other  
) sealed
```

## ▣ Parameters

### *other* ([JValue](#))

An object to compare with this object.

## ▣ Return Value

true if the current object is equal to the *other* parameter; otherwise, false

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

C#

Determines whether the specified [Object](#) is equal to the current [Object](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public override bool Equals(  
    Object obj  
)
```

```
Public Overrides Function Equals ( _  
    obj As Object _  
) As Boolean
```

```
public:  
virtual bool Equals(  
    Object^ obj  
) override
```

## Parameters

### *obj* ([Object](#))

The [Object](#) to compare with the current [Object](#).

## Return Value

true if the specified [Object](#) is equal to the current [Object](#); otherwise, false

## Exceptions

Exception	Condition
<a href="#">NullReferenceException</a>	The <i>obj</i> parameter is null.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► **GetHashCode()**

C#

Serves as a hash function for a particular type.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override int GetHashCode()
```

```
Public Overrides Function GetHashCode As Integer
```

```
public:  
virtual int GetHashCode() override
```

## ▣ Return Value

A hash code for the current [Object](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (parameter)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ►

### GetMetaObject(Expression)

Returns the [DynamicMetaObject](#) responsible for binding operations performed on this object.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>protected override <a href="#">DynamicMetaObject</a> GetMetaObject(     <a href="#">Expression</a> <i>parameter</i> )</pre>		
	<pre>Protected Overrides Function GetMetaObject (      <i>parameter</i> As <a href="#">Expression</a>  ) As <a href="#">DynamicMetaObject</a></pre>	
		<pre>protected: virtual <a href="#">DynamicMetaObject</a>^ GetMetaObject(     <a href="#">Expression</a>^ <i>parameter</i> ) override</pre>

#### ▣ Parameters

##### *parameter* ([Expression](#))

The expression tree representation of the runtime value.

#### ▣ Return Value

The [DynamicMetaObject](#) to bind this object.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► **HasValues**

C#

Gets a value indicating whether this token has children tokens.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override bool HasValues { get; }
```

```
Public Overrides ReadOnly Property HasValues As Bool  
    Get
```

```
public:  
virtual property bool HasValues {  
    bool get () override;  
}
```

## ▣ Value

`true` if this token has child values; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

C# ▼

## Members

Icon	Member	Description
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents this instance.  (Overrides <a href="#">JToken.ToString()</a> .)
	<a href="#">ToString(String)</a>	Returns a <a href="#">String</a> that represents this instance.
	<a href="#">ToString(IFormatProvider)</a>	Returns a <a href="#">String</a> that represents this instance.
	<a href="#">ToString(String, IFormatProvider)</a>	Returns a <a href="#">String</a> that represents this instance.
	<a href="#">ToString(Formatting, JsonConverter[])</a>	Returns the JSON for this token using the given formatting and converters.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► **ToString()**

C#

Returns a [String](#) that represents this instance.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

```
public:  
virtual String^ ToString() override
```

## ▣ Return Value

A [String](#) that represents this instance.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## ToString Method (formatProvider)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ►

### ToString(IFormatProvider)

C#

Returns a [String](#) that represents this instance.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string ToString(  
    IFormatProvider formatProvider  
)
```

```
Public Function ToString ( _  
    formatProvider As IFormatProvider _  
) As String
```

```
public:  
String^ ToString(  
    IFormatProvider^ formatProvider  
)
```

#### ▣ Parameters

***formatProvider*** ([IFormatProvider](#))

The format provider.

#### ▣ Return Value

A [String](#) that represents this instance.

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► **Tostring(String)**

C#

Returns a [String](#) that represents this instance.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string ToString(  
    string format  
)
```

```
Public Function ToString ( _  
    format As String _  
) As String
```

```
public:  
String^ ToString(  
    String^ format  
)
```

## ▣ Parameters

***format*** ([String](#))

The format.

## ▣ Return Value

A [String](#) that represents this instance.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Tostring Method (format, formatProvider)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [ToString\(String, IFormatProvider\)](#)

Returns a [String](#) that represents this instance.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public string ToString(     string format,     IFormatProvider formatProvider )</pre>		
	<pre>Public Function ToString ( _     format As String, _     formatProvider As IFormatProvider _ ) As String</pre>	
		<pre>public: virtual String^ ToString(     String^ format,     IFormatProvider^ formatProvider ) sealed</pre>

### ▣ Parameters

**format (String)**

The format.

**formatProvider (IFormatProvider)**

The format provider.

### ▣ Return Value

A [String](#) that represents this instance.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► **Type** C#

Gets the node type for this [JToken](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override JTokenType Type { get; }
```

```
Public Overrides ReadOnly Property Type As JTokenType  
    Get
```

```
public:  
virtual property JTokenType Type {  
    JTokenType get () override;  
}
```

## ▣ Value

The type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets the underlying token value.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">Object</a> Value { <a href="#">get</a>; <a href="#">set</a>; }</pre>		
<pre>Public Property Value As <a href="#">Object</a>     Get     Set</pre>		
<pre>public: property <a href="#">Object</a>^ Value {     <a href="#">Object</a>^ <a href="#">get</a> ();     void <a href="#">set</a> (<a href="#">Object</a>^ <i>value</i>); }</pre>		

## ▣ Value

The underlying token value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## WriteTo Method (writer, converters)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ►

### WriteTo(JsonWriter, JsonConverter[])

C#

Writes this token to a [JsonWriter](#).

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override void WriteTo(  
    JsonWriter writer,  
    params JsonConverter[] converters  
)
```

```
Public Overrides Sub WriteTo ( _  
    writer As JsonWriter, _  
    ParamArray converters As JsonConverter() _  
)
```

```
public:  
virtual void WriteTo(  
    JsonWriter^ writer,  
    ... array<JsonConverter^>^ converters  
) override
```

#### ▣ Parameters

##### **writer** ([JsonWriter](#))

A [JsonWriter](#) into which this method will write.

##### **converters** ([JsonConverter](#)[])

A collection of [JsonConverter](#) which will be used when writing the token.

(4.5.6.14930)

## Namespace

[Namespaces](#) ► **Newtonsoft.Json.Schema**

C#

The **Newtonsoft.Json.Schema** namespace provides classes that are used to implement JSON schema.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<code>namespace Newtonsoft.Json.Schema</code>		

```
Namespace Newtonsoft.Json.Schema
```

```
namespace Newtonsoft.Json.Schema
```

### ▣ Types

All Types	Classes	Enumerations	Delegates
Icon	Type	Description	
	<a href="#">Extensions</a>	Contains the JSON schema extension methods.	
	<a href="#">JsonSchema</a>	An in-memory representation of JSON Schema.	
	<a href="#">JsonSchemaException</a>	Returns detailed information about the schema exception.	
	<a href="#">JsonSchemaGenerator</a>	Generates a <a href="#">JsonSchema</a> from a specified <a href="#">Type</a> .	

	<a href="#"><u>JsonSchemaResolver</u></a>	Resolves <a href="#"><u>JsonSchema</u></a> from an id.
	<a href="#"><u>JsonSchemaType</u></a>	The value types allowed by the <a href="#"><u>JsonSchema</u></a> .
	<a href="#"><u>UndefinedSchemaldHandling</u></a>	Specifies undefined schema Id handling options for the <a href="#"><u>JsonSchemaGenerator</u></a> .
	<a href="#"><u>ValidationEventArgs</u></a>	Returns detailed information related to the <a href="#"><u>ValidationEventHandler</u></a> .
	<a href="#"><u>ValidationEventHandler</u></a>	Represents the callback method that will handle JSON schema validation events and the <a href="#"><u>ValidationEventArgs</u></a> .

---

Contains the JSON schema extension methods.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static class Extensions
```

```
<ExtensionAttribute> _  
Public NotInheritable Class Extensions
```

```
[ExtensionAttribute]  
public ref class Extensions abstract sealed
```

## ▣ Members

All Members

Methods

Public

Protected

Instance

Static

Declared

Inherited

Icon	Member	Description
	<a href="#">IsValid(JToken, JsonSchema)</a>	Determines whether the <a href="#">JToken</a> is valid.
	<a href="#">IsValid(JToken, JsonSchema, IList&lt;String&gt;)</a>	Determines whether the <a href="#">JToken</a> is valid.
	<a href="#">Validate(JToken, JsonSchema)</a>	Validates the specified <a href="#">JToken</a> .
	<a href="#">Validate(JToken,</a>	Validates the specified <a href="#">JToken</a> .

[JsonSchema.  
ValidationEventHandler\)](#)

## [-] Inheritance Hierarchy

[Object](#)

└─ **Extensions**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

▣ **Members**

Icon	Member	Description
	<a href="#">IsValid(JToken, JsonSchema)</a>	Determines whether the <a href="#">JToken</a> is valid.
	<a href="#">IsValid(JToken, JsonSchema, IList&lt;String&gt;)</a>	Determines whether the <a href="#">JToken</a> is valid.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# Json.NET - Quick Starts & API Documentation **IsValid Method (source, schema)**

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [Extensions](#) ►

## IsValid(JToken, JsonSchema)

C#

Determines whether the [JToken](#) is valid.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static bool IsValid(     this JToken source,     JsonSchema schema )</pre>		

```
<ExtensionAttribute> _  
Public Shared Function IsValid ( _  
    source As JToken, _  
    schema As JsonSchema _  
) As Boolean
```

```
[ExtensionAttribute]  
public:  
static bool IsValid(  
    JToken^ source,  
    JsonSchema^ schema  
)
```

### ▣ Parameters

**source (JToken)**  
The source [JToken](#) to test.

**schema (JsonSchema)**  
The schema to test with.

### ▣ Return Value

`true` if the specified [JToken](#) is valid; otherwise, `false`.

#### ▣ Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [JToken](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Json.NET - Quick Starts & API Documentation **IsValid Method (source, schema, errorMessages)**

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [Extensions](#) ►

### IsValid(JToken, JsonSchema, IList<String>)

C#

Determines whether the [JToken](#) is valid.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static bool IsValid(     this JToken source,     JsonSchema schema,     out IList&lt;string&gt; errorMessages )</pre>	<pre>&lt;ExtensionAttribute&gt; _ Public Shared Function IsValid ( _     source As JToken, _     schema As JsonSchema, _     &lt;OutAttribute&gt; ByRef errorMessages As IList( ) As Boolean</pre>	<pre>[ExtensionAttribute] public: static bool IsValid(     JToken^ source,     JsonSchema^ schema,     [OutAttribute] IList&lt;String&gt;^% errorMessage )</pre>

#### ▣ Parameters

##### **source (JToken)**

The source [JToken](#) to test.

##### **schema (JsonSchema)**

The schema to test with.

***errorMessagees*** ([IList<String>](#))

When this method returns, contains any error messages generated while validating.

▣ **Return Value**

`true` if the specified [JToken](#) is valid; otherwise, `false`.

▣ **Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type [JToken](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [Extensions](#) ► **Validate()**

C# ▼

## Members

Icon	Member	Description
	<a href="#">Validate(JToken, JsonSchema)</a>	Validates the specified <a href="#">JToken</a> .
	<a href="#">Validate(JToken, JsonSchema, ValidationEventHandler)</a>	Validates the specified <a href="#">JToken</a> .

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Validate Method (source, schema)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [Extensions](#) ►

### Validate(JToken, JsonSchema)

Validates the specified [JToken](#).

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static void Validate(  
    this JToken source,  
    JsonSchema schema  
)
```

```
<ExtensionAttribute> _  
Public Shared Sub Validate ( _  
    source As JToken, _  
    schema As JsonSchema _  
)
```

```
[ExtensionAttribute]  
public:  
static void Validate(  
    JToken^ source,  
    JsonSchema^ schema  
)
```

#### ▣ Parameters

##### **source** ([JToken](#))

The source [JToken](#) to test.

##### **schema** ([JsonSchema](#))

The schema to test with.

#### ▣ Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [JToken](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Validate Method (source, schema, validationEventHandler)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [Extensions](#) ►

**Validate(JToken, JsonSchema, ValidationEventHandler)**

C#

Validates the specified [JToken](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static void Validate(  
    this JToken source,  
    JsonSchema schema,  
    ValidationEventHandler validationEventHandle  
)
```

```
<ExtensionAttribute> _  
Public Shared Sub Validate ( _  
    source As JToken, _  
    schema As JsonSchema, _  
    validationEventHandler As ValidationEventHan  
)
```

```
[ExtensionAttribute]  
public:  
static void Validate(  
    JToken^ source,  
    JsonSchema^ schema,  
    ValidationEventHandler^ validationEventHandl  
)
```

### ▣ Parameters

#### **source (JToken)**

The source [JToken](#) to test.

#### **schema (JsonSchema)**

The schema to test with.

***validationEventHandler*** ([ValidationEventHandler](#))

The validation event handler.

▣ **Usage Note**

In Visual Basic and C#, you can call this method as an instance method on any object of type [JToken](#). When you use instance method syntax to call this method, omit the first parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



An in-memory representation of a JSON Schema.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonSchema
```

```
Public Class JsonSchema
```

```
public ref class JsonSchema
```

## ▣ Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonSchema()</a>	Initializes a new instance of the <b>JsonSchema</b> class.
	<a href="#">AdditionalProperties</a>	Gets or sets the <b>JsonSchema</b> of additional properties.
	<a href="#">AllowAdditionalProperties</a>	Gets or sets a value indicating whether additional properties are allowed.

	<a href="#">Default</a>	Gets or sets the default value.
	<a href="#">Description</a>	Gets or sets the description of the object.
	<a href="#">Disallow</a>	Gets or sets disallowed types.
	<a href="#">DivisibleBy</a>	Gets or sets a number that the value should be divisible by.
	<a href="#">Enum</a>	Gets or sets the a collection of valid enum values allowed.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> (Inherited from <a href="#">Object</a> .)
	<a href="#">ExclusiveMaximum</a>	Gets or sets a flag indicating whether the value can not equal the number defined by the "maximum" attribute.
	<a href="#">ExclusiveMinimum</a>	Gets or sets a flag indicating whether the value can not equal the number defined by the "minimum" attribute.
	<a href="#">Extends</a>	Gets or sets the extend <b>JsonSchema</b> .

	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">Format</a>	Gets or sets the format.
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">Hidden</a>	Gets or sets whether the object is visible to users.
	<a href="#">Id</a>	Gets or sets the id.
	<a href="#">Identity</a>	Gets or sets the identity.
	<a href="#">Items</a>	Gets or sets the <b>JsonSchema</b> of items.
	<a href="#">Maximum</a>	Gets or sets the maximum.

	<a href="#">MaximumItems</a>	Gets or sets the maximum number of items.
	<a href="#">MaximumLength</a>	Gets or sets the maximum length.
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Minimum</a>	Gets or sets the minimum.
	<a href="#">MinimumItems</a>	Gets or sets the minimum number of items.
	<a href="#">MinimumLength</a>	Gets or sets the minimum length.
	<a href="#">Options</a>	Gets or sets a collection of options.
	<a href="#">Parse(String)</a>	Load a <b>JsonSchema</b> from a string that contains schema JSON.
	<a href="#">Parse(String, JsonSchemaResolver)</a>	Parses the specified json.
	<a href="#">Pattern</a>	Gets or sets the pattern.

	<a href="#">PatternProperties</a>	Gets or sets the pattern properties.
	<a href="#">Properties</a>	Gets or sets the <b>JsonSchema</b> of properties.
	<a href="#">Read(JsonReader)</a>	Reads a <b>JsonSchema</b> from the specified <a href="#">JsonReader</a> .
	<a href="#">Read(JsonReader, JsonSchemaResolver)</a>	Reads a <b>JsonSchema</b> from the specified <a href="#">JsonReader</a> .
	<a href="#">ReadOnly</a>	Gets or sets whether the object is read only.
	<a href="#">Required</a>	Gets or sets whether the object is required.
	<a href="#">Requires</a>	Gets or sets the required property if this property is present.
	<a href="#">Title</a>	Gets or sets the title.
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Overrides <a href="#">Object.ToString()</a> .)

	<a href="#">Transient</a>	Gets or sets whether the object is transient.
	<a href="#">Type</a>	Gets or sets the types of values allowed by the object.
	<a href="#">WriteTo(JsonWriter)</a>	Writes this schema to a <a href="#">JsonWriter</a> .
	<a href="#">WriteTo(JsonWriter, JsonSchemaResolver)</a>	Writes this schema to a <a href="#">JsonWriter</a> using the specified <a href="#">JsonSchemaResolver</a> .

## ▣ Inheritance Hierarchy

[Object](#)

└─ **JsonSchema**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## JsonSchema()

Initializes a new instance of the [JsonSchema](#) class.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchema()
```

```
Public Sub New
```

```
public:  
JsonSchema()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# AdditionalProperties

## Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

### AdditionalProperties

C#

Gets or sets the [JsonSchema](#) of additional properties.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchema AdditionalProperties { get; set; }
```

```
Public Property AdditionalProperties As JsonSchema  
    Get  
    Set
```

```
public:  
property JsonSchema^ AdditionalProperties {  
    JsonSchema^ get ();  
    void set (JsonSchema^ value);  
}
```

#### Value

The [JsonSchema](#) of additional properties.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## AllowAdditionalProperties

C#

Gets or sets a value indicating whether additional properties are allowed.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool AllowAdditionalProperties { get; set; }
```

```
Public Property AllowAdditionalProperties As Boolean  
    Get  
    Set
```

```
public:  
property bool AllowAdditionalProperties {  
    bool get ();  
    void set (bool value);  
}
```

### Value

`true` if additional properties are allowed; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets the default value.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">JToken</a> Default { get; set; }</pre>		
	<pre>Public Property Default As <a href="#">JToken</a>     Get     Set</pre>	
		<pre>public: property <a href="#">JToken</a>^ Default {     <a href="#">JToken</a>^ get ();     void set (<a href="#">JToken</a>^ <i>value</i>); }</pre>

## ▣ Value

The default value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## Description

Gets or sets the description of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Description { get; set; }
```

```
Public Property Description As String  
    Get  
    Set
```

```
public:  
property String^ Description {  
    String^ get ();  
    void set (String^ value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets disallowed types.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<JsonSchemaType> Disallow { get; set;
```

```
Public Property Disallow As Nullable(Of JsonSchemaTy  
    Get  
    Set
```

```
public:  
property Nullable<JsonSchemaType> Disallow {  
    Nullable<JsonSchemaType> get ();  
    void set (Nullable<JsonSchemaType> value);  
}
```

## ▣ Value

The disallow types.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## DivisibleBy

Gets or sets a number that the value should be divisible by.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<double> DivisibleBy { get; set; }
```

```
Public Property DivisibleBy As Nullable(Of Double)  
    Get  
    Set
```

```
public:  
property Nullable<double> DivisibleBy {  
    Nullable<double> get ();  
    void set (Nullable<double> value);  
}
```

### Value

A number that the value should be divisible by.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets the a collection of valid enum values allowed.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IList<JToken> Enum { get; set; }
```

```
Public Property Enum As IList(Of JToken)  
    Get  
    Set
```

```
public:  
property IList<JToken^>^ Enum {  
    IList<JToken^>^ get ();  
    void set (IList<JToken^>^ value);  
}
```

## ▣ Value

A collection of valid enum values allowed.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

### ExclusiveMaximum

Gets or sets a flag indicating whether the value can not equal the number defined by the "maximum" attribute.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<bool> ExclusiveMaximum { get; set; }
```

```
Public Property ExclusiveMaximum As Nullable(Of Bool)  
    Get  
    Set
```

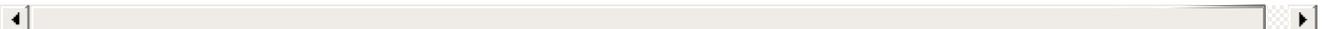
```
public:  
property Nullable<bool> ExclusiveMaximum {  
    Nullable<bool> get ();  
    void set (Nullable<bool> value);  
}
```

#### Value

A flag indicating whether the value can not equal the number defined by the "maximum" attribute.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

### ExclusiveMinimum

Gets or sets a flag indicating whether the value can not equal the number defined by the "minimum" attribute.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<bool> ExclusiveMinimum { get; set; }
```

