

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.



PayPal  
Donate

## 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
```



## 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 JavaScriptDateHandler());
serializer.NullValueHandling = NullValueHandling.Ignore;

using (StreamWriter sw = new StreamWriter(@"c:\temp\test.json"))
using (JsonWriter writer = new JsonTextWriter(sw))
{
    serializer.Serialize(writer, product);
    // {"Expiry":new Date(1230375600000),"Price":100}
}
```

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.

### ReferenceLoopHandling

Controls how circular referencing objects are serialized. Error, ignore or serialize.

### MissingMemberHandling

Controls how missing members (e.g. JSON contains a property that isn't a member on the object) are handled during deserialization. Ignore or error.

#### **NullValueHandling**

Controls how null values are handled during serialization and deserialization. Include or ignore.

#### **DefaultValueHandling**

Controls whether a value will be written to JSON or not if it matches the value specified in the member's DefaultValueAttribute. Include or ignore.

#### **ObjectCreationHandling**

Controls how objects are created during deserialization. Auto, reuse, replace.

#### **TypeNameHandling**

Controls whether .NET type names are included in serialized JSON and read during deserialization when creating objects. None, Objects, Arrays or All.

#### **ConstructorHandling**

Controls how constructors are used when initializing objects during deserialization. Default or AllowNonPublicDefaultConstructor.

#### **Converters**

A collection of JsonConverters that will be used during serialization and deserialization.

## **JsonConverters**

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.

To create your own custom converter inherit from the `JsonConverter` class. Json.NET also comes with a number of `JsonConverters`:

#### **DateTime JSON Converters**

Json.NET comes with a number of `JsonConverters` for serializing and deserializing `DateTimes`. Read more about dates and Json.NET [here](#).

#### **XmlNodeConverter**

Converts an `XmlNode` to and from JSON. Note that to convert a JSON object it must have only a single property or you must define a root node name to be inserted when using this converter. This is required because properties are converted into nodes and well formed XML can only have one root node. `XmlNodeConverter` has an option to insert a root node for you.

#### **BinaryConverter**

Converts binary data like the `SqlBinary` object to JSON. The binary data is written as a string in JSON and is encoded in Base64.

#### **CustomCreationConverter**

An abstract `JsonConverter` for customizing how an object is created during deserialization. Inherit from this class and implement the `Create` method with your own code to create and return an object. The object will then be populated with JSON values by the serializer.

A possible example of using this converter would be to call out to a dependency injection framework to resolve what object should be created.

---

## Customizing JSON serialization with attributes

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.
- [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 [DataContract](#) and [DataMember](#) 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]
    [JsonConverter(typeof(IsoDateTimeConverter))]
    public DateTime BirthDate { get; set; }

    // new Date(976918263055)
    [JsonProperty]
```

```
[JsonConverter(typeof(JavaScriptDateTimeCo
public DateTime LastModified { get; set; }

// not serialized
public string Department { get; set; }
}
```

### **JsonObjectAttribute**

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

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

### **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.

### **JsonIgnoreAttribute**

Excludes a field or property from serialization.

### **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 JsonConverter 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 JsonConverter will always be used to serialize that value.

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

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

This example shows the JsonConverterAttribute being applied to a property.

```
DateTime date = Convert.ToDateTime("1970-01-01");
MemberConverterClass c = new MemberConverterClass()
{
    DefaultConverter = date,
    MemberConverter = date
};

string json = JsonConvert.SerializeObject(c);
Console.WriteLine(json);
//{
```

```
//  "DefaultConverter": "/Date(0)/",
//  "MemberConverter": "1970-01-01T00:00:00Z
//}
```



# 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](#) for dates is the same one used by Microsoft: "VDate(1198908717056)V". You can read more about it [here](#).

## 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."
    };
}
```

```
string defaultJson = JsonConvert.SerializeObject(  
    // {"Details":"Application started."}, "LogD  
  
string javascriptJson = JsonConvert.SerializeObject(  
    // {"Details":"Application started."}, "LogD  
  
string isoJson = JsonConvert.SerializeObject(  
    // {"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.

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

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

One final thing to note is all date values returned by Json.NET are in [UTC time](#).

---

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.DeserializeObject<List<Product>>(json);

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 = JsonConvert.DeserializeObject<Dictionary<string, string>>(json);

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

Console.WriteLine(values["key1"]);
// value1
```

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 deserialized.
    public int Member1 { get; set; }

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

    // This field is not serialized. The OnDeserialized
    // attribute is used to set the member value after
    // it has been deserialized.
    [JsonIgnore]
    public string Member3 { get; set; }