```
Public Property ExclusiveMinimum As Nullable(Of Bool)  
    Get  
    Set
```

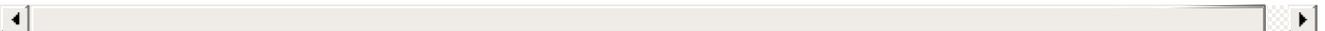
```
public:  
property Nullable<bool> ExclusiveMinimum {  
    Nullable<bool> get ();  
    void set (Nullable<bool> value);  
}
```

#### Value

A flag indicating whether the value can not equal the number defined by the "minimum" attribute.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



Gets or sets the extend [JsonSchema](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">JsonSchema</a> Extends { get; set; }</pre>		
<pre>Public Property Extends As <a href="#">JsonSchema</a>     Get     Set</pre>		
<pre>public: property <a href="#">JsonSchema</a>^ Extends {     <a href="#">JsonSchema</a>^ get ();     void set (<a href="#">JsonSchema</a>^ <i>value</i>); }</pre>		

## ▣ Value

The extended [JsonSchema](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets the format.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public string Format { get; set; }</pre>		
	<pre>Public Property Format As String     Get     Set</pre>	
		<pre>public: property String^ Format {     String^ get ();     void set (String^ value); }</pre>

## ▣ Value

The format.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets whether the object is visible to users.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">Nullable</a>&lt;<a href="#">bool</a>&gt; Hidden { <a href="#">get</a>; <a href="#">set</a>; }</pre>		
<pre>Public Property Hidden As <a href="#">Nullable</a>(Of <a href="#">Boolean</a>)     Get     Set</pre>		
<pre>public: property <a href="#">Nullable</a>&lt;<a href="#">bool</a>&gt; Hidden {     <a href="#">Nullable</a>&lt;<a href="#">bool</a>&gt; <a href="#">get</a> ();     void <a href="#">set</a> (<a href="#">Nullable</a>&lt;<a href="#">bool</a>&gt; <i>value</i>); }</pre>		

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets the id.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public string Id { get; set; }
```

```
Public Property Id As String
    Get
    Set
```

```
public:
property String^ Id {
    String^ get ();
    void set (String^ value);
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets the identity.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">IList</a>&lt;<a href="#">string</a>&gt; Identity { <a href="#">get</a>; <a href="#">set</a>; }</pre>		
	<pre>Public Property Identity As <a href="#">IList</a>(Of <a href="#">String</a>)     Get     Set</pre>	
		<pre>public: property <a href="#">IList</a>&lt;<a href="#">String</a>&gt;^ Identity {     <a href="#">IList</a>&lt;<a href="#">String</a>&gt;^ <a href="#">get</a> ();     void <a href="#">set</a> (<a href="#">IList</a>&lt;<a href="#">String</a>&gt;^ <i>value</i>); }</pre>

## ▣ Value

The identity.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets the [JsonSchema](#) of items.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IList<JsonSchema> Items { get; set; }
```

```
Public Property Items As IList(Of JsonSchema)  
    Get  
    Set
```

```
public:  
property IList<JsonSchema> Items {  
    IList<JsonSchema> get ();  
    void set (IList<JsonSchema> value);  
}
```

### ▣ Value

The [JsonSchema](#) of items.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## Maximum

Gets or sets the maximum.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<double> Maximum { get; set; }
```

```
Public Property Maximum As Nullable(Of Double)  
    Get  
    Set
```

```
public:  
property Nullable<double> Maximum {  
    Nullable<double> get ();  
    void set (Nullable<double> value);  
}
```

### ▣ Value

The maximum.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## MaximumItems

Gets or sets the maximum number of items.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<int> MaximumItems { get; set; }
```

```
Public Property MaximumItems As Nullable(Of Integer)  
    Get  
    Set
```

```
public:  
property Nullable<int> MaximumItems {  
    Nullable<int> get ();  
    void set (Nullable<int> value);  
}
```

### ▣ Value

The maximum number of items.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

### MaxLength

Gets or sets the maximum length.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<int> MaxLength { get; set; }
```

```
Public Property MaxLength As Nullable(Of Integer)  
    Get  
    Set
```

```
public:  
property Nullable<int> MaxLength {  
    Nullable<int> get ();  
    void set (Nullable<int> value);  
}
```

#### Value

The maximum length.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## Minimum

Gets or sets the minimum.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<double> Minimum { get; set; }
```

```
Public Property Minimum As Nullable(Of Double)  
    Get  
    Set
```

```
public:  
property Nullable<double> Minimum {  
    Nullable<double> get ();  
    void set (Nullable<double> value);  
}
```

### ▣ Value

The minimum.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## MinimumItems

C#

Gets or sets the minimum number of items.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<int> MinimumItems { get; set; }
```

```
Public Property MinimumItems As Nullable(Of Integer)  
    Get  
    Set
```

```
public:  
property Nullable<int> MinimumItems {  
    Nullable<int> get ();  
    void set (Nullable<int> value);  
}
```

### Value

The minimum number of items.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## MinimumLength

Gets or sets the minimum length.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<int> MinimumLength { get; set; }
```

```
Public Property MinimumLength As Nullable(Of Integer)  
    Get  
    Set
```

```
public:  
property Nullable<int> MinimumLength {  
    Nullable<int> get ();  
    void set (Nullable<int> value);  
}
```

### ▣ Value

The minimum length.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets a collection of options.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IDictionary<JToken, string> Options { get; se
```

```
Public Property Options As IDictionary(Of JToken, St  
    Get  
    Set
```

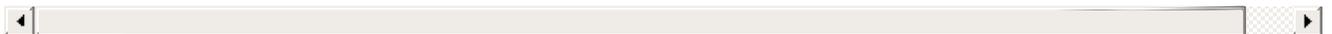
```
public:  
property IDictionary<JToken^, String^>^ Options {  
    IDictionary<JToken^, String^>^ get ();  
    void set (IDictionary<JToken^, String^>^ val  
}
```

## ▣ Value

A collection of options.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ► **Parse()**

C# ▼

## Members

Icon	Member	Description
	<a href="#">Parse(String)</a>	Load a <a href="#">JsonSchema</a> from a string that contains schema JSON.
	<a href="#">Parse(String, JsonSchemaResolver)</a>	Parses the specified json.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## Parse(String)

Load a [JsonSchema](#) from a string that contains schema JSON.

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JsonSchema Parse(  
    string json  
)
```

```
Public Shared Function Parse ( _  
    json As String _  
) As JsonSchema
```

```
public:  
static JsonSchema^ Parse(  
    String^ json  
)
```

### ▣ Parameters

*json* ([String](#))

A [String](#) that contains JSON.

### ▣ Return Value

A [JsonSchema](#) populated from the string that contains JSON.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Parse Method (json, resolver)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

### Parse(String, JsonSchemaResolver)

C#

Parses the specified json.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JsonSchema Parse(  
    string json,  
    JsonSchemaResolver resolver  
)
```

```
Public Shared Function Parse ( _  
    json As String, _  
    resolver As JsonSchemaResolver _  
) As JsonSchema
```

```
public:  
static JsonSchema^ Parse(  
    String^ json,  
    JsonSchemaResolver^ resolver  
)
```

#### Parameters

**json** ([String](#))

The json.

**resolver** ([JsonSchemaResolver](#))

The resolver.

#### Return Value

A [JsonSchema](#) populated from the string that contains JSON.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

Gets or sets the pattern.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Pattern { get; set; }
```

```
Public Property Pattern As String  
    Get  
    Set
```

```
public:  
property String^ Pattern {  
    String^ get ();  
    void set (String^ value);  
}
```

## ▣ Value

The pattern.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

### PatternProperties

Gets or sets the pattern properties.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public IDictionary<string, JsonSchema> PatternProperties
```

```
Public Property PatternProperties As IDictionary(Of  
    Get  
    Set
```

```
public:  
property IDictionary<String^, JsonSchema^>^ PatternP  
    IDictionary<String^, JsonSchema^>^ get ();  
    void set (IDictionary<String^, JsonSchema^>^  
}
```

#### Value

The pattern properties.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## Properties

Gets or sets the [JsonSchema](#) of properties.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IDictionary<string, JsonSchema> Properties {
```

```
Public Property Properties As IDictionary(Of String,  
Get  
Set
```

```
public:  
property IDictionary<String^, JsonSchema^>^ Properti  
IDictionary<String^, JsonSchema^>^ get ();  
void set (IDictionary<String^, JsonSchema^>^  
}
```

### ▣ Value

The [JsonSchema](#) of properties.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Members

Icon	Member	Description
	<a href="#">Read(JsonReader)</a>	Reads a <a href="#">JsonSchema</a> from the specified <a href="#">JsonReader</a> .
	<a href="#">Read(JsonReader, JsonSchemaResolver)</a>	Reads a <a href="#">JsonSchema</a> from the specified <a href="#">JsonReader</a> .

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## Read(JsonReader)

Reads a [JsonSchema](#) from the specified [JsonReader](#).

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public static <a href="#">JsonSchema</a> Read(     <a href="#">JsonReader</a> reader )</pre>		
	<pre>Public Shared Function Read (      reader As <a href="#">JsonReader</a>  ) As <a href="#">JsonSchema</a></pre>	
		<pre>public: static <a href="#">JsonSchema</a>^ Read(     <a href="#">JsonReader</a>^ reader )</pre>

### ▣ Parameters

#### ***reader*** ([JsonReader](#))

The [JsonReader](#) containing the JSON Schema to read.

### ▣ Return Value

The [JsonSchema](#) object representing the JSON Schema.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Read Method (reader, resolver)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

### Read(JsonReader, JsonSchemaResolver)

Reads a [JsonSchema](#) from the specified [JsonReader](#).

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public static JsonSchema Read(  
    JsonReader reader,  
    JsonSchemaResolver resolver  
)
```

```
Public Shared Function Read ( _  
    reader As JsonReader, _  
    resolver As JsonSchemaResolver _  
) As JsonSchema
```

```
public:  
static JsonSchema^ Read(  
    JsonReader^ reader,  
    JsonSchemaResolver^ resolver  
)
```

#### ▣ Parameters

##### **reader** ([JsonReader](#))

The [JsonReader](#) containing the JSON Schema to read.

##### **resolver** ([JsonSchemaResolver](#))

The [JsonSchemaResolver](#) to use when resolving schema references.

#### ▣ Return Value

The [JsonSchema](#) object representing the JSON Schema.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## ReadOnly

Gets or sets whether the object is read only.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<bool> ReadOnly { get; set; }
```

```
Public Property ReadOnly As Nullable(Of Boolean)  
    Get  
    Set
```

```
public:  
property Nullable<bool> ReadOnly {  
    Nullable<bool> get ();  
    void set (Nullable<bool> value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## Required

C#

Gets or sets whether the object is required.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<bool> Required { get; set; }
```

```
Public Property Required As Nullable(Of Boolean)  
    Get  
    Set
```

```
public:  
property Nullable<bool> Required {  
    Nullable<bool> get ();  
    void set (Nullable<bool> value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## Requires

C#

Gets or sets the required property if this property is present.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Requires { get; set; }
```

```
Public Property Requires As String  
    Get  
    Set
```

```
public:  
property String^ Requires {  
    String^ get ();  
    void set (String^ value);  
}
```

### ▣ Value

The required property if this property is present.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets the title.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public string Title { get; set; }
```

```
Public Property Title As String  
    Get  
    Set
```

```
public:  
property String^ Title {  
    String^ get ();  
    void set (String^ value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## ToString()

Returns a [String](#) that represents the current [Object](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

```
public:  
virtual String^ ToString() override
```

### ▣ Return Value

A [String](#) that represents the current [Object](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## Transient

Gets or sets whether the object is transient.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<bool> Transient { get; set; }
```

```
Public Property Transient As Nullable(Of Boolean)  
    Get  
    Set
```

```
public:  
property Nullable<bool> Transient {  
    Nullable<bool> get ();  
    void set (Nullable<bool> value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets the types of values allowed by the object.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">Nullable</a>&lt;<a href="#">JsonSchemaType</a>&gt; Type { <a href="#">get</a>; <a href="#">set</a>; }</pre>		
	<pre>Public Property Type As <a href="#">Nullable</a>(Of <a href="#">JsonSchemaType</a>)     Get     Set</pre>	
		<pre>public: property <a href="#">Nullable</a>&lt;<a href="#">JsonSchemaType</a>&gt; Type {     <a href="#">Nullable</a>&lt;<a href="#">JsonSchemaType</a>&gt; <a href="#">get</a> ();     void <a href="#">set</a> (<a href="#">Nullable</a>&lt;<a href="#">JsonSchemaType</a>&gt; <i>value</i>); }</pre>

## ▣ Value

The type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## WriteTo()

### Members

Icon	Member	Description
	<a href="#">WriteTo(JsonWriter)</a>	Writes this schema to a <a href="#">JsonWriter</a> .
	<a href="#">WriteTo(JsonWriter, JsonSchemaResolver)</a>	Writes this schema to a <a href="#">JsonWriter</a> using the specified <a href="#">JsonSchemaResolver</a> .

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## WriteTo(JsonWriter)

Writes this schema to a [JsonWriter](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void WriteTo(  
    JsonWriter writer  
)
```

```
Public Sub WriteTo ( _  
    writer As JsonWriter _  
)
```

```
public:  
void WriteTo(  
    JsonWriter^ writer  
)
```

### ▣ Parameters

#### ***writer*** ([JsonWriter](#))

A [JsonWriter](#) into which this method will write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## WriteTo Method (writer, resolver)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

### WriteTo(JsonWriter, JsonSchemaResolver)

Writes this schema to a [JsonWriter](#) using the specified [JsonSchemaResolver](#).

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public void WriteTo(  
    JsonWriter writer,  
    JsonSchemaResolver resolver  
)
```

```
Public Sub WriteTo ( _  
    writer As JsonWriter, _  
    resolver As JsonSchemaResolver _  
)
```

```
public:  
void WriteTo(  
    JsonWriter^ writer,  
    JsonSchemaResolver^ resolver  
)
```

#### Parameters

##### **writer** ([JsonWriter](#))

A [JsonWriter](#) into which this method will write.

##### **resolver** ([JsonSchemaResolver](#))

The resolver used.

(4.5.6.14930)

# JsonSchemaException

## Class

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► **JsonSchemaException**

C#

Returns detailed information about the schema exception.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
[SerializableAttribute]  
public class JsonSchemaException : JsonException
```

```
<SerializableAttribute> _  
Public Class JsonSchemaException _  
    Inherits JsonException
```

```
[SerializableAttribute]  
public ref class JsonSchemaException : public JsonEx
```

### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherite

Icon	Member	Description
	<a href="#">JsonSchemaException()</a>	Initializes a new instance of the <b>JsonSchemaException</b> class.
	<a href="#">JsonSchemaException(String)</a>	Initializes a new instance of the

		<b>JsonSchemaExceptio</b> class with a specified error message.
	<a href="#">JsonSchemaException(String, Exception)</a>	Initializes a new instance of the <b>JsonSchemaExceptio</b> class with a specified error message and a reference to the inner exception that is the cause of this exception
	<a href="#">JsonSchemaException(SerializationInfo, StreamingContext)</a>	Initializes a new instance of the <b>JsonSchemaExceptio</b> class.
	<a href="#">Data</a>	Gets a collection of key/value pairs that provide additional user- defined information about the exception.  (Inherited from <a href="#">Exception</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equi to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free

		resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetBaseException()</a>	When overridden in a derived class, returns the <a href="#">Exception</a> that is the root cause of one or more subsequent exceptions.  (Inherited from <a href="#">Exception</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetObjectData(SerializationInfo, StreamingContext)</a>	When overridden in a derived class, sets the <a href="#">SerializationInfo</a> with information about the exception.  (Inherited from <a href="#">Exception</a> .)
	<a href="#">GetType()</a>	Gets the runtime type of the current instance.  (Inherited from <a href="#">Exception</a> .)
	<a href="#">HelpLink</a>	Gets or sets a link to the help file associated with

		<p>this exception.</p> <p>(Inherited from <a href="#">Exception</a>.)</p>
	<a href="#">HRESULT</a>	<p>Gets or sets HRESULT a coded numerical value that is assigned to a specific exception.</p> <p>(Inherited from <a href="#">Exception</a>.)</p>
	<a href="#">InnerException</a>	<p>Gets the <a href="#">Exception</a> instance that caused the current exception.</p> <p>(Inherited from <a href="#">Exception</a>.)</p>
	<a href="#">LineNumber</a>	<p>Gets the line number indicating where the error occurred.</p>
	<a href="#">LinePosition</a>	<p>Gets the line position indicating where the error occurred.</p>
	<a href="#">MemberwiseClone()</a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">Message</a>	<p>Gets a message that describes the current exception.</p>

		(Inherited from <a href="#">Exception.</a> )
	<a href="#">Path</a>	Gets the path to the JSON where the error occurred.
	<a href="#">Source</a>	Gets or sets the name of the application or the object that causes the error.  (Inherited from <a href="#">Exception.</a> )
	<a href="#">StackTrace</a>	Gets a string representation of the frames on the call stack at the time the current exception was thrown.  (Inherited from <a href="#">Exception.</a> )
	<a href="#">TargetSite</a>	Gets the method that throws the current exception.  (Inherited from <a href="#">Exception.</a> )
	<a href="#">ToString()</a>	Creates and returns a string representation of the current exception.  (Inherited from <a href="#">Exception.</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [Exception](#)

└─ [JsonException](#)

└─ **JsonSchemaException**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaException](#) ► **JsonSchemaException()**

C#

### Members

Icon	Member	Description
	<a href="#">JsonSchemaException()</a>	Initializes a new instance of the <a href="#">JsonSchemaException</a> class.
	<a href="#">JsonSchemaException(String)</a>	Initializes a new instance of the <a href="#">JsonSchemaException</a> class with a specified error message.
	<a href="#">JsonSchemaException(String, Exception)</a>	Initializes a new instance of the <a href="#">JsonSchemaException</a> class with a specified error message and a reference to the inner exception that is the cause of this exception
	<a href="#">JsonSchemaException(SerializationInfo, StreamingContext)</a>	Initializes a new instance of the <a href="#">JsonSchemaException</a> class.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaException](#) ►

### JsonSchemaException()

Initializes a new instance of the [JsonSchemaException](#) class.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchemaException()
```

```
Public Sub New
```

```
public:  
JsonSchemaException()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonSchemaException

## Constructor (info, context)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaException](#) ► **JsonSchemaException(SerializationInfo, StreamingContext)**

Initializes a new instance of the [JsonSchemaException](#) class.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchemaException(  
    SerializationInfo info,  
    StreamingContext context  
)
```

```
Public Sub New ( _  
    info As SerializationInfo, _  
    context As StreamingContext _  
)
```

```
public:  
JsonSchemaException(  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

### Parameters

#### **info** ([SerializationInfo](#))

The [SerializationInfo](#) that holds the serialized object data about the exception being thrown.

#### **context** ([StreamingContext](#))

The [StreamingContext](#) that contains contextual information about the source or destination.

### Exceptions

Exception	Condition
<a href="#">ArgumentNullException</a>	The <i>info</i> parameter is null.
<a href="#">SerializationException</a>	The class name is null or <a href="#">HResult</a> is zero (0).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonSchemaException

## Constructor (message)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaException](#) ►

### JsonSchemaException(String)

Initializes a new instance of the [JsonSchemaException](#) class with a specified error message.

#### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonSchemaException(     <a href="#">string</a> message )</pre>	<pre>Public Sub New (      <a href="#">message</a> As <a href="#">String</a>  )</pre>	<pre>public: JsonSchemaException(     <a href="#">String</a>^ message )</pre>

#### Parameters

##### **message** ([String](#))

The error message that explains the reason for the exception.

## JsonSchemaException Constructor (message, innerException)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaException](#) ► **JsonSchemaException(String, Exception)**

Initializes a new instance of the [JsonSchemaException](#) class with a specified error message and a reference to the inner exception that is the cause of this exception.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchemaException(  
    string message,  
    Exception innerException  
)
```

```
Public Sub New ( _  
    message As String, _  
    innerException As Exception _  
)
```

```
public:  
JsonSchemaException(  
    String^ message,  
    Exception^ innerException  
)
```

### Parameters

#### **message** ([String](#))

The error message that explains the reason for the exception.

#### **innerException** ([Exception](#))

The exception that is the cause of the current exception, or a null reference (Nothing in Visual Basic) if no inner exception is specified.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaException](#) ►

## LineNumber

Gets the line number indicating where the error occurred.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public int LineNumber { get; private set; }
```

```
Public Property LineNumber As Integer  
    Get  
    Private Set
```

```
public:  
property int LineNumber {  
    int get ();  
    private: void set (int value);  
}
```

### ▣ Value

The line number indicating where the error occurred.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaException](#) ►

## LinePosition

C#

Gets the line position indicating where the error occurred.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public int LinePosition { get; private set; }
```

```
Public Property LinePosition As Integer  
    Get  
    Private Set
```

```
public:  
property int LinePosition {  
    int get ();  
    private: void set (int value);  
}
```

### Value

The line position indicating where the error occurred.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaException](#) ►

## Path

Gets the path to the JSON where the error occurred.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Path { get; private set; }
```

```
Public Property Path As String  
    Get  
    Private Set
```

```
public:  
property String^ Path {  
    String^ get ();  
    private: void set (String^ value);  
}
```

### ▣ Value

The path to the JSON where the error occurred.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► **JsonSchemaGenerator**

Generates a [JsonSchema](#) from a specified [Type](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonSchemaGenerator
```

```
Public Class JsonSchemaGenerator
```

```
public ref class JsonSchemaGenerator
```

### Members

All Members

Constructors

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declare
- Inherited

Icon	Member	Description
	<a href="#">JsonSchemaGenerator()</a>	Initializes a new instance of the <b>JsonSchemaGenerator</b> class
	<a href="#">ContractResolver</a>	Gets or sets the contract resolver.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)

	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">Generate(Type)</a>	Generate a <a href="#">JsonSchema</a> from the specified type.
	<a href="#">Generate(Type, JsonSchemaResolver)</a>	Generate a <a href="#">JsonSchema</a> from the specified type.
	<a href="#">Generate(Type, Boolean)</a>	Generate a <a href="#">JsonSchema</a> from the specified type.
	<a href="#">Generate(Type, JsonSchemaResolver, Boolean)</a>	Generate a <a href="#">JsonSchema</a> from the specified type.
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)

	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">UndefinedSchemaIdHandling</a>	Gets or sets how undefined schemas are handled by the serializer.