    // This field is set to null, but populated
    // by the OnDeserialized attribute.
    public string Member4 { get; set; }
```

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

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

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

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

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

---

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

```
SerializationEventTestObject obj = new SerializationEventTestObject();

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 data
//   "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<SerializationEventTestObject>(json);

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.
```



## 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.DeserializeObject<List<DateTime>>(
    """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 serialized 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)
{
    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"  
//}
```



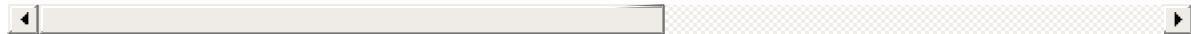
## Serialization and Preserving Object References

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 = JsonConvert.DeserializeObject<List<Person>>(json, 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 JsonConverter 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.DeserializeObject<List<IPerson>>(json);

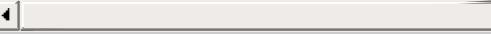
IPerson person = people[0];

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"
//  ]
//}
```



# 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 will 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 { NullValueHandling
```

```
// {
//   "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 _startsWithChar;  
    public DynamicContractResolver(char starti  
    {  
        _startsWithChar = 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.SerializeObject(
    new JsonSerializerSettings { ContractResol

// {
//     "AuthorName": "Brandon Sanderson",
//     "AuthorAge": 34,
```

```
//    "AuthorCountry": "United States of Amer
// }

string startingWithB = JsonConvert.SerializeObject(
    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.NET 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"",
            ""unescapeUrl"": ""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 Hilt
            ""content"": ""[1] In 2006, she rele
        },
        {
    ]}
```

```
    "GsearchResultClass": "GwebSearch",
    "unesapedUrl": "http://www.imdb.com/title/tt0000000",
    "url": "http://www.imdb.com/name/nm0000000",
    "visibleUrl": "www.imdb.com",
    "cacheUrl": "http://www.google.com/search?q=Paris+Hilton",
    "title": "<b>Paris Hilton</b>",
    "titleNoFormatting": "Paris Hilton",
    "content": "Self: Zoolander. Society: None. Profession: Model, Socialite, Entrepreneur. Nationality: American. Date of Birth: March 19, 1983. Height: 5 feet 7 inches. Weight: 110 pounds. Eye Color: Blue. Hair Color: Brown. Education: Attended the University of California, Los Angeles (UCLA) for a year before dropping out. Career: Started modeling at age 14. Known for her work with Victoria's Secret and Sports Illustrated Swimsuit Issue. Also known for her role in the TV show 'The Simple Life' and her own reality show 'Paris Hilton: My Life Story'. Personal Life: Married to Carter Oosterhouse from 2004 to 2007. Engaged to Nick Lachey from 2008 to 2010. Currently engaged to Michael Jackson's son Prince Jackson. Children: One son, Bronx, born in 2011. Family: Daughter of Conrad Hilton and Kathy Hilton. Granddaughter of the founder of Hilton Hotels. Religious Beliefs: Christian. Political Beliefs: Liberal. Hobbies: Shopping, Traveling, Photography, Fashion, Music, Dancing. Interests: Politics, Current Events, Social Media, Celebrities, Sports. Social Media Presence: Instagram (@paris_hilton), Twitter (@paris_hilton), Facebook (@paris_hilton). Other: Known for her lavish lifestyle and controversial behavior. Controversies include her stay at the Ritz-Carlton in New York City, her appearance on 'The Simple Life', and her legal troubles related to a car accident and a dog attack. She has also faced criticism for her comments about race and ethnicity on social media. Despite these controversies, she remains a popular figure in the public eye." },
  ],
  "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.google.com/search?q=Paris+Hilton"
  }
}
```

```
        },
        ""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 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 JProperty("channel",
```

```

new JObject(
    new JProperty("title", "James Newton
    new JProperty("link", "http://james.
    new JProperty("description", "James
    new JProperty("item",
        new JArray(
            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 JArray(
                        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": "Annoucing 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 CodePlex

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("Manufacturers[0].Name")
```

### 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"": """",
    ""Products"": [
        {
            ""Name"": """",
            ""Price"": 99.95
        },
        {
            ""Name"": """",
            ""Price"": 4
        }
    ]
}
"]);