## ▣ Inheritance Hierarchy

[Object](#)

└─ **JsonSchemaGenerator**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaGenerator](#) ► **JsonSchemaGenerator()**

Initializes a new instance of the [JsonSchemaGenerator](#) class

  
class

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchemaGenerator()
```

```
Public Sub New
```

```
public:  
JsonSchemaGenerator()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaGenerator](#) ►

### ContractResolver

Gets or sets the contract resolver.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public IContractResolver ContractResolver { get; set
```

```
Public Property ContractResolver As IContractResol  
    Get  
    Set
```

```
public:  
property IContractResolver^ ContractResolver {  
    IContractResolver^ get ();  
    void set (IContractResolver^ value);  
}
```

#### ▣ Value

The contract resolver.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Generate()

### Members

Icon	Member	Description
	<a href="#">Generate(Type)</a>	Generate a <a href="#">JsonSchema</a> from the specified type.
	<a href="#">Generate(Type, JsonSchemaResolver)</a>	Generate a <a href="#">JsonSchema</a> from the specified type.
	<a href="#">Generate(Type, Boolean)</a>	Generate a <a href="#">JsonSchema</a> from the specified type.
	<a href="#">Generate(Type, JsonSchemaResolver, Boolean)</a>	Generate a <a href="#">JsonSchema</a> from the specified type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Generate(Type)

Generate a [JsonSchema](#) from the specified type.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">JsonSchema</a> Generate(     <i>Type type</i> )</pre>		

```
Public Function Generate ( _
    type As Type _
) As JsonSchema
```

```
public:
JsonSchema^ Generate(
    Type^ type
)
```

### ▣ Parameters

#### *type* ([Type](#))

The type to generate a [JsonSchema](#) from.

### ▣ Return Value

A [JsonSchema](#) generated from the specified type.

## Generate Method (type, resolver)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaGenerator](#) ► **Generate(Type, JsonSchemaResolver)**

Generate a [JsonSchema](#) from the specified type.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">JsonSchema</a> Generate(     <a href="#">Type</a> <i>type</i>,     <a href="#">JsonSchemaResolver</a> <i>resolver</i> )</pre>		
	<pre>Public Function Generate ( _     <i>type</i> As <a href="#">Type</a>, _     <i>resolver</i> As <a href="#">JsonSchemaResolver</a> _ ) As <a href="#">JsonSchema</a></pre>	
		<pre>public: <a href="#">JsonSchema</a>^ Generate(     <a href="#">Type</a>^ <i>type</i>,     <a href="#">JsonSchemaResolver</a>^ <i>resolver</i> )</pre>

### ▣ Parameters

#### ***type*** ([Type](#))

The type to generate a [JsonSchema](#) from.

#### ***resolver*** ([JsonSchemaResolver](#))

The [JsonSchemaResolver](#) used to resolve schema references.

### ▣ Return Value

A [JsonSchema](#) generated from the specified type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## Generate Method (type, resolver, rootSchemaNullable)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaGenerator](#) ►

**Generate(Type, JsonSchemaResolver, Boolean)**

Generate a [JsonSchema](#) from the specified type.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchema Generate(  
    Type type,  
    JsonSchemaResolver resolver,  
    bool rootSchemaNullable  
)
```

```
Public Function Generate ( _  
    type As Type, _  
    resolver As JsonSchemaResolver, _  
    rootSchemaNullable As Boolean _  
) As JsonSchema
```

```
public:  
JsonSchema^ Generate(  
    Type^ type,  
    JsonSchemaResolver^ resolver,  
    bool rootSchemaNullable  
)
```

### ▣ Parameters

#### **type** ([Type](#))

The type to generate a [JsonSchema](#) from.

#### **resolver** ([JsonSchemaResolver](#))

The [JsonSchemaResolver](#) used to resolve schema references.

***rootSchemaNullable*** ([Boolean](#))

Specify whether the generated root [JsonSchema](#) will be nullable.

▣ **Return Value**

A [JsonSchema](#) generated from the specified type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Generate Method (type, rootSchemaNullable)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaGenerator](#) ►

### Generate(Type, Boolean)

Generate a [JsonSchema](#) from the specified type.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">JsonSchema</a> Generate(     <i>Type</i> type,     <i>bool</i> rootSchemaNullable )</pre>		
	<pre>Public Function Generate ( _     <i>type</i> As <a href="#">Type</a>, _     <i>rootSchemaNullable</i> As <a href="#">Boolean</a> _ ) As <a href="#">JsonSchema</a></pre>	
		<pre>public: <a href="#">JsonSchema</a>^ Generate(     <i>Type</i>^ type,     <i>bool</i> rootSchemaNullable )</pre>

#### ▣ Parameters

##### **type** ([Type](#))

The type to generate a [JsonSchema](#) from.

##### **rootSchemaNullable** ([Boolean](#))

Specify whether the generated root [JsonSchema](#) will be nullable.

#### ▣ Return Value

A [JsonSchema](#) generated from the specified type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## UndefinedSchemaIdHandling

Gets or sets how undefined schemas are handled by the serializer.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public UndefinedSchemaIdHandling UndefinedSchemaIdHa
```

```
Public Property UndefinedSchemaIdHandling As Undefin  
    Get  
    Set
```

```
public:  
property UndefinedSchemaIdHandling UndefinedSchemaId  
    UndefinedSchemaIdHandling get ();  
    void set (UndefinedSchemaIdHandling value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Class

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► **JsonSchemaResolver**

C# ▼

Resolves [JsonSchema](#) from an id.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonSchemaResolver
```

```
Public Class JsonSchemaResolver
```

```
public ref class JsonSchemaResolver
```

### Members

All Members

Constructors

Methods

Properties

Public  
 Protected

Instance  
 Static

Declare  
 Inherited

Icon	Member	Description
	<a href="#">JsonSchemaResolver()</a>	Initializes a new instance of the <b>JsonSchemaResolver</b> class.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is

		reclaimed by garbage collection. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a> .)
	<a href="#">GetSchema(String)</a>	Gets a <a href="#">JsonSchema</a> for the specified id.
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance (Inherited from <a href="#">Object</a> .)
	<a href="#">LoadedSchemas</a>	Gets or sets the loaded schemas.
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> . (Inherited from <a href="#">Object</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

└─ **JsonSchemaResolver**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaResolver](#) ► **JsonSchemaResolver()**

Initializes a new instance of the [JsonSchemaResolver](#) class.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonSchemaResolver()</pre>		
	<pre>Public Sub New</pre>	
		<pre>public: JsonSchemaResolver()</pre>

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## GetSchema(String)

Gets a [JsonSchema](#) for the specified id.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public virtual JsonSchema GetSchema(
    string id
)
```

```
Public Overridable Function GetSchema ( _
    id As String _
) As JsonSchema
```

```
public:
virtual JsonSchema^ GetSchema(
    String^ id
)
```

### ▣ Parameters

***id*** ([String](#))

The id.

### ▣ Return Value

A [JsonSchema](#) for the specified id.

## Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaResolver](#) ►

### LoadedSchemas

Gets or sets the loaded schemas.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public IList<JsonSchema> LoadedSchemas { get; protec
```

```
Public Property LoadedSchemas As IList(Of JsonSchema  
    Get  
    Protected Set
```

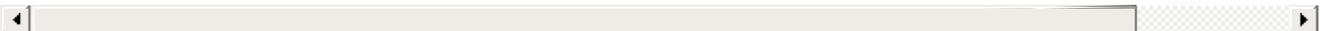
```
public:  
property IList<JsonSchema^> LoadedSchemas {  
    IList<JsonSchema^> get ();  
    protected: void set (IList<JsonSchema^> val  
}
```

#### Value

The loaded schemas.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Enumeration

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► **JsonSchemaType**

The value types allowed by the [JsonSchema](#).

### [-] Declaration Syntax

C#

Visual Basic

Visual C++

```
[FlagsAttribute]  
public enum JsonSchemaType
```

```
<FlagsAttribute> _  
Public Enumeration JsonSchemaType
```

```
[FlagsAttribute]  
public enum class JsonSchemaType
```

### [-] Members

Member	Description
None	No type specified.
String	String type.
Float	Float type.
Integer	Integer type.
Boolean	Boolean type.
Object	Object type.
Array	Array type.
Null	Null type.
Any	Any type.

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

Documentation **UndefinedSchemaIdHandling Enumeration**

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ►

**UndefinedSchemaIdHandling**

Specifies undefined schema Id handling options for the [JsonSchemaGenerator](#).

▣ **Declaration Syntax**

C#                      Visual Basic                      Visual C++

```
public enum UndefinedSchemaIdHandling
```

```
Public Enumeration UndefinedSchemaIdHandling
```

```
public enum class UndefinedSchemaIdHandling
```

▣ **Members**

Member	Description
<b>None</b>	Do not infer a schema Id.
<b>UseTypeName</b>	Use the .NET type name as the schema Id.
<b>UseAssemblyQualifiedName</b>	Use the assembly qualified .NET type name as the schema Id.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► **ValidationEventArgs**

C# ▼

Returns detailed information related to the [ValidationEventHandler](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class ValidationEventArgs : EventArgs
```

```
Public Class ValidationEventArgs _
    Inherits EventArgs
```

```
public ref class ValidationEventArgs : public EventArgs
```

### Members

All Members

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declare
- Inherited

Icon	Member	Description
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Exception</a>	Gets the <a href="#">JsonSchemaException</a> associated with the validation error.
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup

		operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Message</a>	Gets the text description corresponding to the validation error.
	<a href="#">Path</a>	Gets the path of the JSON location where the validation error occurred.
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

└─ [EventArgs](#)

└─ **ValidationEventArgs**

(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [ValidationEventArgs](#) ►

## Exception

Gets the [JsonSchemaException](#) associated with the validation error.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchemaException Exception { get; }
```

```
Public ReadOnly Property Exception As JsonSchemaExce  
Get
```

```
public:  
property JsonSchemaException^ Exception {  
    JsonSchemaException^ get ();  
}
```

### Value

The [JsonSchemaException](#) associated with the validation error.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [ValidationEventArgs](#) ►

## Message

Gets the text description corresponding to the validation error.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Message { get; }
```

```
Public ReadOnly Property Message As String  
    Get
```

```
public:  
property String^ Message {  
    String^ get ();  
}
```

### ▣ Value

The text description.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [ValidationEventArgs](#) ►

## Path

Gets the path of the JSON location where the validation error occurred.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Path { get; }
```

```
Public ReadOnly Property Path As String  
    Get
```

```
public:  
property String^ Path {  
    String^ get ();  
}
```

### ▣ Value

The path of the JSON location where the validation error occurred.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Delegate

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [ValidationEventHandler](#)

Represents the callback method that will handle JSON schema validation events and the [ValidationEventArgs](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public delegate void ValidationEventHandler(  
    Object sender,  
    ValidationEventArgs e  
)
```

```
Public Delegate Sub ValidationEventHandler ( _  
    sender As Object, _  
    e As ValidationEventArgs _  
)
```

```
public delegate void ValidationEventHandler(  
    Object^ sender,  
    ValidationEventArgs^ e  
)
```

### Parameters

*sender* ([Object](#))

*e* ([ValidationEventArgs](#))

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Namespace

[Namespaces](#) ► **Newtonsoft.Json.Serialization**

The **Newtonsoft.Json.Linq** namespace provides classes that are used when serializing and deserializing JSON.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
namespace Newtonsoft.Json.Serialization
```

```
Namespace Newtonsoft.Json.Serialization
```

```
namespace Newtonsoft.Json.Serialization
```

### ▣ Types

All Types	Classes	Interfaces	Delegates
-----------	---------	------------	-----------

Icon	Type	Description
	<a href="#">CamelCasePropertyNamesContractResolver</a>	Resolves member mappings for a type, camel casing property names.
	<a href="#">DefaultContractResolver</a>	Used by <a href="#">JsonSerializer</a> to resolve a <a href="#">JsonContract</a> for a given <a href="#">Type</a> .

	<a href="#"><u>DefaultSerializationBinder</u></a>	The default serialization binder used when resolving and loading classes from type names.
	<a href="#"><u>DynamicValueProvider</u></a>	Get and set values for a <a href="#"><u>MemberInfo</u></a> using dynamic methods.
	<a href="#"><u>ErrorContext</u></a>	Provides information surrounding an error.
	<a href="#"><u>ErrorEventArgs</u></a>	Provides data for the Error event.
	<a href="#"><u>IContractResolver</u></a>	Used by <a href="#"><u>JsonSerializer</u></a> to resolve a <a href="#"><u>JsonContract</u></a> for a given <a href="#"><u>Type</u></a> .
	<a href="#"><u>IReferenceResolver</u></a>	Used to resolve

		references when serializing and deserializing JSON by the <a href="#">JsonSerializer</a> .
	<a href="#"><b>IValueProvider</b></a>	Provides methods to get and set values.
	<a href="#"><b>JsonArrayContract</b></a>	Contract details for a <a href="#">Type</a> used by the <a href="#">JsonSerializer</a> .
	<a href="#"><b>JsonContainerContract</b></a>	Contract details for a <a href="#">Type</a> used by the <a href="#">JsonSerializer</a> .
	<a href="#"><b>JsonContract</b></a>	Contract details for a <a href="#">Type</a> used by the <a href="#">JsonSerializer</a> .
	<a href="#"><b>JsonDictionaryContract</b></a>	Contract details for a <a href="#">Type</a> used by the <a href="#">JsonSerializer</a> .

	<a href="#"><b>JsonDynamicContract</b></a>	Contract details for a <a href="#">Type</a> used by the <a href="#">JsonSerializer</a> .
	<a href="#"><b>JsonSerializableContract</b></a>	Contract details for a <a href="#">Type</a> used by the <a href="#">JsonSerializer</a> .
	<a href="#"><b>JsonLinqContract</b></a>	Contract details for a <a href="#">Type</a> used by the <a href="#">JsonSerializer</a> .
	<a href="#"><b>JsonObjectContract</b></a>	Contract details for a <a href="#">Type</a> used by the <a href="#">JsonSerializer</a> .
	<a href="#"><b>JsonPrimitiveContract</b></a>	Contract details for a <a href="#">Type</a> used by the <a href="#">JsonSerializer</a> .
	<a href="#"><b>JsonProperty</b></a>	Maps a JSON property to a .NET member or constructor parameter.

	<a href="#"><u>JsonPropertyCollection</u></a>	A collection of <a href="#"><u>JsonProperty</u></a> objects.
	<a href="#"><u>JsonStringContract</u></a>	Contract details for a <a href="#"><u>Type</u></a> used by the <a href="#"><u>JsonSerializer</u></a> .
	<a href="#"><u>ObjectConstructor&lt;T&gt;</u></a>	Represents a method that constructs an object.
	<a href="#"><u>OnErrorAttribute</u></a>	When applied to a method, specifies that the method is called when an error occurs serializing an object.
	<a href="#"><u>ReflectionValueProvider</u></a>	Get and set values for a <a href="#"><u>MemberInfo</u></a> using reflection.

---



Documentation **CamelCasePropertyNamesContractResolver Class**

[Namespaces](#) ▶ [Newtonsoft.Json.Serialization](#) ▶

**CamelCasePropertyNamesContractResolver**

Resolves member mappings for a type, camel casing property names.

▣ **Declaration Syntax**

C#      Visual Basic      Visual C++

```
public class CamelCasePropertyNamesContractResolver
```

```
Public Class CamelCasePropertyNamesContractResolver
    Inherits DefaultContractResolver
```

```
public ref class CamelCasePropertyNamesContractResol
```

▣ **Members**

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declare <input checked="" type="checkbox"/> Inherited

Icon	Member	Descr
	<a href="#">CamelCasePropertyNamesContractResolver()</a>	Initiali: <b>Came</b>
	<a href="#">CreateArrayContract(Type)</a>	Create (Inheri
	<a href="#">CreateConstructorParameters(ConstructorInfo, JsonPropertyCollection)</a>	Create

		(Inheri
	<a href="#">CreateContract(Type)</a>	Deterr type.  (Inheri
	<a href="#">CreateDictionaryContract(Type)</a>	Create  (Inheri
	<a href="#">CreateDynamicContract(Type)</a>	Create  (Inheri
	<a href="#">CreateISerializableContract(Type)</a>	Create  (Inheri
	<a href="#">CreateLinqContract(Type)</a>	Create  (Inheri
	<a href="#">CreateMemberValueProvider(MemberInfo)</a>	Create and se  (Inheri
	<a href="#">CreateObjectContract(Type)</a>	Create  (Inheri
	<a href="#">CreatePrimitiveContract(Type)</a>	Create  (Inheri
	<a href="#">CreateProperties(Type, MemberSerialization)</a>	Create  (Inheri
	<a href="#">CreateProperty(MemberInfo, MemberSerialization)</a>	Create  (Inheri
	<a href="#">CreatePropertyFromConstructorParameter(JsonProperty,</a>	Create

	<a href="#">ParameterInfo</a>	(Inheri
	<a href="#">CreateStringContract(Type)</a>	Create (Inheri
	<a href="#">DefaultMembersSearchFlags</a>	Gets c (Inheri
	<a href="#">DynamicCodeGeneration</a>	Gets a and se determ (Inheri
	<a href="#">Equals(Object)</a>	Determ curren (Inheri
	<a href="#">Finalize()</a>	Allows perfor reclai (Inheri
	<a href="#">GetHashCode()</a>	Serve: (Inheri
	<a href="#">GetResolvedPropertyName(String)</a>	Gets t (Inheri
	<a href="#">GetSerializableMembers(Type)</a>	Gets t (Inheri
	<a href="#">GetType()</a>	Gets t (Inheri

	<a href="#">IgnoreSerializableAttribute</a>	Gets c <a href="#">Seriali</a> deseri  (Inheri
	<a href="#">IgnoreSerializableInterface</a>	Gets c <a href="#">Serial</a> types.  (Inheri
	<a href="#">MemberwiseClone()</a>	Create  (Inheri
	<a href="#">ResolveContract(Type)</a>	Resolv  (Inheri
	<a href="#">ResolveContractConverter(Type)</a>	Resolv  (Inheri
	<a href="#">ResolvePropertyName(String)</a>	Resolv  (Overr <a href="#">Defau</a>
	<a href="#">SerializeCompilerGeneratedMembers</a>	Gets c genera  (Inheri
	<a href="#">ToString()</a>	Returr  (Inheri

## [-] Inheritance Hierarchy

[Object](#)

└─ [DefaultContractResolver](#)

## CamelCasePropertyNamesContractResolver

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



Documentation **CamelCasePropertyNamesContractResolver**  
**Constructor**

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[CamelCasePropertyNamesContractResolver](#) ►

### CamelCasePropertyNamesContractResolver()

Initializes a new instance of the [CamelCasePropertyNamesContractResolver](#) class.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public CamelCasePropertyNamesContractResolver()
```

```
Public Sub New
```

```
public:  
CamelCasePropertyNamesContractResolver()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Method (propertyName)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[CamelCasePropertyNameContractResolver](#) ►

### ResolvePropertyName(String)

Resolves the name of the property.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected internal override string ResolvePropertyName(
    string propertyName
)
```

```
Protected Friend Overrides Function ResolvePropertyName(
    propertyName As String _
) As String
```

```
protected public:
virtual String^ ResolvePropertyName(
    String^ propertyName
) override
```

#### ▣ Parameters

**propertyName** ([String](#))

Name of the property.

#### ▣ Return Value

The property name camel cased.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

### DefaultContractResolver

Used by [JsonSerializer](#) to resolves a [JsonContract](#) for a given [Type](#).

#### Declaration Syntax

C#	Visual Basic	Visual C++
<code>public class DefaultContractResolver : <a href="#">IContractReso</a></code>		

```
Public Class DefaultContractResolver _
    Implements IContractResolver
```

```
public ref class DefaultContractResolver : IContract
```

#### Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declare <input checked="" type="checkbox"/> Inherited
Icon	Member	Description	
	<a href="#">DefaultContractResolver()</a>	Initiali: of the <b>Defau</b> class.	
	<a href="#">DefaultContractResolver(Boolean)</a>	Initiali: of the <b>Defau</b> class.	

	<a href="#">CreateArrayContract(Type)</a>	Create <a href="#">JsonA</a> given
	<a href="#">CreateConstructorParameters(ConstructorInfo, JsonPropertyCollection)</a>	Create param
	<a href="#">CreateContract(Type)</a>	Deterr contra for the
	<a href="#">CreateDictionaryContract(Type)</a>	Create <a href="#">JsonD</a> for the
	<a href="#">CreateDynamicContract(Type)</a>	Create <a href="#">JsonD</a> the giv
	<a href="#">CreateISerializableContract(Type)</a>	Create <a href="#">JsonIS</a> for the
	<a href="#">CreateLinqContract(Type)</a>	Create <a href="#">JsonL</a> given

	<a href="#">CreateMemberValueProvider(MemberInfo)</a>	Create <a href="#">IValue</a> the se set va memb
	<a href="#">CreateObjectContract(Type)</a>	Create <a href="#">JsonC</a> the giv
	<a href="#">CreatePrimitiveContract(Type)</a>	Create <a href="#">JsonP</a> the giv
	<a href="#">CreateProperties(Type, MemberSerialization)</a>	Create given .
	<a href="#">CreateProperty(MemberInfo, MemberSerialization)</a>	Create for the
	<a href="#">CreatePropertyFromConstructorParameter(JsonProperty, ParameterInfo)</a>	Create for the <a href="#">Param</a>
	<a href="#">CreateStringContract(Type)</a>	Create <a href="#">JsonS</a> given .

	<a href="#">DefaultMembersSearchFlags</a>	Gets c memb
	<a href="#">DynamicCodeGeneration</a>	Gets a wheth being dynam This v by the permis
	<a href="#">Equals(Object)</a>	Deterr specifi to the  (Inheri
	<a href="#">Finalize()</a>	Allows attemp and pe operat <a href="#">Object</a> garbaq  (Inheri
	<a href="#">GetHashCode()</a>	Serve: for a p  (Inheri
	<a href="#">GetResolvedPropertyName(String)</a>	Gets t the pro

	<a href="#">GetSerializableMembers(Type)</a>	Gets t memb
	<a href="#">GetType()</a>	Gets t curren  (Inheri
	<a href="#">IgnoreSerializableAttribute</a>	Gets c indica ignore <a href="#">Seriali</a> attribu and de
	<a href="#">IgnoreSerializableInterface</a>	Gets c indica ignore interfa and de
	<a href="#">MemberwiseClone()</a>	Create the cu  (Inheri
	<a href="#">ResolveContract(Type)</a>	Resolv a give
	<a href="#">ResolveContractConverter(Type)</a>	Resolv <a href="#">JsonC</a> contra

	<a href="#">ResolvePropertyName(String)</a>	Resolv propert
	<a href="#">SerializeCompilerGeneratedMembers</a>	Gets c indica compi memb seriali.
	<a href="#">ToString()</a>	Return repres <a href="#">Object</a>  (Inheri

## ▣ Inheritance Hierarchy

[Object](#)

└─ **DefaultContractResolver**

└─ [CamelCasePropertyNamesContractResolver](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DefaultContractResolver](#) ► **DefaultContractResolver()**

### Members

Icon	Member	Description
	<a href="#">DefaultContractResolver()</a>	Initializes a new instance of the <a href="#">DefaultContractResolver</a> class.
	<a href="#">DefaultContractResolver(Boolean)</a>	Initializes a new instance of the <a href="#">DefaultContractResolver</a> class.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DefaultContractResolver](#) ► **DefaultContractResolver()**

Initializes a new instance of the [DefaultContractResolver](#) class.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public DefaultContractResolver()
```

```
Public Sub New
```

```
public:  
DefaultContractResolver()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# DefaultContractResolver Constructor (shareCache)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DefaultContractResolver](#) ► **DefaultContractResolver(Boolean)**

Initializes a new instance of the [DefaultContractResolver](#) class.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public DefaultContractResolver(  
    bool shareCache  
)
```

```
Public Sub New (   
    shareCache As Boolean   
)
```

```
public:  
DefaultContractResolver(  
    bool shareCache  
)
```

## Parameters

### **shareCache (Boolean)**

If set to `true` the [DefaultContractResolver](#) will use a cached share with other resolvers of the same type. Sharing the cache will significantly performance because expensive reflection will only happen once but could cause unexpected behavior if different instances of the resolver are suppose to produce different results. When set to `false` it is highly recommended to reuse [DefaultContractResolver](#) instances with the [JsonSerializer](#).

# CreateArrayContract Method (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[DefaultContractResolver](#) ► **CreateArrayContract(Type)**

Creates a [JsonArrayContract](#) for the given type.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual JsonArrayContract CreateArrayContract(
    Type objectType
)
```

```
Protected Overridable Function CreateArrayContract (
    objectType As Type _
) As JsonArrayContract
```

```
protected:
virtual JsonArrayContract^ CreateArrayContract(
    Type^ objectType
)
```

## ▣ Parameters

***objectType*** ([Type](#))

Type of the object.

## ▣ Return Value

A [JsonArrayContract](#) for the given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



Documentation **CreateConstructorParameters Method  
(constructor, memberProperties)**

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DefaultContractResolver](#) ►

**CreateConstructorParameters(ConstructorInfo, JsonPropertyCollection)**

Creates the constructor parameters.

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
<pre>protected virtual IList&lt;JsonProperty&gt; CreateConstructorParameters(ConstructorInfo constructor, JsonPropertyCollection memberProperties)</pre>		

```
Protected Overridable Function CreateConstructorParameters(constructor As ConstructorInfo, _ memberProperties As JsonPropertyCollection _ ) As IList(Of JsonProperty)
```

```
protected: virtual IList<JsonProperty^>^ CreateConstructorParameters(ConstructorInfo^ constructor, JsonPropertyCollection^ memberProperties)
```

▣ **Parameters**

**constructor** ([ConstructorInfo](#))

The constructor to create properties for.

**memberProperties** ([JsonPropertyCollection](#))

The type's member properties.

## Return Value

Properties for the given [ConstructorInfo](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DefaultContractResolver](#) ► **CreateContract(Type)**

Determines which contract type is created for the given type.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
protected virtual JsonContract CreateContract(  
    Type objectType  
)
```

```
Protected Overridable Function CreateContract ( _  
    objectType As Type _  
) As JsonContract
```

```
protected:  
virtual JsonContract^ CreateContract(  
    Type^ objectType  
)
```

### ▣ Parameters

#### ***objectType*** ([Type](#))

Type of the object.

### ▣ Return Value

A [JsonContract](#) for the given type.

# CreateDictionaryContract Method (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DefaultContractResolver](#) ► **CreateDictionaryContract(Type)**

Creates a [JsonDictionaryContract](#) for the given type.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual JsonDictionaryContract CreateDicti  
    Type objectType  
)
```

```
Protected Overridable Function CreateDictionaryContr  
    objectType As Type _  
) As JsonDictionaryContract
```

```
protected:  
virtual JsonDictionaryContract^ CreateDictionaryCont  
    Type^ objectType  
)
```

## ▣ Parameters

***objectType*** ([Type](#))

Type of the object.

## ▣ Return Value

A [JsonDictionaryContract](#) for the given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# CreateDynamicContract Method (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DefaultContractResolver](#) ► **CreateDynamicContract(Type)**

Creates a [JsonDynamicContract](#) for the given type.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual JsonDynamicContract CreateDynamicContract(
    Type objectType
)
```

```
Protected Overridable Function CreateDynamicContract(
    objectType As Type _
) As JsonDynamicContract
```

```
protected:
virtual JsonDynamicContract^ CreateDynamicContract(
    Type^ objectType
)
```

## ▣ Parameters

***objectType*** ([Type](#))

Type of the object.

## ▣ Return Value

A [JsonDynamicContract](#) for the given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



Documentation **CreateISerializableContract Method (objectType)**

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DefaultContractResolver](#) ► **CreateISerializableContract(Type)**

Creates a [JsonISerializableContract](#) for the given type.

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
<pre>protected virtual <a href="#">JsonISerializableContract</a> CreateISerializableContract(     Type objectType )</pre>		

```
Protected Overridable Function CreateISerializableContract(  
    objectType As Type _  
) As JsonISerializableContract
```

```
protected:  
virtual JsonISerializableContract^ CreateISerializableContract(  
    Type^ objectType  
)
```

▣ **Parameters**

***objectType*** ([Type](#))  
Type of the object.

▣ **Return Value**

A [JsonISerializableContract](#) for the given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)





## Method (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DefaultContractResolver](#) ► **CreateLinqContract(Type)**

Creates a [JsonLinqContract](#) for the given type.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual JsonLinqContract CreateLinqContract  
    (  
        Type objectType  
    )
```

```
Protected Overridable Function CreateLinqContract (  
    objectType As Type _  
) As JsonLinqContract
```

```
protected:  
virtual JsonLinqContract^ CreateLinqContract(  
    Type^ objectType  
)
```

### ▣ Parameters

**objectType** ([Type](#))

Type of the object.

### ▣ Return Value

A [JsonLinqContract](#) for the given type.

Documentation **CreateMemberValueProvider Method (member)**

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DefaultContractResolver](#) ►

**CreateMemberValueProvider(MemberInfo)**

C# ▼

Creates the [IValueProvider](#) used by the serializer to get and set values from a member.

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
<pre>protected virtual <a href="#">IValueProvider</a> CreateMemberValuePr     <a href="#">MemberInfo</a> member )</pre>		

```
Protected Overridable Function CreateMemberValueProv
    member As MemberInfo _
) As IValueProvider
```

```
protected:
virtual IValueProvider^ CreateMemberValueProvider(
    MemberInfo^ member
)
```

▣ **Parameters**

***member*** ([MemberInfo](#))  
The member.

▣ **Return Value**

The [IValueProvider](#) used by the serializer to get and set values from a member.

(4.5.6.14930)



# CreateObjectContract Method (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DefaultContractResolver](#) ► **CreateObjectContract(Type)**

C#

Creates a [JsonObjectContract](#) for the given type.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual JsonObjectContract CreateObjectContract(
    Type objectType
)
```

```
Protected Overridable Function CreateObjectContract(
    objectType As Type _
) As JsonObjectContract
```

```
protected:
virtual JsonObjectContract^ CreateObjectContract(
    Type^ objectType
)
```

## Parameters

***objectType*** ([Type](#))

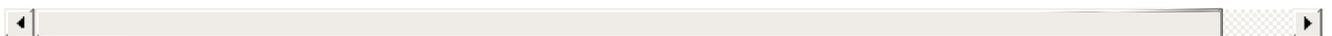
Type of the object.

## Return Value

A [JsonObjectContract](#) for the given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# CreatePrimitiveContract Method (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DefaultContractResolver](#) ► **CreatePrimitiveContract(Type)**

Creates a [JsonPrimitiveContract](#) for the given type.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual JsonPrimitiveContract CreatePrimit  
    Type objectType  
)
```

```
Protected Overridable Function CreatePrimitiveContra  
    objectType As Type _  
) As JsonPrimitiveContract
```

```
protected:  
virtual JsonPrimitiveContract^ CreatePrimitiveContra  
    Type^ objectType  
)
```

## Parameters

***objectType*** ([Type](#))

Type of the object.

## Return Value

A [JsonPrimitiveContract](#) for the given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## CreateProperties Method (type, memberSerialization)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DefaultContractResolver](#) ► **CreateProperties(Type, MemberSerialization)**

Creates properties for the given [JsonContract](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual IList<JsonProperty> CreateProperti
    Type type,
    MemberSerialization memberSerialization
)
```

```
Protected Overridable Function CreateProperties ( _
    type As Type, _
    memberSerialization As MemberSerialization _
) As IList(Of JsonProperty)
```

```
protected:
virtual IList<JsonProperty^>^ CreateProperties(
    Type^ type,
    MemberSerialization memberSerialization
)
```

### Parameters

#### **type (Type)**

The type to create properties for.

#### **memberSerialization (MemberSerialization)**

The member serialization mode for the type.

### Return Value

Properties for the given [JsonContract](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## CreateProperty Method (member, memberSerialization)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DefaultContractResolver](#) ► **CreateProperty(MemberInfo, MemberSerialization)**

Creates a [JsonProperty](#) for the given [MemberInfo](#).

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual JsonProperty CreateProperty(  
    MemberInfo member,  
    MemberSerialization memberSerialization  
)
```

```
Protected Overridable Function CreateProperty ( _  
    member As MemberInfo, _  
    memberSerialization As MemberSerialization _  
) As JsonProperty
```

```
protected:  
virtual JsonProperty^ CreateProperty(  
    MemberInfo^ member,  
    MemberSerialization memberSerialization  
)
```

### ▣ Parameters

#### **member** ([MemberInfo](#))

The member to create a [JsonProperty](#) for.

#### **memberSerialization** ([MemberSerialization](#))

The member's parent [MemberSerialization](#).

### ▣ Return Value

A created [JsonProperty](#) for the given [MemberInfo](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

Documentation **CreatePropertyFromConstructorParameter Method (matchingMemberProperty, parameterInfo)**

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DefaultContractResolver](#) ►

**CreatePropertyFromConstructorParameter(JsonProperty, ParameterInfo)**

Creates a [JsonProperty](#) for the given [ParameterInfo](#).

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
<pre>protected virtual <a href="#">JsonProperty</a> CreatePropertyFromCon     <a href="#">JsonProperty</a> <i>matchingMemberProperty</i>,     <a href="#">ParameterInfo</a> <i>parameterInfo</i> ) </pre>		

```
Protected Overridable Function CreatePropertyFromCon
    matchingMemberProperty As JsonProperty, _
    parameterInfo As ParameterInfo _
) As JsonProperty
```

```
protected:
virtual JsonProperty^ CreatePropertyFromConstructorP
    JsonProperty^ matchingMemberProperty,
    ParameterInfo^ parameterInfo
)

```

▣ **Parameters**

***matchingMemberProperty* ([JsonProperty](#))**

The matching member property.

***parameterInfo* ([ParameterInfo](#))**

The constructor parameter.

## ▣ Return Value

A created [JsonProperty](#) for the given [ParameterInfo](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)



# CreateStringContract Method (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DefaultContractResolver](#) ► **CreateStringContract(Type)**

C#

Creates a [JsonStringContract](#) for the given type.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual JsonStringContract CreateStringContract(
    Type objectType
)
```

```
Protected Overridable Function CreateStringContract(
    objectType As Type _
) As JsonStringContract
```

```
protected:
virtual JsonStringContract^ CreateStringContract(
    Type^ objectType
)
```

## Parameters

***objectType*** ([Type](#))

Type of the object.

## Return Value

A [JsonStringContract](#) for the given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



Gets or sets the default members search flags.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public BindingFlags DefaultMembersSearchFlags { get;
```

```
Public Property DefaultMembersSearchFlags As Binding  
    Get  
    Set
```

```
public:  
property BindingFlags DefaultMembersSearchFlags {  
    BindingFlags get ();  
    void set (BindingFlags value);  
}
```

### ▣ Value

The default members search flags.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



**Property**

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DefaultContractResolver](#) ► **DynamicCodeGeneration**

Gets a value indicating whether members are being get and set using dynamic code generation. This value is determined by the runtime permissions available.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public bool DynamicCodeGeneration { get; }
```

```
Public ReadOnly Property DynamicCodeGeneration As Boolean  
    Get
```

```
public:  
property bool DynamicCodeGeneration {  
    bool get ();  
}
```

**Value**

**true** if using dynamic code generation; otherwise, **false**.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



Documentation **GetResolvedPropertyName Method (propertyName)**

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DefaultContractResolver](#) ► **GetResolvedPropertyName(String)**

Gets the resolved name of the property.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public string GetResolvedPropertyName(  
    string propertyName  
)
```

```
Public Function GetResolvedPropertyName ( _  
    propertyName As String _  
) As String
```

```
public:  
String^ GetResolvedPropertyName(  
    String^ propertyName  
)
```

▣ **Parameters**

***propertyName*** (**String**)

Name of the property.

▣ **Return Value**

Name of the property.

## Method (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DefaultContractResolver](#) ► **GetSerializableMembers(Type)**

Gets the serializable members for the type.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual List<MemberInfo> GetSerializableMembers(
    Type objectType
)
```

```
Protected Overridable Function GetSerializableMembers(
    objectType As Type _
) As List(Of MemberInfo)
```

```
protected:
virtual List<MemberInfo^>^ GetSerializableMembers(
    Type^ objectType
)
```

### ▣ Parameters

#### ***objectType*** ([Type](#))

The type to get serializable members for.

### ▣ Return Value

The serializable members for the type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Gets or sets a value indicating whether to ignore the [SerializableAttribute](#) attribute when serializing and deserializing types.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool IgnoreSerializableAttribute { get; set; }
```

```
Public Property IgnoreSerializableAttribute As Boolean  
    Get  
    Set
```

```
public:  
property bool IgnoreSerializableAttribute {  
    bool get ();  
    void set (bool value);  
}
```

### ▣ Value

`true` if the [SerializableAttribute](#) attribute will be ignored when serializing and deserializing types; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►[DefaultContractResolver](#) ► **IgnoreSerializableInterface**

Gets or sets a value indicating whether to ignore the [ISerializable](#) interface when serializing and deserializing types.

### ▣ Declaration Syntax

 C# Visual Basic Visual C++

```
public bool IgnoreSerializableInterface { get; set; }
```

```
Public Property IgnoreSerializableInterface As Boolean  
    Get  
    Set
```

```
public:  
property bool IgnoreSerializableInterface {  
    bool get ();  
    void set (bool value);  
}
```

### ▣ Value

`true` if the [ISerializable](#) interface will be ignored when serializing and deserializing types; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (type)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DefaultContractResolver](#) ► **ResolveContract(Type)**

Resolves the contract for a given type.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual JsonContract ResolveContract(  
    Type type  
)
```

```
Public Overridable Function ResolveContract ( _  
    type As Type _  
) As JsonContract
```

```
public:  
virtual JsonContract^ ResolveContract(  
    Type^ type  
)
```

### ▣ Parameters

**type** ([Type](#))

The type to resolve a contract for.

### ▣ Return Value

The contract for a given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Documentation **ResolveContractConverter Method (objectType)**

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DefaultContractResolver](#) ► **ResolveContractConverter(Type)**

Resolves the default [JsonConverter](#) for the contract.

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
----	--------------	------------

```
protected virtual JsonConverter ResolveContractConverter(
    Type objectType
)
```

```
Protected Overridable Function ResolveContractConverter(
    objectType As Type _
) As JsonConverter
```

```
protected:
virtual JsonConverter^ ResolveContractConverter(
    Type^ objectType
)
```

▣ **Parameters**

***objectType*** ([Type](#))  
 Type of the object.

▣ **Return Value**

**[Missing <returns> documentation for "M:Newtonsoft.Json.Serialization.DefaultContractResolver.ResolveContractConverter(Type)"]**



## Method (propertyName)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DefaultContractResolver](#) ► **ResolvePropertyName(String)**

Resolves the name of the property.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected internal virtual string ResolvePropertyName(  
    string propertyName  
)
```

```
Protected Friend Overridable Function ResolveProperty  
    propertyName As String _  
) As String
```

```
protected public:  
virtual String^ ResolvePropertyName(  
    String^ propertyName  
)
```

### ▣ Parameters

***propertyName*** (**String**)

Name of the property.

### ▣ Return Value

Name of the property.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Documentation **SerializeCompilerGeneratedMembers**  
**Property**[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►[DefaultContractResolver](#) ► **SerializeCompilerGeneratedMembers**

C#

Gets or sets a value indicating whether compiler generated members should be serialized.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public bool SerializeCompilerGeneratedMembers { get;
```

```
Public Property SerializeCompilerGeneratedMembers As  
    Get  
    Set
```

```
public:  
property bool SerializeCompilerGeneratedMembers {  
    bool get ();  
    void set (bool value);  
}
```

▣ **Value**

true if serialized compiler generated members; otherwise, false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



Documentation **DefaultSerializationBinder Class**

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

**DefaultSerializationBinder**

C# ▼

The default serialization binder used when resolving and loading classes from type names.

[-] **Declaration Syntax**

C# Visual Basic Visual C++

```
public class DefaultSerializationBinder : SerializationBinder
```

```
Public Class DefaultSerializationBinder _
    Inherits SerializationBinder
```

```
public ref class DefaultSerializationBinder : public
```

[-] **Members**

All Members	Constructors	Methods	
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">DefaultSerializationBinder()</a>	Initializes a new instance of the <b>DefaultSerializationBinder</b> class
	<a href="#">BindToName(Type, String, String)</a>	When overridden in a derived class, controls the binding of a serialized object to a type.

	<a href="#">BindToType(String, String)</a>	When overridden in a derived class, controls the binding of a serialized object to a type.  (Overrides <a href="#">SerializationBinder.BindToType(String, String)</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

[SerializationBinder](#)

[DefaultSerializationBinder](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



Initializes a new instance of the [DefaultSerializationBinder](#) class

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public DefaultSerializationBinder()
```

```
Public Sub New
```

```
public:  
DefaultSerializationBinder()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## BindToName Method (serializedType, assemblyName, typeName)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DefaultSerializationBinder](#) ► **BindToName(Type, String, String)**

When overridden in a derived class, controls the binding of a serialized object to a type.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void BindToName(  
    Type serializedType,  
    out string assemblyName,  
    out string typeName  
)
```

```
Public Overridable Sub BindToName ( _  
    serializedType As Type, _  
    <OutAttribute> ByRef assemblyName As String,  
    <OutAttribute> ByRef typeName As String _  
)
```

```
public:  
virtual void BindToName(  
    Type^ serializedType,  
    [OutAttribute] String^% assemblyName,  
    [OutAttribute] String^% typeName  
)
```

### ▣ Parameters

#### **serializedType (Type)**

The type of the object the formatter creates a new instance of.

#### **assemblyName (String)**

Specifies the [Assembly](#) name of the serialized object.