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

decimal productPrice = (decimal)o.SelectToken("Price");
// 50

string productName = (string)o.SelectToken("Name");
// 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["Manufacturers"]
// null
// Headlight Fluid

decimal totalPrice = o["Manufacturers"].Sum(
// 149.95
```



---

# Reading and Writing 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 JsonTextWriter(sw))
{
    jsonWriter.Formatting = Formatting.Indented;

    jsonWriter.WriteStartObject();
    jsonWriter.WritePropertyName("CPU");
    jsonWriter.WriteLine("Intel");
    jsonWriter.WritePropertyName("PSU");
    jsonWriter.WriteLine("500W");
    jsonWriter.WritePropertyName("Drives");
    jsonWriter.WriteStartArray();
    jsonWriter.WriteLine("DVD read/writer");
    jsonWriter.WriteComment("(broken)");
    jsonWriter.WriteLine("500 gigabyte hard disk");
    jsonWriter.WriteLine("200 gigabyte hard disk");
    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 @.
- 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 ?.
- Charater data, comments, whitespace and significate whitespace nodes are accessed via #cdata-section, #comment, #whitespace and #significate-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"" standalone=""yes""?>
<root>
    <person id=""1"">
        <name>John Doe</name>
        <age>30</age>
        <city>New York</city>
    </person>
</root>"
```

```
<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.SerializeXmlNodeNoFormatting(doc);
//{
//  "?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"": """",
                ""name"": """",
                ""url"": """http://www.google.com"""
            },
            {
                ""@id"": """",
                ""name"": """",
                ""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++

```
namespace Newtonsoft.Json
```

```
Namespace Newtonsoft.Json
```

```
namespace Newtonsoft.Json
```

### Types

All Types

Classes

Interfaces

Enumerations

Icon	Type	Description
	<a href="#">ConstructorHandling</a>	Specifies how constructors are used when initializing objects during deserialization by the <a href="#">JsonSerializer</a> .
	<a href="#">DefaultValueHandling</a>	Specifies default value handling options for the <a href="#">JsonSerializer</a> .
	<a href="#">Formatting</a>	Specifies formatting options for the <a href="#">JsonTextWriter</a> .
	<a href="#">IJsonLineInfo</a>	Provides an interface to enable a

		class to return line and position information.
	<a href="#">JsonArrayAttribute</a>	Instructs the <a href="#">JsonSerializer</a> how to serialize the collection.
	<a href="#">JsonConstructorAttribute</a>	Instructs the <a href="#">JsonSerializer</a> not to serialize the public field or public read/write property value.
	<a href="#">JsonContainerAttribute</a>	Instructs the <a href="#">JsonSerializer</a> how to serialize the object.
	<a href="#">JsonConvert</a>	Provides methods for converting between common language runtime types and JSON types.
	<a href="#">JsonConverter</a>	Converts an object to and from JSON.
	<a href="#">JsonConverterAttribute</a>	Instructs the <a href="#">JsonSerializer</a> to use the specified <a href="#">JsonConverter</a> when serializing the member or class.
	<a href="#">JsonConverterCollection</a>	Represents a collection of <a href="#">JsonConverter</a> .

	<a href="#"><b>JsonIgnoreAttribute</b></a>	Instructs the <a href="#">JsonSerializer</a> not to serialize the public field or public read/write property value.
	<a href="#"><b>JsonObjectAttribute</b></a>	Instructs the <a href="#">JsonSerializer</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 serialized Json data.
	<a href="#"><b>JsonTextWriter</b></a>	Represents a writer that provides a fast, non-cached, forward-only way of generating Json data.
	<a href="#"><b>JsonToken</b></a>	Specifies the type of Json token.
	<a href="#"><b>JsonValidatingReader</b></a>	Represents a reader that provides <a href="#">JsonSchema</a> validation.
	<a href="#"><b>JsonWriter</b></a>	Represents a writer that provides a fast, non-cached, forward-only way of generating Json data.
	<a href="#"><b>JsonWriterException</b></a>	The exception thrown when an error occurs while reading Json text.
	<a href="#"><b>MemberSerialization</b></a>	Specifies the member serialization options for the <a href="#">JsonSerializer</a> .

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

[JsonWriter.](#)

---

## Enumeration

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

C#

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.
<b>AllowNonPublicDefaultConstructor</b>	Allow Json.NET to use a non-public default constructor.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# DefaultValueHandling

## Enumeration

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

C#

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

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum DefaultValueHandling
```

```
Public Enumeration DefaultValueHandling
```

```
public enum class DefaultValueHandling
```

### Members

Member	Description
<b>Include</b>	Include default values when serializing and deserializing objects.
<b>Ignore</b>	Ignore default values when serializing and deserializing objects.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)

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

Public  
 Protected

Instance  
 Static

Declared  
 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.