***typeName*** ([String](#))

Specifies the [Type](#) name of the serialized object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## BindToType Method (assemblyName, typeName)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DefaultSerializationBinder](#) ► **BindToType(String, String)**

When overridden in a derived class, controls the binding of a serialized object to a type.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Type BindToType(  
    string assemblyName,  
    string typeName  
)
```

```
Public Overrides Function BindToType ( _  
    assemblyName As String, _  
    typeName As String _  
) As Type
```

```
public:  
virtual Type^ BindToType(  
    String^ assemblyName,  
    String^ typeName  
) override
```

### Parameters

#### **assemblyName (String)**

Specifies the [Assembly](#) name of the serialized object.

#### **typeName (String)**

Specifies the [Type](#) name of the serialized object.

### Return Value

The type of the object the formatter creates a new instance of.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

# DynamicValueProvider

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

### DynamicValueProvider

Get and set values for a [MemberInfo](#) using dynamic methods.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class DynamicValueProvider : IValueProvider
```

```
Public Class DynamicValueProvider _
    Implements IValueProvider
```

```
public ref class DynamicValueProvider : IValueProvid
```

#### ▣ Members

All Members	Constructors	Methods	
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherite

Icon	Member	Description
	<a href="#">DynamicValueProvider(MemberInfo)</a>	Initializes a new instance of the <b>DynamicValueProvide</b> class.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)

	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetValue(Object)</a>	Gets the value.
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">SetValue(Object, Object)</a>	Sets the value.
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

## DynamicValueProvider

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# DynamicValueProvider Constructor (memberInfo)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[DynamicValueProvider](#) ► **DynamicValueProvider(MemberInfo)**

Initializes a new instance of the [DynamicValueProvider](#) class.

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public DynamicValueProvider(     <a href="#">MemberInfo</a> memberInfo )</pre>		
	<pre>Public Sub New (      <i>memberInfo</i> As <a href="#">MemberInfo</a>  )</pre>	
		<pre>public: DynamicValueProvider(     <a href="#">MemberInfo</a>^ <i>memberInfo</i> )</pre>

## ▣ Parameters

### *memberInfo* ([MemberInfo](#))

The member info.

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DynamicValueProvider](#) ► **GetValue(Object)**

Gets the value.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Object GetValue(  
    Object target  
)
```

```
Public Function GetValue ( _  
    target As Object _  
) As Object
```

```
public:  
virtual Object^ GetValue(  
    Object^ target  
) sealed
```

## ▣ Parameters

**target** ([Object](#))

The target to get the value from.

## ▣ Return Value

The value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## SetValue Method (target, value)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DynamicValueProvider](#) ► **SetValue(Object, Object)**

Sets the value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void SetValue(  
    Object target,  
    Object value  
)
```

```
Public Sub SetValue ( _  
    target As Object, _  
    value As Object _  
)
```

```
public:  
virtual void SetValue(  
    Object^ target,  
    Object^ value  
) sealed
```

### ▣ Parameters

#### **target (Object)**

The target to set the value on.

#### **value (Object)**

The value to set on the target.

Provides information surrounding an error.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public class ErrorContext
```

```
Public Class ErrorContext
```

```
public ref class ErrorContext
```

## Members

All Members

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declare
- Inherited

Icon	Member	Description
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Error</a>	Gets or sets the error.
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)

	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance  (Inherited from <a href="#">Object</a> .)
	<a href="#">Handled</a>	Gets or sets a value indicating whether this <b>ErrorContext</b> is handled.
	<a href="#">Member</a>	Gets the member that caused the error.
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">OriginalObject</a>	Gets the original object that caused the error.
	<a href="#">Path</a>	Gets the path of the JSON location where the error occurred.
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

## **ErrorContext**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [ErrorContext](#) ► **Error**

C# ▼

Gets or sets the error.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Exception Error { get; private set; }
```

```
Public Property Error As Exception  
    Get  
    Private Set
```

```
public:  
property Exception^ Error {  
    Exception^ get ();  
    private: void set (Exception^ value);  
}
```

## ▣ Value

The error.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [ErrorContext](#) ►

## Handled

Gets or sets a value indicating whether this [ErrorContext](#) is handled.

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Handled { get; set; }
```

```
Public Property Handled As Boolean  
    Get  
    Set
```

```
public:  
property bool Handled {  
    bool get ();  
    void set (bool value);  
}
```

### ▣ Value

`true` if handled; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [ErrorContext](#) ►

## Member

Gets the member that caused the error.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Object Member { get; private set; }
```

```
Public Property Member As Object  
    Get  
    Private Set
```

```
public:  
property Object^ Member {  
    Object^ get ();  
    private: void set (Object^ value);  
}
```

### ▣ Value

The member that caused the error.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [ErrorContext](#) ►**OriginalObject**

Gets the original object that caused the error.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public Object OriginalObject { get; private set; }
```

```
Public Property OriginalObject As Object  
    Get  
    Private Set
```

```
public:  
property Object^ OriginalObject {  
    Object^ get ();  
    private: void set (Object^ value);  
}
```

**Value**

The original object that caused the error.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [ErrorContext](#) ► **Path**

C#

Gets the path of the JSON location where the error occurred.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string Path { get; private set; }
```

```
Public Property Path As String  
    Get  
    Private Set
```

```
public:  
property String^ Path {  
    String^ get ();  
    private: void set (String^ value);  
}
```

## ▣ Value

The path of the JSON location where the error occurred.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# ErrorEventArgs Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► **ErrorEventArgs**

Provides data for the Error event.

## Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class ErrorEventArgs : EventArgs
```

```
Public Class ErrorEventArgs _
    Inherits EventArgs
```

```
public ref class ErrorEventArgs : public EventArgs
```

## Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declared <input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">ErrorEventArgs(Object, ErrorContext)</a>	Initializes a new instance of the <b>ErrorEventArgs</b> class.
	<a href="#">CurrentObject</a>	Gets the current object the error event is being raised against.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)

	<a href="#">ErrorContext</a>	Gets the error context.
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

└─ [EventArgs](#)

└─ **ErrorEventArgs**

# ErrorEventArgs

## Constructor (currentObject, errorContext)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [ErrorEventArgs](#) ►

### ErrorEventArgs(Object, ErrorContext)

Initializes a new instance of the [ErrorEventArgs](#) class.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public ErrorEventArgs(  
    Object currentObject,  
    ErrorContext errorContext  
)
```

```
Public Sub New ( _  
    currentObject As Object, _  
    errorContext As ErrorContext _  
)
```

```
public:  
ErrorEventArgs(  
    Object^ currentObject,  
    ErrorContext^ errorContext  
)
```

#### ▣ Parameters

##### **currentObject** ([Object](#))

The current object.

##### **errorContext** ([ErrorContext](#))

The error context.

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [ErrorEventArgs](#) ►

## CurrentObject

Gets the current object the error event is being raised against.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Object CurrentObject { get; private set; }
```

```
Public Property CurrentObject As Object  
    Get  
    Private Set
```

```
public:  
property Object^ CurrentObject {  
    Object^ get ();  
    private: void set (Object^ value);  
}
```

### Value

The current object the error event is being raised against.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [ErrorEventArgs](#) ►

## ErrorContext

Gets the error context.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public ErrorContext ErrorContext { get; private set;
```

```
Public Property ErrorContext As ErrorContext  
    Get  
    Private Set
```

```
public:  
property ErrorContext^ ErrorContext {  
    ErrorContext^ get ();  
    private: void set (ErrorContext^ value);  
}
```

### ▣ Value

The error context.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# Interface

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► **Interface**

Used by [JsonSerializer](#) to resolves a [JsonContract](#) for a given [Type](#). C# ▼

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public interface IContractResolver
```

```
Public Interface IContractResolver
```

```
public interface class IContractResolver
```

## Members

All Members

Methods

Public

Protected

Instance

Declared

Static

Inherited

Icon	Member	Description
	<a href="#">ResolveContract(Type)</a>	Resolves the contract for a given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## (type)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [IContractResolver](#) ►

### ResolveContract(Type)

Resolves the contract for a given type.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
JsonContract ResolveContract(  
    Type type  
)
```

```
Function ResolveContract ( _  
    type As Type _  
) As JsonContract
```

```
JsonContract^ ResolveContract(  
    Type^ type  
)
```

#### ▣ Parameters

**type** ([Type](#))

The type to resolve a contract for.

#### ▣ Return Value

The contract for a given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Interface

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► **ReferenceResolver**

C# ▼

Used to resolve references when serializing and deserializing JSON by the [JsonSerializer](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public interface IReferenceResolver
```

```
Public Interface IReferenceResolver
```

```
public interface class IReferenceResolver
```

### Members

All Members

Methods

Public

Instance

Declared

Protected

Static

Inherited

Icon	Member	Description
	<a href="#">AddReference(Object, String, Object)</a>	Adds a reference to the specified object.
	<a href="#">GetReference(Object, Object)</a>	Gets the reference for the sepecified object.
	<a href="#">IsReferenced(Object, Object)</a>	Determines whether the specified object is referenced.



[ResolveReference\(Object, String\)](#)

Resolves a reference to its object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## AddReference Method (context, reference, value)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [IReferenceResolver](#)

### ► AddReference(Object, String, Object)

Adds a reference to the specified object.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>void AddReference(     <a href="#">Object</a> context,     <a href="#">string</a> reference,     <a href="#">Object</a> value )</pre>	<pre>Sub AddReference (      <i>context</i> As <a href="#">Object</a>,      <i>reference</i> As <a href="#">String</a>,      <i>value</i> As <a href="#">Object</a>  )</pre>	<pre>void AddReference(     <a href="#">Object</a>^ context,     <a href="#">String</a>^ reference,     <a href="#">Object</a>^ value )</pre>

#### ▣ Parameters

**context** ([Object](#))

The serialization context.

**reference** ([String](#))

The reference.

**value** ([Object](#))

The object to reference.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## (context, value)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [IReferenceResolver](#)

### ► GetReference(Object, Object)

Gets the reference for the sepecified object.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>string GetReference(     Object context,     Object value )</pre>		
	<pre>Function GetReference ( _     context As Object, _     value As Object _ ) As String</pre>	
		<pre>String^ GetReference(     Object^ context,     Object^ value )</pre>

#### ▣ Parameters

##### **context** ([Object](#))

The serialization context.

##### **value** ([Object](#))

The object to get a reference for.

#### ▣ Return Value

The reference to the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## Json.NET - Quick Starts & API Documentation **IsReferenced Method** **(context, value)**

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [IReferenceResolver](#)

### ► **IsReferenced(Object, Object)**

Determines whether the specified object is referenced.

C#

#### ▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
bool IsReferenced(  
    Object context,  
    Object value  
)
```

```
Function IsReferenced ( _  
    context As Object, _  
    value As Object _  
) As Boolean
```

```
bool IsReferenced(  
    Object^ context,  
    Object^ value  
)
```

#### ▣ **Parameters**

##### **context (Object)**

The serialization context.

##### **value (Object)**

The object to test for a reference.

#### ▣ **Return Value**

true if the specified object is referenced; otherwise, false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

# ResolveReference

## Method (context, reference)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [IReferenceResolver](#)

### ► ResolveReference(Object, String)

Resolves a reference to its object.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre><code><a href="#">Object</a> ResolveReference(     <a href="#">Object</a> context,     <a href="#">string</a> reference )</code></pre>		
<pre><code>Function ResolveReference (      context As <a href="#">Object</a>,      reference As <a href="#">String</a>  ) As <a href="#">Object</a></code></pre>		
<pre><code><a href="#">Object</a>^ ResolveReference(     <a href="#">Object</a>^ context,     <a href="#">String</a>^ reference )</code></pre>		

#### ▣ Parameters

##### **context** ([Object](#))

The serialization context.

##### **reference** ([String](#))

The reference to resolve.

#### ▣ Return Value

The object that

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

Provides methods to get and set values.

▣ **Declaration Syntax**

C#	Visual Basic	Visual C++
----	--------------	------------

```
public interface IValueProvider
```

```
Public Interface IValueProvider
```

```
public interface class IValueProvider
```

▣ **Members**

All Members	Methods
-------------	---------

- |   |  |   |
|---|--|---|
| <input checked="" type="checkbox"/> Public    | <input checked="" type="checkbox"/> Instance | <input checked="" type="checkbox"/> Declared  |
| <input checked="" type="checkbox"/> Protected | <input checked="" type="checkbox"/> Static   | <input checked="" type="checkbox"/> Inherited |

Icon	Member	Description
	<a href="#">GetValue(Object)</a>	Gets the value.
	<a href="#">SetValue(Object, Object)</a>	Sets the value.

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [IValueProvider](#) ►

## GetValue(Object)

C# ▼

Gets the value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
Object GetValue(  
    Object target  
)
```

```
Function GetValue ( _  
    target As Object _  
) As Object
```

```
Object^ GetValue(  
    Object^ target  
)
```

### ▣ Parameters

**target (Object)**

The target to get the value from.

### ▣ Return Value

The value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## SetValue Method (target, value)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [IValueProvider](#) ►

### SetValue(Object, Object)

Sets the value.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
void SetValue(  
    Object target,  
    Object value  
)
```

```
Sub SetValue ( _  
    target As Object, _  
    value As Object _  
)
```

```
void SetValue(  
    Object^ target,  
    Object^ value  
)
```

#### ▣ Parameters

##### **target** ([Object](#))

The target to set the value on.

##### **value** ([Object](#))

The value to set on the target.

# JsonArrayContract Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► **JsonArrayContract**

C# ▼

Contract details for a [Type](#) used by the [JsonSerializer](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonArrayContract : JsonContainerContract
```

```
Public Class JsonArrayContract _  
    Inherits JsonContainerContract
```

```
public ref class JsonArrayContract : public JsonCont
```

## Members

All Members

Constructors

Methods

Properties

Public  
 Protected

Instance  
 Static

Declare  
 Inherited

Icon	Member	Description
	<a href="#">JsonArrayContract(Type)</a>	Initializes a new instance of the <b>JsonArrayContract</b> class.
	<a href="#">CollectionItemType</a>	Gets the <a href="#">Type</a> of the collection items.
	<a href="#">Converter</a>	Gets or sets the default <a href="#">JsonConverter</a> for this contract.  (Inherited from <a href="#">JsonContract</a> .)

	<a href="#">CreatedType</a>	<p>Gets or sets the type created during deserialization.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">DefaultCreator</a>	<p>Gets or sets the default creator method used to create the object.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">DefaultCreatorNonPublic</a>	<p>Gets or sets a value indicating whether the default creator is non public.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">Equals(Object)</a>	<p>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">Finalize()</a>	<p>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetHashCode()</a>	<p>Serves as a hash function for a particular type.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetType()</a>	<p>Gets the <a href="#">Type</a> of the current instance.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">IsReference</a>	<p>Gets or sets whether this type contract is serialized as a reference.</p>

		(Inherited from <a href="#">JsonContract.</a> )
	<a href="#">ItemConverter</a>	Gets or sets the default collection items <a href="#">JsonConverter</a> .  (Inherited from <a href="#">JsonContainerContract.</a> )
	<a href="#">ItemsReference</a>	Gets or sets a value indicating whether the collection items preserve object references.  (Inherited from <a href="#">JsonContainerContract.</a> )
	<a href="#">ItemReferenceLoopHandling</a>	Gets or sets the collection item reference loop handling.  (Inherited from <a href="#">JsonContainerContract.</a> )
	<a href="#">ItemTypeHandling</a>	Gets or sets the collection item type name handling.  (Inherited from <a href="#">JsonContainerContract.</a> )
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">OnDeserialized</a>	Gets or sets the method called immediately after deserialization of the object.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">OnDeserializing</a>	Gets or sets the method called during deserialization of the object.  (Inherited from <a href="#">JsonContract.</a> )

	<a href="#">OnError</a>	Gets or sets the method called when an error is thrown during the serialization of the object.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">OnSerialized</a>	Gets or sets the method called after serialization of the object graph.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">OnSerializing</a>	Gets or sets the method called before serialization of the object.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">UnderlyingType</a>	Gets the underlying type for the contract.  (Inherited from <a href="#">JsonContract</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

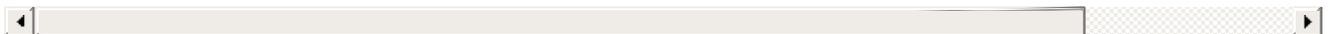
└─ [JsonContract](#)

└─ [JsonContainerContract](#)

└─ **JsonArrayContract**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# JsonArrayContract Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonArrayContract](#) ►

## JsonArrayContract(Type)

Initializes a new instance of the [JsonArrayContract](#) class.

### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonArrayContract(     Type underlyingType )</pre>		
	<pre>Public Sub New ( _     underlyingType As Type _ )</pre>	
		<pre>public: JsonArrayContract(     Type^ underlyingType )</pre>

### Parameters

#### *underlyingType* ([Type](#))

The underlying type for the contract.

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonArrayContract](#) ►

### CollectionItemType

Gets the [Type](#) of the collection items.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Type CollectionItemType { get; private set; }
```

```
Public Property CollectionItemType As Type  
    Get  
    Private Set
```

```
public:  
property Type^ CollectionItemType {  
    Type^ get ();  
    private: void set (Type^ value);  
}
```

#### ▣ Value

The [Type](#) of the collection items.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

### JsonContainerContract

Contract details for a [Type](#) used by the [JsonSerializer](#).

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<code>public class JsonContainerContract : <a href="#">JsonContract</a></code>		

```
Public Class JsonContainerContract _
    Inherits JsonContract
```

```
public ref class JsonContainerContract : public Json
```

#### ▣ Members

All Members	Methods	Properties	
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">Converter</a>	Gets or sets the default <a href="#">JsonConverter</a> for this contract.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">CreatedType</a>	Gets or sets the type created during deserialization.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">DefaultCreator</a>	Gets or sets the default creator

		method used to create the object.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">DefaultCreatorNonPublic</a>	Gets or sets a value indicating whether the default creator is non public.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object.</a> )
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object.</a> )
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object.</a> )
	<a href="#">IsReference</a>	Gets or sets whether this type contract is serialized as a reference.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">ItemConverter</a>	Gets or sets the default collection items <a href="#">JsonConverter</a> .

	<a href="#">ItemsReference</a>	Gets or sets a value indicating whether the collection items preserve object references.
	<a href="#">ItemReferenceLoopHandling</a>	Gets or sets the collection item reference loop handling.
	<a href="#">ItemTypeNameHandling</a>	Gets or sets the collection item type name handling.
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">OnDeserialized</a>	Gets or sets the method called immediately after deserialization of the object.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">OnDeserializing</a>	Gets or sets the method called during deserialization of the object.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">OnError</a>	Gets or sets the method called when an error is thrown during the serialization of the object.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">OnSerialized</a>	Gets or sets the method called after serialization of the object graph.

		(Inherited from <a href="#">JsonContract.</a> )
	<a href="#">OnSerializing</a>	Gets or sets the method called before serialization of the object.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">UnderlyingType</a>	Gets the underlying type for the contract.  (Inherited from <a href="#">JsonContract.</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonContract](#)

└─ **JsonContainerContract**

└─ [JsonArrayContract](#)

└─ [JsonDictionaryContract](#)

└─ [JsonObjectContract](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[JsonContainerContract](#) ► **ItemConverter**

Gets or sets the default collection items [JsonConverter](#).

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonConverter ItemConverter { get; set; }
```

```
Public Property ItemConverter As JsonConverter  
    Get  
    Set
```

```
public:  
property JsonConverter^ ItemConverter {  
    JsonConverter^ get ();  
    void set (JsonConverter^ value);  
}
```

## ▣ Value

The converter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContainerContract](#) ► **ItemIsReference**

Gets or sets a value indicating whether the collection items preserve object references.

C# ▼

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<code>public Nullable&lt;bool&gt; ItemIsReference { get; set; }</code>		

```
Public Property ItemIsReference As Nullable(Of Boolean)
    Get
    Set
```

```
public:
property Nullable<bool> ItemIsReference {
    Nullable<bool> get ();
    void set (Nullable<bool> value);
}
```

### ▣ Value

`true` if collection items preserve object references; otherwise, `false`

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

Documentation **ItemReferenceLoopHandling Property**

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[JsonContainerContract](#) ► **ItemReferenceLoopHandling**

Gets or sets the collection item reference loop handling.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public Nullable<ReferenceLoopHandling> ItemReference
```

```
Public Property ItemReferenceLoopHandling As Nullabl  
    Get  
    Set
```

```
public:  
property Nullable<ReferenceLoopHandling> ItemReferen  
    Nullable<ReferenceLoopHandling> get ();  
    void set (Nullable<ReferenceLoopHandling> va  
}
```

▣ **Value**

The reference loop handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContainerContract](#) ► **ItemTypeNameHandling**

Gets or sets the collection item type name handling.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<code>public <a href="#">Nullable</a>&lt;<a href="#">TypeNameHandling</a>&gt; ItemTypeNameHandli</code>		

```
Public Property ItemTypeNameHandling As Nullable(Of
    Get
    Set
```

```
public:
property Nullable<TypeNameHandling> ItemTypeNameHand
    Nullable<TypeNameHandling> get ();
    void set (Nullable<TypeNameHandling> value);
}
```

### ▣ Value

The type name handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



Contract details for a [Type](#) used by the [JsonSerializer](#).

## ▣ Declaration Syntax

C#      Visual Basic      Visual C++

```
public abstract class JsonContract
```

```
Public MustInherit Class JsonContract
```

```
public ref class JsonContract abstract
```

## ▣ Members

All Members	Methods	Properties	
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declared
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">Converter</a>	Gets or sets the default <a href="#">JsonConverter</a> for this contract.
	<a href="#">CreatedType</a>	Gets or sets the type created during deserialization.
	<a href="#">DefaultCreator</a>	Gets or sets the default creator method used to create the object.
		

	<a href="#">DefaultCreatorNonPublic</a>	Gets or sets a value indicating whether the default creator is non public.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">IsReference</a>	Gets or sets whether this type contract is serialized as a reference.
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">OnDeserialized</a>	Gets or sets the method called immediately after deserialization of the object.

	<a href="#">OnDeserializing</a>	Gets or sets the method called during deserialization of the object.
	<a href="#">OnError</a>	Gets or sets the method called when an error is thrown during the serialization of the object.
	<a href="#">OnSerialized</a>	Gets or sets the method called after serialization of the object graph.
	<a href="#">OnSerializing</a>	Gets or sets the method called before serialization of the object.
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">UnderlyingType</a>	Gets the underlying type for the contract.

## ▣ Inheritance Hierarchy

### [Object](#)

- └─ **JsonContract**
  - └─ [JsonContainerContract](#)
  - └─ [JsonDynamicContract](#)
  - └─ [JsonISerializableContract](#)
  - └─ [JsonLinqContract](#)

## [JsonPrimitiveContract](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#) ►

## Converter

Gets or sets the default [JsonConverter](#) for this contract.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonConverter Converter { get; set; }
```

```
Public Property Converter As JsonConverter  
    Get  
    Set
```

```
public:  
property JsonConverter^ Converter {  
    JsonConverter^ get ();  
    void set (JsonConverter^ value);  
}
```

### ▣ Value

The converter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#) ►

## CreatedType

Gets or sets the type created during deserialization.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Type CreatedType { get; set; }
```

```
Public Property CreatedType As Type  
    Get  
    Set
```

```
public:  
property Type^ CreatedType {  
    Type^ get ();  
    void set (Type^ value);  
}
```

### ▣ Value

The type created during deserialization.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#) ►

## DefaultCreator

Gets or sets the default creator method used to create the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Func DefaultCreator { get; set; }
```

```
Public Property DefaultCreator As Func  
    Get  
    Set
```

```
public:  
property Func^ DefaultCreator {  
    Func^ get ();  
    void set (Func^ value);  
}
```

### ▣ Value

The default creator method used to create the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#) ►

### DefaultCreatorNonPublic

Gets or sets a value indicating whether the default creator is non public.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool DefaultCreatorNonPublic { get; set; }
```

```
Public Property DefaultCreatorNonPublic As Boolean  
    Get  
    Set
```

```
public:  
property bool DefaultCreatorNonPublic {  
    bool get ();  
    void set (bool value);  
}
```

#### ▣ Value

`true` if the default object creator is non-public; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#) ►

## IsReference

Gets or sets whether this type contract is serialized as a reference.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<bool> IsReference { get; set; }
```

```
Public Property IsReference As Nullable(Of Boolean)  
    Get  
    Set
```

```
public:  
property Nullable<bool> IsReference {  
    Nullable<bool> get ();  
    void set (Nullable<bool> value);  
}
```

### Value

Whether this type contract is serialized as a reference.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#) ►

## OnDeserialized

Gets or sets the method called immediately after deserialization of the object.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public MethodInfo OnDeserialized { get; set; }
```

```
Public Property OnDeserialized As MethodInfo  
    Get  
    Set
```

```
public:  
property MethodInfo^ OnDeserialized {  
    MethodInfo^ get ();  
    void set (MethodInfo^ value);  
}
```

### Value

The method called immediately after deserialization of the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#) ►

## OnDeserializing

Gets or sets the method called during deserialization of the object.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public MethodInfo OnDeserializing { get; set; }
```

```
Public Property OnDeserializing As MethodInfo  
    Get  
    Set
```

```
public:  
property MethodInfo^ OnDeserializing {  
    MethodInfo^ get ();  
    void set (MethodInfo^ value);  
}
```

### ▣ Value

The method called during deserialization of the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#) ►

## OnError

Gets or sets the method called when an error is thrown during the serialization of the object.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public MethodInfo OnError { get; set; }
```

```
Public Property OnError As MethodInfo  
    Get  
    Set
```

```
public:  
property MethodInfo^ OnError {  
    MethodInfo^ get ();  
    void set (MethodInfo^ value);  
}
```

### Value

The method called when an error is thrown during the serialization of the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#) ►

## OnSerialized

Gets or sets the method called after serialization of the object graph.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public MethodInfo OnSerialized { get; set; }
```

```
Public Property OnSerialized As MethodInfo  
    Get  
    Set
```

```
public:  
property MethodInfo^ OnSerialized {  
    MethodInfo^ get ();  
    void set (MethodInfo^ value);  
}
```

### ▣ Value

The method called after serialization of the object graph.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#) ►

## OnSerializing

Gets or sets the method called before serialization of the object.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">MethodInfo</a> OnSerializing { <a href="#">get</a>; <a href="#">set</a>; }</pre>		
<pre>Public Property OnSerializing As <a href="#">MethodInfo</a>     Get     Set</pre>		
<pre>public: property <a href="#">MethodInfo</a>^ OnSerializing {     <a href="#">MethodInfo</a>^ <a href="#">get</a> ();     void <a href="#">set</a> (<a href="#">MethodInfo</a>^ <i>value</i>); }</pre>		

### ▣ Value

The method called before serialization of the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#) ►

## UnderlyingType

Gets the underlying type for the contract.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Type UnderlyingType { get; private set; }
```

```
Public Property UnderlyingType As Type  
    Get  
    Private Set
```

```
public:  
property Type^ UnderlyingType {  
    Type^ get ();  
    private: void set (Type^ value);  
}
```

### ▣ Value

The underlying type for the contract.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonDictionaryContract

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

### JsonDictionaryContract

Contract details for a [Type](#) used by the [JsonSerializer](#).

#### Declaration Syntax

C#	Visual Basic	Visual C++
<code>public class JsonDictionaryContract : <a href="#">JsonContainerContract</a></code>		

```
Public Class JsonDictionaryContract _
    Inherits JsonContainerContract
```

```
public ref class JsonDictionaryContract : public JsonContainerContract
```

#### Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declared <input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonDictionaryContract(Type)</a>	Initializes a new instance of the <b>JsonDictionaryContract</b> class.
	<a href="#">Converter</a>	Gets or sets the default <a href="#">JsonConverter</a> for this contract.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">CreatedType</a>	Gets or sets the type created

		during deserialization.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">DefaultCreator</a>	Gets or sets the default creator method used to create the object.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">DefaultCreatorNonPublic</a>	Gets or sets a value indicating whether the default creator is non public.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">DictionaryKeyType</a>	Gets the <a href="#">Type</a> of the dictionary keys.
	<a href="#">DictionaryValueType</a>	Gets the <a href="#">Type</a> of the dictionary values.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object.</a> )
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.

		(Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">IsReference</a>	Gets or sets whether this type contract is serialized as a reference.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">ItemConverter</a>	Gets or sets the default collection items <a href="#">JsonConverter</a> .  (Inherited from <a href="#">JsonContainerContract</a> .)
	<a href="#">ItemIsReference</a>	Gets or sets a value indicating whether the collection items preserve object references.  (Inherited from <a href="#">JsonContainerContract</a> .)
	<a href="#">ItemReferenceLoopHandling</a>	Gets or sets the collection item reference loop handling.  (Inherited from <a href="#">JsonContainerContract</a> .)
	<a href="#">ItemTypeNameHandling</a>	Gets or sets the collection item type name handling.  (Inherited from <a href="#">JsonContainerContract</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)

	<a href="#">OnDeserialized</a>	<p>Gets or sets the method called immediately after deserialization of the object.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">OnDeserializing</a>	<p>Gets or sets the method called during deserialization of the object.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">OnError</a>	<p>Gets or sets the method called when an error is thrown during the serialization of the object.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">OnSerialized</a>	<p>Gets or sets the method called after serialization of the object graph.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">OnSerializing</a>	<p>Gets or sets the method called before serialization of the object.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">PropertyNameResolver</a>	<p>Gets or sets the property name resolver.</p>
	<a href="#">ToString()</a>	<p>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">UnderlyingType</a>	<p>Gets the underlying type for the contract.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>

---

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonContract](#)

└─ [JsonContainerContract](#)

└─ **JsonDictionaryContract**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



# JsonDictionaryContract Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[JsonDictionaryContract](#) ► **JsonDictionaryContract(Type)**

Initializes a new instance of the [JsonDictionaryContract](#) class.

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonDictionaryContract(  
    Type underlyingType  
)
```

```
Public Sub New (   
    underlyingType As Type   
)
```

```
public:  
JsonDictionaryContract(  
    Type^ underlyingType  
)
```

## Parameters

### *underlyingType* ([Type](#))

The underlying type for the contract.

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonDictionaryContract](#) ► **DictionaryKeyType**

Gets the [Type](#) of the dictionary keys.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Type DictionaryKeyType { get; private set; }
```

```
Public Property DictionaryKeyType As Type  
    Get  
    Private Set
```

```
public:  
property Type^ DictionaryKeyType {  
    Type^ get ();  
    private: void set (Type^ value);  
}
```

### ▣ Value

The [Type](#) of the dictionary keys.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonDictionaryContract](#) ► **DictionaryValueType**

Gets the [Type](#) of the dictionary values.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Type DictionaryValueType { get; private set;
```

```
Public Property DictionaryValueType As Type  
    Get  
    Private Set
```

```
public:  
property Type^ DictionaryValueType {  
    Type^ get ();  
    private: void set (Type^ value);  
}
```

### ▣ Value

The [Type](#) of the dictionary values.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonDictionaryContract](#) ► **PropertyNameResolver**

Gets or sets the property name resolver.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Func PropertyNameResolver { get; set; }
```

```
Public Property PropertyNameResolver As Func  
    Get  
    Set
```

```
public:  
property Func^ PropertyNameResolver {  
    Func^ get ();  
    void set (Func^ value);  
}
```

### ▣ Value

The property name resolver.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

### JsonDynamicContract

Contract details for a [Type](#) used by the [JsonSerializer](#).

#### [-] Declaration Syntax

C#	Visual Basic	Visual C++
<code>public class JsonDynamicContract : <a href="#">JsonContract</a></code>		

```
Public Class JsonDynamicContract _
    Inherits JsonContract
```

```
public ref class JsonDynamicContract : public JsonCo
```

#### [-] Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declared <input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonDynamicContract(Type)</a>	Initializes a new instance of the <b>JsonDynamicContract</b> class.
	<a href="#">Converter</a>	Gets or sets the default <a href="#">JsonConverter</a> for this contract.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">CreatedType</a>	Gets or sets the type created during

		<p>deserialization.</p> <p>(Inherited from <a href="#">JsonContract.</a>)</p>
	<a href="#">DefaultCreator</a>	<p>Gets or sets the default creator method used to create the object.</p> <p>(Inherited from <a href="#">JsonContract.</a>)</p>
	<a href="#">DefaultCreatorNonPublic</a>	<p>Gets or sets a value indicating whether the default creator is non public.</p> <p>(Inherited from <a href="#">JsonContract.</a>)</p>
	<a href="#">Equals(Object)</a>	<p>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object.</a>)</p>
	<a href="#">Finalize()</a>	<p>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object.</a>)</p>
	<a href="#">GetHashCode()</a>	<p>Serves as a hash function for a particular type.</p> <p>(Inherited from <a href="#">Object.</a>)</p>
	<a href="#">GetType()</a>	<p>Gets the <a href="#">Type</a> of the current instance.</p> <p>(Inherited from <a href="#">Object.</a>)</p>
	<a href="#">IsReference</a>	<p>Gets or sets whether this type contract is serialized as a reference</p>

		(Inherited from <a href="#">JsonContract.</a> )
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">OnDeserialized</a>	Gets or sets the method called immediately after deserialization of the object.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">OnDeserializing</a>	Gets or sets the method called during deserialization of the object.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">OnError</a>	Gets or sets the method called when an error is thrown during the serialization of the object.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">OnSerialized</a>	Gets or sets the method called after serialization of the object graph.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">OnSerializing</a>	Gets or sets the method called before serialization of the object.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">Properties</a>	Gets the object's properties.
	<a href="#">PropertyNameResolver</a>	Gets or sets the property name resolver.
		

	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">UnderlyingType</a>	Gets the underlying type for the contract.  (Inherited from <a href="#">JsonContract</a> .)

## ▣ Inheritance Hierarchy

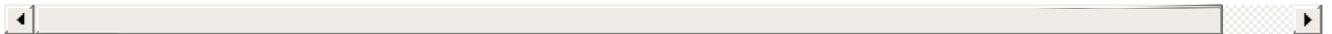
[Object](#)

└─ [JsonContract](#)

└─ **JsonDynamicContract**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



# JsonDynamicContract Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonDynamicContract](#)

## ► JsonDynamicContract(Type)

Initializes a new instance of the [JsonDynamicContract](#) class.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonDynamicContract(     <a href="#">Type</a> <i>underlyingType</i> )</pre>		
	<pre>Public Sub New (      <i>underlyingType</i> As <a href="#">Type</a>  )</pre>	
		<pre>public: JsonDynamicContract(     <a href="#">Type</a>^ <i>underlyingType</i> )</pre>

### ▣ Parameters

#### *underlyingType* ([Type](#))

The underlying type for the contract.

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonDynamicContract](#)► **Properties**

Gets the object's properties.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public JsonPropertyCollection Properties { get; priv
```

```
Public Property Properties As JsonPropertyCollection  
    Get  
    Private Set
```

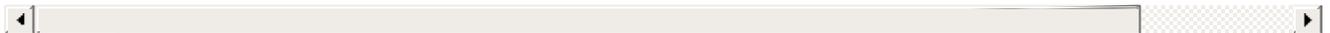
```
public:  
property JsonPropertyCollection^ Properties {  
    JsonPropertyCollection^ get ();  
    private: void set (JsonPropertyCollection^ v  
}
```

▣ **Value**

The object's properties.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonDynamicContract](#)

### ► PropertyNameResolver

Gets or sets the property name resolver.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Func PropertyNameResolver { get; set; }
```

```
Public Property PropertyNameResolver As Func  
    Get  
    Set
```

```
public:  
property Func^ PropertyNameResolver {  
    Func^ get ();  
    void set (Func^ value);  
}
```

#### ▣ Value

The property name resolver.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

### JsonSerializableContract

Contract details for a [Type](#) used by the [JsonSerializer](#).

#### [-] Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class JsonSerializableContract : JsonContract
```

```
Public Class JsonSerializableContract _
    Inherits JsonContract
```

```
public ref class JsonSerializableContract : public
```

#### [-] Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
			<input checked="" type="checkbox"/> Declare
			<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonSerializableContract(Type)</a>	Initializes a new instance of the <b>JsonSerializableContract</b> class.
	<a href="#">Converter</a>	Gets or sets the default <a href="#">JsonConverter</a> for this contract.  (Inherited from <a href="#">JsonContract</a> .)

	<a href="#">CreatedType</a>	<p>Gets or sets the type created during deserialization.</p> <p>(Inherited from <a href="#">JsonContract.</a>)</p>
	<a href="#">DefaultCreator</a>	<p>Gets or sets the default creator method used to create the object.</p> <p>(Inherited from <a href="#">JsonContract.</a>)</p>
	<a href="#">DefaultCreatorNonPublic</a>	<p>Gets or sets a value indicating whether the default creator is non public.</p> <p>(Inherited from <a href="#">JsonContract.</a>)</p>
	<a href="#">Equals(Object)</a>	<p>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object.</a>)</p>
	<a href="#">Finalize()</a>	<p>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object.</a>)</p>
	<a href="#">GetHashCode()</a>	<p>Serves as a hash function for a particular type.</p> <p>(Inherited from <a href="#">Object.</a>)</p>
	<a href="#">GetType()</a>	<p>Gets the <a href="#">Type</a> of the current instance.</p> <p>(Inherited from <a href="#">Object.</a>)</p>
	<a href="#">ISerializableCreator</a>	<p>Gets or sets the ISerializable</p>

		object constructor.
	<a href="#">IsReference</a>	<p>Gets or sets whether this type contract is serialized as a reference.</p> <p>(Inherited from <a href="#">JsonContract.</a>)</p>
	<a href="#">MemberwiseClone()</a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object.</a>)</p>
	<a href="#">OnDeserialized</a>	<p>Gets or sets the method called immediately after deserialization of the object.</p> <p>(Inherited from <a href="#">JsonContract.</a>)</p>
	<a href="#">OnDeserializing</a>	<p>Gets or sets the method called during deserialization of the object.</p> <p>(Inherited from <a href="#">JsonContract.</a>)</p>
	<a href="#">OnError</a>	<p>Gets or sets the method called when an error is thrown during the serialization of the object.</p> <p>(Inherited from <a href="#">JsonContract.</a>)</p>
	<a href="#">OnSerialized</a>	<p>Gets or sets the method called after serialization of the object graph.</p> <p>(Inherited from <a href="#">JsonContract.</a>)</p>
	<a href="#">OnSerializing</a>	<p>Gets or sets the method called before serialization of the object.</p>

		(Inherited from <a href="#">JsonContract.</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">UnderlyingType</a>	Gets the underlying type for the contract.  (Inherited from <a href="#">JsonContract.</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonContract](#)

└─ **JsonSerializableContract**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



# JsonSerializableContract Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[JsonSerializableContract](#) ► **JsonSerializableContract(Type)**

Initializes a new instance of the [JsonSerializableContract](#) class.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSerializableContract(  
    Type underlyingType  
)
```

```
Public Sub New (   
    underlyingType As Type   
)
```

```
public:  
JsonSerializableContract(  
    Type^ underlyingType  
)
```

## Parameters

### *underlyingType* ([Type](#))

The underlying type for the contract.

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonISerializableContract](#) ► **ISerializableCreator**

C#

Gets or sets the ISerializable object constructor.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public ObjectConstructor<Object> ISerializableCreato
```

```
Public Property ISerializableCreator As ObjectConstr
    Get
    Set
```

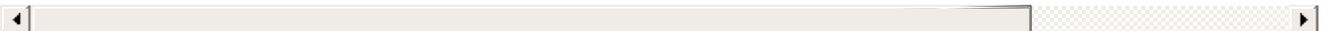
```
public:
property ObjectConstructor<Object^>^ ISerializableCr
    ObjectConstructor<Object^>^ get ();
    void set (ObjectConstructor<Object^>^ value)
}
```

### Value

The ISerializable object constructor.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# JsonLinqContract Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► **JsonLinqContract**

Contract details for a [Type](#) used by the [JsonSerializer](#).

## ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class JsonLinqContract : JsonContract
```

```
Public Class JsonLinqContract _  
    Inherits JsonContract
```

```
public ref class JsonLinqContract : public JsonContr
```

## ▣ Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declare <input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonLinqContract(Type)</a>	Initializes a new instance of the <b>JsonLinqContract</b> class.
	<a href="#">Converter</a>	Gets or sets the default <a href="#">JsonConverter</a> for this contract.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">CreatedType</a>	Gets or sets the type created during deserialization.  (Inherited from <a href="#">JsonContract</a> .)

	<a href="#">DefaultCreator</a>	<p>Gets or sets the default creator method used to create the object.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">DefaultCreatorNonPublic</a>	<p>Gets or sets a value indicating whether the default creator is non public.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">Equals(Object)</a>	<p>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a></p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">Finalize()</a>	<p>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetHashCode()</a>	<p>Serves as a hash function for a particular type.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetType()</a>	<p>Gets the <a href="#">Type</a> of the current instance</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">IsReference</a>	<p>Gets or sets whether this type contract is serialized as a reference.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">MemberwiseClone()</a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">OnDeserialized</a>	<p>Gets or sets the method called</p>

		immediately after deserialization of the object.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">OnDeserializing</a>	Gets or sets the method called during deserialization of the object.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">OnError</a>	Gets or sets the method called when an error is thrown during the serialization of the object.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">OnSerialized</a>	Gets or sets the method called after serialization of the object graph.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">OnSerializing</a>	Gets or sets the method called before serialization of the object.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )
	<a href="#">UnderlyingType</a>	Gets the underlying type for the contract.  (Inherited from <a href="#">JsonContract.</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonContract](#)

    └─ **JsonLinqContract**

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

# JsonLinqContract Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonLinqContract](#) ►

## JsonLinqContract(Type)

Initializes a new instance of the [JsonLinqContract](#) class.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonLinqContract(  
    Type underlyingType  
)
```

```
Public Sub New (   
    underlyingType As Type   
)
```

```
public:  
JsonLinqContract(  
    Type^ underlyingType  
)
```

### ▣ Parameters

#### *underlyingType* ([Type](#))

The underlying type for the contract.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonObjectContract

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► **JsonObjectContract**

Contract details for a [Type](#) used by the [JsonSerializer](#).

### Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class JsonObjectContract : JsonContainerContract
```

```
Public Class JsonObjectContract _
    Inherits JsonContainerContract
```

```
public ref class JsonObjectContract : public JsonCon
```

### Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
			<input checked="" type="checkbox"/> Declare
			<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonObjectContract(Type)</a>	Initializes a new instance of the <b>JsonObjectContract</b> class.
	<a href="#">ConstructorParameters</a>	Gets the constructor parameters required for any non-default constructor
	<a href="#">Converter</a>	Gets or sets the default

		<p><a href="#">JsonConverter</a> for this contract.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">CreatedType</a>	<p>Gets or sets the type created during deserialization.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">DefaultCreator</a>	<p>Gets or sets the default creator method used to create the object.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">DefaultCreatorNonPublic</a>	<p>Gets or sets a value indicating whether the default creator is non public.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">Equals(Object)</a>	<p>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">Finalize()</a>	<p>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetHashCode()</a>	<p>Serves as a hash function for a particular type.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetType()</a>	<p>Gets the <a href="#">Type</a> of the current instance.</p>

		(Inherited from <a href="#">Object</a> .)
	<a href="#">IsReference</a>	Gets or sets whether this type contract is serialized as a reference.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">ItemConverter</a>	Gets or sets the default collection items <a href="#">JsonConverter</a> .  (Inherited from <a href="#">JsonContainerContract</a> .)
	<a href="#">ItemsReference</a>	Gets or sets a value indicating whether the collection items preserve object references.  (Inherited from <a href="#">JsonContainerContract</a> .)
	<a href="#">ItemReferenceLoopHandling</a>	Gets or sets the collection item reference loop handling.  (Inherited from <a href="#">JsonContainerContract</a> .)
	<a href="#">ItemRequired</a>	Gets or sets a value that indicates whether the object's properties are required.
	<a href="#">ItemTypeNameHandling</a>	Gets or sets the collection item type name handling.  (Inherited from <a href="#">JsonContainerContract</a> .)
	<a href="#">MemberSerialization</a>	Gets or sets the object member serialization.

	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">OnDeserialized</a>	Gets or sets the method called immediately after deserialization of the object.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">OnDeserializing</a>	Gets or sets the method called during deserialization of the object.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">OnError</a>	Gets or sets the method called when an error is thrown during the serialization of the object.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">OnSerialized</a>	Gets or sets the method called after serialization of the object graph.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">OnSerializing</a>	Gets or sets the method called before serialization of the object.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">OverrideConstructor</a>	Gets or sets the override constructor used to create the object. This is set when a constructor is marked up using the <code>JsonConstructor</code> attribute.
	<a href="#">ParametrizedConstructor</a>	Gets or sets the parametrized

		constructor used to create the object.
	<a href="#">Properties</a>	Gets the object's properties.
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">UnderlyingType</a>	Gets the underlying type for the contract.  (Inherited from <a href="#">JsonContract</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

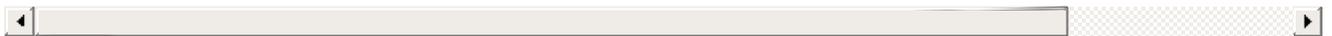
└─ [JsonContract](#)

└─ [JsonContainerContract](#)

└─ **JsonObjectContract**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# JsonObjectContract Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonObjectContract](#)

## ► JsonObjectContract(Type)

Initializes a new instance of the [JsonObjectContract](#) class.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonObjectContract(     Type underlyingType )</pre>		
	<pre>Public Sub New ( _     underlyingType As Type _ )</pre>	
		<pre>public: JsonObjectContract(     Type^ underlyingType )</pre>

### ▣ Parameters

#### *underlyingType* ([Type](#))

The underlying type for the contract.

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonObjectContract](#)

### ► ConstructorParameters

Gets the constructor parameters required for any non-default constructo

C#

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonPropertyCollection ConstructorParameters
```

```
Public Property ConstructorParameters As JsonPropertyCollection  
    Get  
    Private Set
```

```
public:  
property JsonPropertyCollection^ ConstructorParameters  
    JsonPropertyCollection^ get ();  
private: void set (JsonPropertyCollection^ v  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonObjectContract](#)

## ► ItemRequired

Gets or sets a value that indicates whether the object's properties are required.

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<Required> ItemRequired { get; set; }
```

```
Public Property ItemRequired As Nullable(Of Required)  
    Get  
    Set
```

```
public:  
property Nullable<Required> ItemRequired {  
    Nullable<Required> get ();  
    void set (Nullable<Required> value);  
}
```

### ▣ Value

A value indicating whether the object's properties are required.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonObjectContract](#)

### ► MemberSerialization

Gets or sets the object member serialization.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public MemberSerialization MemberSerialization { get
```

```
Public Property MemberSerialization As MemberSeriali  
    Get  
    Set
```

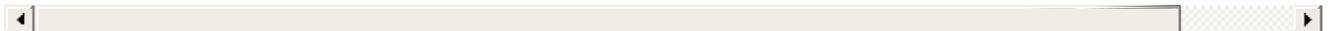
```
public:  
property MemberSerialization MemberSerialization {  
    MemberSerialization get ();  
    void set (MemberSerialization value);  
}
```

#### ▣ Value

The member object serialization.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonObjectContract](#)

### ► OverrideConstructor

Gets or sets the override constructor used to create the object. This is used when a constructor is marked up using the `JsonConstructor` attribute.

C#

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public ConstructorInfo OverrideConstructor { get; set; }
```

```
Public Property OverrideConstructor As ConstructorInfo  
    Get  
    Set
```

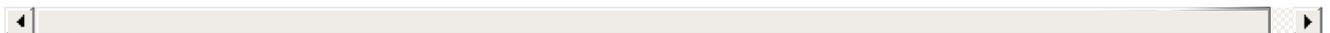
```
public:  
property ConstructorInfo^ OverrideConstructor {  
    ConstructorInfo^ get ();  
    void set (ConstructorInfo^ value);  
}
```

#### ▣ Value

The override constructor.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# ParametrizedConstructor Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonObjectContract](#)

## ► ParametrizedConstructor

Gets or sets the parametrized constructor used to create the object.

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public ConstructorInfo ParametrizedConstructor { get
```

```
Public Property ParametrizedConstructor As Construct  
    Get  
    Set
```

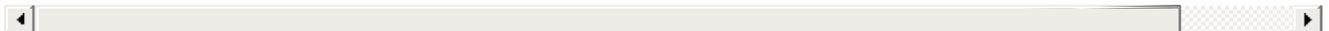
```
public:  
property ConstructorInfo^ ParametrizedConstructor {  
    ConstructorInfo^ get ();  
    void set (ConstructorInfo^ value);  
}
```

### ▣ Value

The parametrized constructor.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonObjectContract](#)► **Properties**

Gets the object's properties.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public JsonPropertyCollection Properties { get; priv
```

```
Public Property Properties As JsonPropertyCollection  
    Get  
    Private Set
```

```
public:  
property JsonPropertyCollection^ Properties {  
    JsonPropertyCollection^ get ();  
    private: void set (JsonPropertyCollection^ v  
}
```

▣ **Value**

The object's properties.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

### JsonPrimitiveContract

Contract details for a [Type](#) used by the [JsonSerializer](#).

#### [-] Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class JsonPrimitiveContract : JsonContract
```

```
Public Class JsonPrimitiveContract _
    Inherits JsonContract
```

```
public ref class JsonPrimitiveContract : public Json
```

#### [-] Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
			<input checked="" type="checkbox"/> Declared
			<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonPrimitiveContract(Type)</a>	Initializes a new instance of the <b>JsonPrimitiveContract</b> class.
	<a href="#">Converter</a>	Gets or sets the default <a href="#">JsonConverter</a> for this contract.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">CreatedType</a>	Gets or sets the type created during

		<p>deserialization.</p> <p>(Inherited from <a href="#">JsonContract.</a>)</p>
	<a href="#">DefaultCreator</a>	<p>Gets or sets the default creator method used to create the object.</p> <p>(Inherited from <a href="#">JsonContract.</a>)</p>
	<a href="#">DefaultCreatorNonPublic</a>	<p>Gets or sets a value indicating whether the default creator is non public.</p> <p>(Inherited from <a href="#">JsonContract.</a>)</p>
	<a href="#">Equals(Object)</a>	<p>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object.</a>)</p>
	<a href="#">Finalize()</a>	<p>Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object.</a>)</p>
	<a href="#">GetHashCode()</a>	<p>Serves as a hash function for a particular type.</p> <p>(Inherited from <a href="#">Object.</a>)</p>
	<a href="#">GetType()</a>	<p>Gets the <a href="#">Type</a> of the current instance.</p> <p>(Inherited from <a href="#">Object.</a>)</p>
	<a href="#">IsReference</a>	<p>Gets or sets whether this type contract is serialized as a reference.</p> <p>(Inherited from <a href="#">JsonContract.</a>)</p>

	<a href="#">MemberwiseClone()</a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">OnDeserialized</a>	<p>Gets or sets the method called immediately after deserialization of the object.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">OnDeserializing</a>	<p>Gets or sets the method called during deserialization of the object.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">OnError</a>	<p>Gets or sets the method called when an error is thrown during the serialization of the object.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">OnSerialized</a>	<p>Gets or sets the method called after serialization of the object graph.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">OnSerializing</a>	<p>Gets or sets the method called before serialization of the object.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>
	<a href="#">ToString()</a>	<p>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">UnderlyingType</a>	<p>Gets the underlying type for the contract.</p> <p>(Inherited from <a href="#">JsonContract</a>.)</p>

## ▣ Inheritance Hierarchy

[Object](#)

└─ [JsonContract](#)

└─ **JsonPrimitiveContract**

└─ [JsonStringContract](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



# JsonPrimitiveContract Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonPrimitiveContract](#)

## ► JsonPrimitiveContract(Type)

Initializes a new instance of the [JsonPrimitiveContract](#) class.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonPrimitiveContract(     <a href="#">Type</a> <i>underlyingType</i> )</pre>		
	<pre>Public Sub New (      <i>underlyingType</i> As <a href="#">Type</a>  )</pre>	
		<pre>public: JsonPrimitiveContract(     <a href="#">Type</a>^ <i>underlyingType</i> )</pre>

### ▣ Parameters

#### *underlyingType* ([Type](#))

The underlying type for the contract.

# JsonProperty Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► **JsonProperty**

Maps a JSON property to a .NET member or constructor parameter.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonProperty
```

```
Public Class JsonProperty
```

```
public ref class JsonProperty
```

## ▣ Members

All Members

Constructors

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declared
- Inherited

Icon	Member	Description
	<a href="#">JsonProperty()</a>	Initializes a new instance of the <b>JsonProperty</b> class
	<a href="#">Converter</a>	Gets or sets the <a href="#">JsonConverter</a> for the property. If set this converter takes precedence over the contract converter for the property type.
	<a href="#">DeclaringType</a>	Gets or sets the type that declared this property.

	<a href="#">DefaultValue</a>	Gets the default value.
	<a href="#">DefaultValueHandling</a>	Gets the property default value handling.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetIsSpecified</a>	Gets or sets a predicate used to determine whether the property should be serialized.
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">Ignored</a>	Gets a value indicating whether this <b>JsonProperty</b> is ignored.

	<a href="#">IsReference</a>	Gets a value indicating whether this property preserves object references.
	<a href="#">ItemConverter</a>	Gets or sets the converter used when serializing the property's collection items.
	<a href="#">ItemIsReference</a>	Gets or sets whether this property's collection items are serialized as a reference.
	<a href="#">ItemReferenceLoopHandling</a>	Gets or sets the the reference loop handling used when serializing the property's collection items.
	<a href="#">ItemTypeNameHandling</a>	Gets or sets the the type name handling used when serializing the property's collection items.
	<a href="#">MemberConverter</a>	Gets the member converter.
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">NullValueHandling</a>	Gets the property null value

		handling.
	<a href="#">ObjectCreationHandling</a>	Gets the property object creation handling.
	<a href="#">Order</a>	Gets or sets the order of serialization and deserialization of a member.
	<a href="#">PropertyName</a>	Gets or sets the name of the property.
	<a href="#">PropertyType</a>	Gets or sets the type of the property.
	<a href="#">Readable</a>	Gets a value indicating whether this <b>JsonProperty</b> is readable.
	<a href="#">ReferenceLoopHandling</a>	Gets the property reference loop handling.
	<a href="#">Required</a>	Gets a value indicating whether this <b>JsonProperty</b> is required.
	<a href="#">SetIsSpecified</a>	Gets or sets an action used to set whether the property has been

		deserialized.
	<a href="#">ShouldSerialize</a>	Gets or sets a predicate used to determine whether the property should be serialize.
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the instance.  (Overrides <a href="#">Object.ToString()</a> .)
	<a href="#">TypeNameHandling</a>	Gets or sets the type name handling.
	<a href="#">UnderlyingName</a>	Gets or sets the name of the underlying member or parameter.
	<a href="#">ValueProvider</a>	Gets the <a href="#">IValueProvider</a> that will get and set the <b>JsonProperty</b> during serialization.
	<a href="#">Writable</a>	Gets a value indicating whether this <b>JsonProperty</b> is writable.

## ▣ Inheritance Hierarchy

[Object](#)

└─ **JsonProperty**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

### JsonProperty()

Initializes a new instance of the [JsonProperty](#) class

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonProperty()
```

```
Public Sub New
```

```
public:  
JsonProperty()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## Converter

Gets or sets the [JsonConverter](#) for the property. If set this converter takes precedence over the contract converter for the property type.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonConverter Converter { get; set; }
```

```
Public Property Converter As JsonConverter  
    Get  
    Set
```

```
public:  
property JsonConverter^ Converter {  
    JsonConverter^ get ();  
    void set (JsonConverter^ value);  
}
```

### Value

The converter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## DeclaringType

Gets or sets the type that declared this property.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Type DeclaringType { get; set; }
```

```
Public Property DeclaringType As Type  
    Get  
    Set
```

```
public:  
property Type^ DeclaringType {  
    Type^ get ();  
    void set (Type^ value);  
}
```

### ▣ Value

The type that declared this property.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## DefaultValue

Gets the default value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Object DefaultValue { get; set; }
```

```
Public Property DefaultValue As Object  
    Get  
    Set
```

```
public:  
property Object^ DefaultValue {  
    Object^ get ();  
    void set (Object^ value);  
}
```

### ▣ Value

The default value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

### DefaultValueHandling

Gets the property default value handling.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<DefaultValueHandling> DefaultValueHa
```

```
Public Property DefaultValueHandling As Nullable(Of  
    Get  
    Set
```

```
public:  
property Nullable<DefaultValueHandling> DefaultValue  
    Nullable<DefaultValueHandling> get ();  
    void set (Nullable<DefaultValueHandling> val  
}
```

#### Value

The default value handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## GetIsSpecified

Gets or sets a predicate used to determine whether the property should be serialized.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Predicate<Object> GetIsSpecified { get; set;
```

```
Public Property GetIsSpecified As Predicate(Of Objec  
    Get  
    Set
```

```
public:  
property Predicate<Object>^ GetIsSpecified {  
    Predicate<Object>^ get ();  
    void set (Predicate<Object>^ value);  
}
```

### Value

A predicate used to determine whether the property should be serialized

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## Ignored

Gets a value indicating whether this [JsonProperty](#) is ignored.

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Ignored { get; set; }
```

```
Public Property Ignored As Boolean  
    Get  
    Set
```

```
public:  
property bool Ignored {  
    bool get ();  
    void set (bool value);  
}
```

### ▣ Value

true if ignored; otherwise, false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## IsReference

Gets a value indicating whether this property preserves object references.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<bool> IsReference { get; set; }
```

```
Public Property IsReference As Nullable(Of Boolean)  
    Get  
    Set
```

```
public:  
property Nullable<bool> IsReference {  
    Nullable<bool> get ();  
    void set (Nullable<bool> value);  
}
```

### Value

`true` if this instance is reference; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## ItemConverter

Gets or sets the converter used when serializing the property's collection items.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonConverter ItemConverter { get; set; }
```

```
Public Property ItemConverter As JsonConverter  
    Get  
    Set
```

```
public:  
property JsonConverter^ ItemConverter {  
    JsonConverter^ get ();  
    void set (JsonConverter^ value);  
}
```

### Value

The collection's items converter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## ItemIsReference

Gets or sets whether this property's collection items are serialized as a reference.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<bool> ItemIsReference { get; set; }
```

```
Public Property ItemIsReference As Nullable(Of Boolean)  
    Get  
    Set
```

```
public:  
property Nullable<bool> ItemIsReference {  
    Nullable<bool> get ();  
    void set (Nullable<bool> value);  
}
```

### Value

Whether this property's collection items are serialized as a reference.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## ItemReferenceLoopHandling

C#

Gets or sets the the reference loop handling used when serializing the property's collection items.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<ReferenceLoopHandling> ItemReference
```

```
Public Property ItemReferenceLoopHandling As Nullabl
    Get
    Set
```

```
public:
property Nullable<ReferenceLoopHandling> ItemReferen
    Nullable<ReferenceLoopHandling> get ();
    void set (Nullable<ReferenceLoopHandling> va
}

```

### Value

The collection's items reference loop handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



# ItemTypeNameHandling Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## ItemTypeNameHandling

Gets or sets the the type name handling used when serializing the property's collection items.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<TypeNameHandling> ItemTypeNameHandli
```

```
Public Property ItemTypeNameHandling As Nullable(Of  
    Get  
    Set
```

```
public:  
property Nullable<TypeNameHandling> ItemTypeNameHand  
    Nullable<TypeNameHandling> get ();  
    void set (Nullable<TypeNameHandling> value);  
}
```

### Value

The collection's items type name handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

### MemberConverter

Gets the member converter.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonConverter MemberConverter { get; set; }
```

```
Public Property MemberConverter As JsonConverter  
    Get  
    Set
```

```
public:  
property JsonConverter^ MemberConverter {  
    JsonConverter^ get ();  
    void set (JsonConverter^ value);  
}
```

#### ▣ Value

The member converter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

### NullValueHandling

Gets the property null value handling.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<NullValueHandling> NullValueHandling
```

```
Public Property NullValueHandling As Nullable(Of Nul  
    Get  
    Set
```

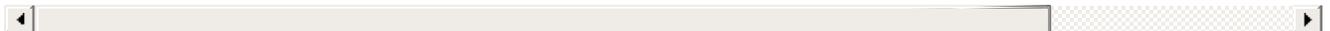
```
public:  
property Nullable<NullValueHandling> NullValueHandli  
    Nullable<NullValueHandling> get ();  
    void set (Nullable<NullValueHandling> value)  
}
```

#### ▣ Value

The null value handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



**Property**[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►**ObjectCreationHandling**

Gets the property object creation handling.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public Nullable<ObjectCreationHandling> ObjectCrea
```

```
Public Property ObjectCreationHandling As Nullable(0  
    Get  
    Set
```

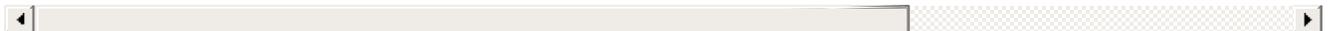
```
public:  
property Nullable<ObjectCreationHandling> ObjectCrea  
    Nullable<ObjectCreationHandling> get ();  
    void set (Nullable<ObjectCreationHandling> v  
}
```

▣ **Value**

The object creation handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## Order

C#

Gets or sets the order of serialization and deserialization of a member.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<int> Order { get; set; }
```

```
Public Property Order As Nullable(Of Integer)  
    Get  
    Set
```

```
public:  
property Nullable<int> Order {  
    Nullable<int> get ();  
    void set (Nullable<int> value);  
}
```

### Value

The numeric order of serialization or deserialization.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►**PropertyName**

Gets or sets the name of the property.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public string PropertyName { get; set; }
```

```
Public Property PropertyName As String  
    Get  
    Set
```

```
public:  
property String^ PropertyName {  
    String^ get ();  
    void set (String^ value);  
}
```

**Value**

The name of the property.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►**PropertyType**

Gets or sets the type of the property.

**Declaration Syntax**

C#

Visual Basic

Visual C++

```
public Type PropertyType { get; set; }
```

```
Public Property PropertyType As Type  
    Get  
    Set
```

```
public:  
property Type^ PropertyType {  
    Type^ get ();  
    void set (Type^ value);  
}
```

**Value**

The type of the property.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## Readable

Gets a value indicating whether this [JsonProperty](#) is readable.

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Readable { get; set; }
```

```
Public Property Readable As Boolean  
    Get  
    Set
```

```
public:  
property bool Readable {  
    bool get ();  
    void set (bool value);  
}
```

### ▣ Value

`true` if readable; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# ReferenceLoopHandling Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## ReferenceLoopHandling

Gets the property reference loop handling.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<ReferenceLoopHandling> ReferenceLoop
```

```
Public Property ReferenceLoopHandling As Nullable(Of  
    Get  
    Set
```

```
public:  
property Nullable<ReferenceLoopHandling> ReferenceLo  
    Nullable<ReferenceLoopHandling> get ();  
    void set (Nullable<ReferenceLoopHandling> va  
}
```

### Value

The reference loop handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## Required

Gets a value indicating whether this [JsonProperty](#) is required.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">Required</a> Required { <a href="#">get</a>; <a href="#">set</a>; }</pre>		
<pre>Public Property Required As <a href="#">Required</a>     Get     Set</pre>		
<pre>public: property <a href="#">Required</a> Required {     <a href="#">Required</a> <a href="#">get</a> ();     void <a href="#">set</a> (<a href="#">Required</a> <i>value</i>); }</pre>		

### ▣ Value

A value indicating whether this [JsonProperty](#) is required.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## SetIsSpecified

Gets or sets an action used to set whether the property has been deserialized.

### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public Action SetIsSpecified { get; set; }</pre>		
	<pre>Public Property SetIsSpecified As Action     Get     Set</pre>	
		<pre>public: property Action^ SetIsSpecified {     Action^ get ();     void set (Action^ value); }</pre>

### Value

An action used to set whether the property has been deserialized.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## ShouldSerialize

Gets or sets a predicate used to determine whether the property should be serialize.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Predicate<Object> ShouldSerialize { get; set;
```

```
Public Property ShouldSerialize As Predicate(Of Object)  
    Get  
    Set
```

```
public:  
property Predicate<Object> ShouldSerialize {  
    Predicate<Object> get ();  
    void set (Predicate<Object> value);  
}
```

### Value

A predicate used to determine whether the property should be serialize.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## ToString()

Returns a [String](#) that represents this instance.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public override string ToString()
```

```
Public Overrides Function ToString As String
```

```
public:  
virtual String^ ToString() override
```

### ▣ Return Value

A [String](#) that represents this instance.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

### TypeNameHandling

Gets or sets the type name handling.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<TypeNameHandling> TypeNameHandling {
```

```
    Public Property TypeNameHandling As Nullable(Of Type)  
        Get  
        Set
```

```
public:  
property Nullable<TypeNameHandling> TypeNameHandling  
    Nullable<TypeNameHandling> get ();  
    void set (Nullable<TypeNameHandling> value);  
}
```

#### ▣ Value

The type name handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)



## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

### UnderlyingName

Gets or sets the name of the underlying member or parameter.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public string UnderlyingName { get; set; }
```

```
Public Property UnderlyingName As String  
    Get  
    Set
```

```
public:  
property String^ UnderlyingName {  
    String^ get ();  
    void set (String^ value);  
}
```

#### ▣ Value

The name of the underlying member or parameter.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## ValueProvider

Gets the [IValueProvider](#) that will get and set the [JsonProperty](#) during serialization.

### Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public <a href="#">IValueProvider</a> ValueProvider { <a href="#">get</a>; <a href="#">set</a>; }</pre>		
<pre>Public Property ValueProvider As <a href="#">IValueProvider</a>     Get     Set</pre>		
<pre>public: property <a href="#">IValueProvider</a>^ ValueProvider {     <a href="#">IValueProvider</a>^ <a href="#">get</a> ();     void <a href="#">set</a> (<a href="#">IValueProvider</a>^ <i>value</i>); }</pre>		

### Value

The [IValueProvider](#) that will get and set the [JsonProperty](#) during serialization.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

## Writable

Gets a value indicating whether this [JsonProperty](#) is writable.

C#

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public bool Writable { get; set; }
```

```
Public Property Writable As Boolean  
    Get  
    Set
```

```
public:  
property bool Writable {  
    bool get ();  
    void set (bool value);  
}
```

### ▣ Value

`true` if writable; otherwise, `false`.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# JsonPropertyCollection

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

### JsonPropertyCollection

A collection of [JsonProperty](#) objects.

#### Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class JsonPropertyCollection : KeyedCollection
```

```
Public Class JsonPropertyCollection _
    Inherits KeyedCollection(Of String, JsonProp
```

```
public ref class JsonPropertyCollection : public Key
```

#### Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Static
			<input checked="" type="checkbox"/> Declare
			<input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	<a href="#">JsonPropertyCollection(Type)</a>	Initializes a new instance of the <b>JsonPropertyCollection</b> class.
	<a href="#">Add(T)</a>	Adds an object to the end of the <a href="#">Collection&lt;T&gt;</a> .  (Inherited from <a href="#">Collection&lt;JsonProperty&gt;</a> .)

	<a href="#">AddProperty(JsonProperty)</a>	Adds a <a href="#">JsonProperty</a> object.
	<a href="#">ChangeItemKey(TItem, TKey)</a>	Changes the key associated with the specified element in the lookup dictionary.  (Inherited from <a href="#">KeyedCollection&lt;String, JsonProperty&gt;</a> .)
	<a href="#">Clear()</a>	Removes all elements from the <a href="#">Collection&lt;T&gt;</a> .  (Inherited from <a href="#">Collection&lt;JsonProperty&gt;</a> .)
	<a href="#">ClearItems()</a>	Removes all elements from the <a href="#">KeyedCollection&lt;TKey, TItem&gt;</a> .  (Inherited from <a href="#">KeyedCollection&lt;String, JsonProperty&gt;</a> .)
	<a href="#">Comparer</a>	Gets the generic equality comparer that is used to determine equality of keys in the collection.  (Inherited from <a href="#">KeyedCollection&lt;String, JsonProperty&gt;</a> .)
	<a href="#">Contains(TKey)</a>	Determines whether the collection contains an element with the specified key.  (Inherited from <a href="#">KeyedCollection&lt;String, JsonProperty&gt;</a> .)
		

	<a href="#">Contains(T)</a>	Determines whether an element exists in the <a href="#">Collection&lt;T&gt;</a> .  (Inherited from <a href="#">Collection&lt;JsonProperty&gt;</a> .)
	<a href="#">CopyTo(T[], Int32)</a>	Copies the entire <a href="#">Collection&lt;T&gt;</a> to a <a href="#">T[]</a> compatible one-dimensional array, starting at the specified index of the target array.  (Inherited from <a href="#">Collection&lt;JsonProperty&gt;</a> .)
	<a href="#">Count</a>	Gets the number of elements contained in the <a href="#">Collection&lt;T&gt;</a> .  (Inherited from <a href="#">Collection&lt;JsonProperty&gt;</a> .)
	<a href="#">Dictionary</a>	Gets the lookup dictionary of the <a href="#">Collection&lt;T, TItem&gt;</a> .  (Inherited from <a href="#">KeyedCollection&lt;String, JsonProperty&gt;</a> .)
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetClosestMatchProperty(String)</a>	Gets the closest matching <a href="#">JsonProperty</a> object. First attempt

		get an exact case match of <code>propertyName</code> and then a case insensitive match.
	<a href="#">GetEnumerator()</a>	Returns an enumerator that iterates through the <a href="#">Collection&lt;T&gt;</a> .  (Inherited from <a href="#">Collection&lt;JsonProperty&gt;</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetKeyForItem(JsonProperty)</a>	When implemented in a derived class, extracts the key from the specified element.  (Overrides <a href="#">KeyedCollection&lt;String, JsonProperty&gt;.GetKeyForItem</a> .)
	<a href="#">GetProperty(String, StringComparison)</a>	Gets a property by property name.
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">IndexOf(T)</a>	Searches for the specified object and returns the zero-based index of its first occurrence within the entire <a href="#">Collection&lt;T&gt;</a> .  (Inherited from <a href="#">Collection&lt;JsonProperty&gt;</a> .)
	<a href="#">Insert(Int32, T)</a>	Inserts an element into the <a href="#">Collection&lt;T&gt;</a> at the specified index.

		(Inherited from <a href="#">Collection&lt;JsonProperty&gt;</a> .)
	<a href="#">InsertItem(Int32, TItem)</a>	<p>Inserts an element into the <a href="#">KeyedCollection&lt;TKey, TItem&gt;</a> specified index.</p> <p>(Inherited from <a href="#">KeyedCollection&lt;String, JsonProperty&gt;</a>.)</p>
	<a href="#">Item[TKey]</a>	<p>Gets the element with the specified key.</p> <p>(Inherited from <a href="#">KeyedCollection&lt;String, JsonProperty&gt;</a>.)</p>
	<a href="#">Item[Int32]</a>	<p>Gets or sets the element at the specified index.</p> <p>(Inherited from <a href="#">Collection&lt;JsonProperty&gt;</a>.)</p>
	<a href="#">Items</a>	<p>Gets a <a href="#">IList&lt;T&gt;</a> wrapper around <a href="#">Collection&lt;T&gt;</a>.</p> <p>(Inherited from <a href="#">Collection&lt;JsonProperty&gt;</a>.)</p>
	<a href="#">MemberwiseClone()</a>	<p>Creates a shallow copy of the <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">Remove(TKey)</a>	<p>Removes the element with the specified key from the <a href="#">KeyedCollection&lt;TKey, TItem&gt;</a>.</p> <p>(Inherited from <a href="#">KeyedCollection&lt;String, JsonProperty&gt;</a>.)</p>

		<a href="#">JsonProperty&gt;.</a> )
	<a href="#">Remove(T)</a>	Removes the first occurrence of a specific object from the <a href="#">Collection</a> .  (Inherited from <a href="#">Collection&lt;JsonProperty&gt;.</a> )
	<a href="#">RemoveAt(Int32)</a>	Removes the element at the specified index of the <a href="#">Collection&lt;T&gt;</a> .  (Inherited from <a href="#">Collection&lt;JsonProperty&gt;.</a> )
	<a href="#">RemoveItem(Int32)</a>	Removes the element at the specified index of the <a href="#">KeyedCollection&lt;TItem&gt;</a> .  (Inherited from <a href="#">KeyedCollection&lt;String, JsonProperty&gt;.</a> )
	<a href="#">SetItem(Int32, TItem)</a>	Replaces the item at the specified index with the specified item.  (Inherited from <a href="#">KeyedCollection&lt;String, JsonProperty&gt;.</a> )
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object.</a> )

## ▣ Inheritance Hierarchy

[Object](#)

└─ [Collection<JsonProperty>](#)

└─ [KeyedCollection<String, JsonProperty>](#)

└─ **JsonPropertyCollection**

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



# JsonPropertyCollection Constructor (type)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[JsonPropertyCollection](#) ► **JsonPropertyCollection(Type)**

Initializes a new instance of the [JsonPropertyCollection](#) class.

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonPropertyCollection(  
    Type type  
)
```

```
Public Sub New ( _  
    type As Type _  
)
```

```
public:  
JsonPropertyCollection(  
    Type^ type  
)
```

## Parameters

**type (Type)**

The type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

# AddProperty Method (property)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[JsonPropertyCollection](#) ► **AddProperty(JsonProperty)**

Adds a [JsonProperty](#) object.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void AddProperty(  
    JsonProperty property  
)
```

```
Public Sub AddProperty ( _  
    property As JsonProperty _  
)
```

```
public:  
void AddProperty(  
    JsonProperty^ property  
)
```

## ▣ Parameters

### *property* ([JsonProperty](#))

The property to add to the collection.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)

Documentation **GetClosestMatchProperty Method**  
**(propertyName)**[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►[JsonPropertyCollection](#) ► **GetClosestMatchProperty(String)**

C# ▼

Gets the closest matching [JsonProperty](#) object. First attempts to get an exact case match of propertyName and then a case insensitive match.

▣ **Declaration Syntax**

C#

Visual Basic

Visual C++

```
public JsonProperty GetClosestMatchProperty(  
    string propertyName  
)
```

```
Public Function GetClosestMatchProperty ( _  
    propertyName As String _  
) As JsonProperty
```

```
public:  
JsonProperty^ GetClosestMatchProperty(  
    String^ propertyName  
)
```

▣ **Parameters****propertyName (String)**

Name of the property.

▣ **Return Value**

A matching property if found.

## (item)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[JsonPropertyCollection](#) ► **GetKeyForItem(JsonProperty)**

When implemented in a derived class, extracts the key from the specified element.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
protected override string GetKeyForItem(  
    JsonProperty item  
)
```

```
Protected Overrides Function GetKeyForItem ( _  
    item As JsonProperty _  
) As String
```

```
protected:  
virtual String^ GetKeyForItem(  
    JsonProperty^ item  
) override
```

### ▣ Parameters

**item** ([JsonProperty](#))

The element from which to extract the key.

### ▣ Return Value

The key for the specified element.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## GetProperty Method (propertyName, comparisonType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[JsonPropertyCollection](#) ► **GetProperty(String, StringComparison)**

Gets a property by property name.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonProperty GetProperty(  
    string propertyName,  
    StringComparison comparisonType  
)
```

```
Public Function GetProperty ( _  
    propertyName As String, _  
    comparisonType As StringComparison _  
) As JsonProperty
```

```
public:  
JsonProperty^ GetProperty(  
    String^ propertyName,  
    StringComparison comparisonType  
)
```

### ▣ Parameters

**propertyName** ([String](#))

The name of the property to get.

**comparisonType** ([StringComparison](#))

Type property name string comparison.

### ▣ Return Value

A matching property if found.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.C  
(4.5.6.14930)

# JsonStringContract

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► **JsonStringContract**

Contract details for a [Type](#) used by the [JsonSerializer](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonStringContract : JsonPrimitiveContr
```

```
Public Class JsonStringContract _  
    Inherits JsonPrimitiveContract
```

```
public ref class JsonStringContract : public JsonPri
```

### Members

All Members

Constructors

Methods

Properties

- Public
- Protected

- Instance
- Static

- Declare
- Inherited

Icon	Member	Description
	<a href="#">JsonStringContract(Type)</a>	Initializes a new instance of the <b>JsonStringContract</b> class.
	<a href="#">Converter</a>	Gets or sets the default <a href="#">JsonConverter</a> for this contract.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">CreatedType</a>	Gets or sets the type created during deserialization.

		(Inherited from <a href="#">JsonContract.</a> )
	<a href="#">DefaultCreator</a>	Gets or sets the default creator method used to create the object.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">DefaultCreatorNonPublic</a>	Gets or sets a value indicating whether the default creator is non public.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>  (Inherited from <a href="#">Object.</a> )
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object.</a> )
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object.</a> )
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object.</a> )
	<a href="#">IsReference</a>	Gets or sets whether this type contract is serialized as a reference.  (Inherited from <a href="#">JsonContract.</a> )
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .

		(Inherited from <a href="#">Object</a> .)
	<a href="#">OnDeserialized</a>	Gets or sets the method called immediately after deserialization of the object.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">OnDeserializing</a>	Gets or sets the method called during deserialization of the object.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">OnError</a>	Gets or sets the method called when an error is thrown during the serialization of the object.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">OnSerialized</a>	Gets or sets the method called after serialization of the object graph.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">OnSerializing</a>	Gets or sets the method called before serialization of the object.  (Inherited from <a href="#">JsonContract</a> .)
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">UnderlyingType</a>	Gets the underlying type for the contract.  (Inherited from <a href="#">JsonContract</a> .)

## ▣ Inheritance Hierarchy

[Object](#)

[L \*\*JsonContract\*\*](#)

[L \*\*JsonPrimitiveContract\*\*](#)

[L \*\*JsonStringContract\*\*](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



# JsonStringContract Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonStringContract](#) ►

## JsonStringContract(Type)

Initializes a new instance of the [JsonStringContract](#) class.

### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public JsonStringContract(     <a href="#">Type</a> <i>underlyingType</i> )</pre>		
	<pre>Public Sub New (      <i>underlyingType</i> As <a href="#">Type</a>  )</pre>	
		<pre>public: JsonStringContract(     <a href="#">Type</a>^ <i>underlyingType</i> )</pre>

### ▣ Parameters

#### *underlyingType* ([Type](#))

The underlying type for the contract.

## Delegate

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

### ObjectConstructor<T>

Represents a method that constructs an object.

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public delegate Object ObjectConstructor<T>(
    params Object[] args
)
```

```
Public Delegate Function ObjectConstructor(Of T) ( _
    ParamArray args As Object() _
) As Object
```

```
generic<typename T>
public delegate Object^ ObjectConstructor(
    ... array<Object^>^ args
)
```

#### ▣ Generic Template Parameters

*T*

**[Missing <typeparam name="T"/> documentation for "T:Newtonsoft.Json.Serialization.ObjectConstructor`1"]**

#### ▣ Parameters

*args* ([Object](#)[])

# OnErrorAttribute Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► **OnErrorAttribute**

C#

When applied to a method, specifies that the method is called when an error occurs serializing an object.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public sealed class OnErrorAttribute : Attribute
```

```
Public NotInheritable Class OnErrorAttribute _
    Inherits Attribute
```

```
public ref class OnErrorAttribute sealed : public At
```

## Members

All Members

Constructors

Methods

Properties

Public  
 Protected

Instance  
 Static

Declared  
 Inherited

Icon	Member	Description
	<a href="#">OnErrorAttribute()</a>	Initializes a new instance of the <b>OnErrorAttribute</b> class
	<a href="#">Equals(Object)</a>	Returns a value that indicates whether this instance is equal to a specified object.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup

		<p>operations before the <a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">GetHashCode()</a>	<p>Returns the hash code for this instance.</p> <p>(Inherited from <a href="#">Attribute</a>.)</p>
	<a href="#">GetType()</a>	<p>Gets the <a href="#">Type</a> of the current instance</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">IsDefaultAttribute()</a>	<p>When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.</p> <p>(Inherited from <a href="#">Attribute</a>.)</p>
	<a href="#">Match(Object)</a>	<p>When overridden in a derived class, returns a value that indicates whether this instance equals a specified object.</p> <p>(Inherited from <a href="#">Attribute</a>.)</p>
	<a href="#">MemberwiseClone()</a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">ToString()</a>	<p>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#">TypeId</a>	<p>When implemented in a derived class gets a unique identifier for this <a href="#">Attribute</a>.</p>

(Inherited from [Attribute](#).)

## ▣ Inheritance Hierarchy

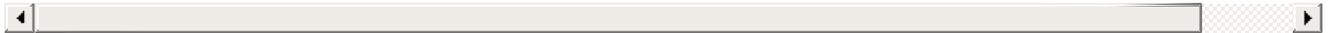
[Object](#)

└─ [Attribute](#)

└─ **OnErrorAttribute**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [OnErrorAttribute](#) ►

### OnErrorAttribute()

Initializes a new instance of the [OnErrorAttribute](#) class

#### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public OnErrorAttribute()
```

```
Public Sub New
```

```
public:  
OnErrorAttribute()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

### ReflectionValueProvider

Get and set values for a [MemberInfo](#) using reflection.

#### ▣ Declaration Syntax

C#	Visual Basic	Visual C++
----	--------------	------------

```
public class ReflectionValueProvider : IValueProvide
```

```
Public Class ReflectionValueProvider _
    Implements IValueProvider
```

```
public ref class ReflectionValueProvider : IValuePro
```

#### ▣ Members

All Members	Constructors	Methods	
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance	<input checked="" type="checkbox"/> Declare
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherite

Icon	Member	Description
	<a href="#">ReflectionValueProvider(MemberInfo)</a>	Initializes a new instance of the <b>ReflectionValueProvider</b> class.
	<a href="#">Equals(Object)</a>	Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a> .

		(Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetHashCode()</a>	Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">GetValue(Object)</a>	Gets the value.
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">SetValue(Object, Object)</a>	Sets the value.
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)

## ▣ Inheritance Hierarchy

## [Object](#)

### └ ReflectionValueProvider

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0  
(4.5.6.14930)



# ReflectionValueProvider Constructor (memberInfo)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►

[ReflectionValueProvider](#) ► **ReflectionValueProvider(MemberInfo)**

Initializes a new instance of the [ReflectionValueProvider](#) class.

## Declaration Syntax

C#	Visual Basic	Visual C++
<pre>public ReflectionValueProvider(     <a href="#">MemberInfo</a> memberInfo )</pre>		
	<pre>Public Sub New (      <i>memberInfo</i> As <a href="#">MemberInfo</a>  )</pre>	
		<pre>public: ReflectionValueProvider(     <a href="#">MemberInfo</a>^ <i>memberInfo</i> )</pre>

## Parameters

***memberInfo*** ([MemberInfo](#))

The member info.

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [ReflectionValueProvider](#) ► **GetValue(Object)**

Gets the value.

## ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public Object GetValue(  
    Object target  
)
```

```
Public Function GetValue ( _  
    target As Object _  
) As Object
```

```
public:  
virtual Object^ GetValue(  
    Object^ target  
) sealed
```

## ▣ Parameters

### **target** ([Object](#))

The target to get the value from.

## ▣ Return Value

The value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.5.0.0 (4.5.6.14930)

## SetValue Method (target, value)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [ReflectionValueProvider](#) ► **SetValue(Object, Object)**

Sets the value.

### ▣ Declaration Syntax

C#

Visual Basic

Visual C++

```
public void SetValue(  
    Object target,  
    Object value  
)
```

```
Public Sub SetValue ( _  
    target As Object, _  
    value As Object _  
)
```

```
public:  
virtual void SetValue(  
    Object^ target,  
    Object^ value  
) sealed
```

### ▣ Parameters

#### **target** ([Object](#))

The target to set the value on.

#### **value** ([Object](#))

The value to set on the target.

## Namespaces

C#

## [-] Namespaces

Namespace	Description
<a href="#">Newtonsoft.Json</a>	The <b>Newtonsoft.Json</b> namespace provides classes that are used to implement the core services of the framework.
<a href="#">Newtonsoft.Json.Bson</a>	The <b>Newtonsoft.Json.Linq</b> namespace provides classes that are used to implement BSON.
<a href="#">Newtonsoft.Json.Converters</a>	The <b>Newtonsoft.Json.Converters</b> namespace provides classes that inherit from <a href="#">JsonConverter</a> .
<a href="#">Newtonsoft.Json.Linq</a>	The <b>Newtonsoft.Json.Linq</b> namespace provides classes that are used to implement LINQ to JSON.
<a href="#">Newtonsoft.Json.Schema</a>	The <b>Newtonsoft.Json.Schema</b> namespace provides classes that are used to implement JSON schema.
<a href="#">Newtonsoft.Json.Serialization</a>	The <b>Newtonsoft.Json.Linq</b> namespace provides classes that are used when serializing and

deserializing JSON.

---