(4.0.2.13623)

C# ▾

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

## 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: 4.0.2.C  
(4.0.2.13623)

# LineNumber Property

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

C# ▾

Gets the current line number.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
int LineNumber { get; }
```

```
ReadOnly Property LineNumber As Integer
    Get
```

```
property int LineNumber {
    int get ();
}
```

## 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: 4.0.2.0  
(4.0.2.13623)

# LinePosition Property

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

C# ▾

Gets the current line position.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
int LinePosition { get; }
```

```
ReadOnly Property LinePosition As Integer
    Get
```

```
property int LinePosition {
    int get ();
}
```

## 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: 4.0.2.0  
(4.0.2.13623)

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"/> 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="#">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>	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 reference data.  (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](#)

    └ **JsonArrayAttribute**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# JsonArrayAttribute

## Constructor

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

**JsonArrayAttribute()**

C# ▾

### 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: 4.0.2.0 (4.0.2.13623)

# JsonArrayAttribute

## Constructor

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

**JsonArrayAttribute()**

C# ▾

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

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonArrayAttribute()
```

```
Public Sub New
```

```
public:  
    JsonArrayAttribute()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# JsonArrayAttribute

## Constructor (allowNullItems)

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

**JsonArrayAttribute(Boolean)**

C#

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonArrayAttribute

## Constructor (id)

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

**JsonArrayAttribute(String)**

C#

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

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonArrayAttribute(  
    string id  
)
```

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

```
public:  
JsonArrayAttribute(  
    String^ id  
)
```

### Parameters

*id* ([String](#))

The container Id.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

## AllowNullItems

C#

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

### 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: 4.0.2.0  
(4.0.2.13623)

# JsonConstructorAttribute Class

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

C#

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 JsonConstructorAttribute : Attribute
```

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

```
public ref class JsonConstructorAttribute sealed : 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"/> Declare <input checked="" type="checkbox"/> Inherit

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 <a href="#">Attribute</a> .

(Inherited from [Attribute](#).)

## **Inheritance Hierarchy**

[Object](#)

  └ [Attribute](#)

    └ **JsonConstructorAttribute**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# JsonConstructorAttribute Constructor

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

C#

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonConstructorAttribute()
```

```
Public Sub New
```

```
public:  
JsonConstructorAttribute()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonContainerAttribute Class

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

C# ▾

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

Public

Protected

Instance

Static

Declare

Inherite

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 reference data.
	<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](#)

      └ [JsonObjectAttribute](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



# JsonContainerAttribute Constructor

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

C# ▾

## ▀ 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: 4.0.2.0 (4.0.2.13623)

# JsonContainerAttribute Constructor

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

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# JsonContainerAttribute

## Constructor (id)

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

**JsonContainerAttribute(String)**

C#

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

### 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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# Description Property

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

C# ▾

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: 4.0.2.0  
(4.0.2.13623)



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: 4.0.2.0  
(4.0.2.13623)

# IsReference Property

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

C#

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

## 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: 4.0.2.C  
(4.0.2.13623)

C# ▾

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);
}
```

## Value

The title.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

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"/> 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="#">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, JsonConverter[])</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, JsonConverter[])</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="#">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="#">DeserializeXNode(String)</a>	Deserializes the <a href="#">XNode</a> from a JSON string.
 <a href="#">DeserializeXNode(String, String)</a>	Deserializes the <a href="#">XNode</a> from a JSON string nested in a root element.
 <a href="#">DeserializeXNode(String, String, Boolean)</a>	Deserializes the <a href="#">XNode</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="#">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, JsonConverter[])</a>	Serializes the specified object to a JSON string using a collection of <a href="#">JsonConverter</a> .

= 	<a href="#">SerializeObject(Object, Formatting, JsonConverter[])</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="#">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="#">SerializeXNode(XObject)</a>	Serializes the <a href="#">XNode</a> to JSON string.
= 	<a href="#">SerializeXNode(XObject, Formatting)</a>	Serializes the <a href="#">XNode</a> to JSON string.
= 	<a href="#">SerializeXNode(XObject, Formatting, Boolean)</a>	Serializes the <a href="#">XNode</a> to JSON string.

=  <a href="#">ToString(DateTime)</a>	Converts the <a href="#">DateTime</a> to its JSON string representation.
=  <a href="#">ToString(DateTimeOffset)</a>	Converts the <a href="#">DateTimeOffset</a> to its JSON string representation.
=  <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.

= ♡ S	<a href="#">ToString(UInt16)</a>	Converts the <a href="#">UInt16</a> to its JSON string representation.
= ♡ S	<a href="#">ToString(UInt32)</a>	Converts the <a href="#">UInt32</a> to its JSON string representation.
= ♡ S	<a href="#">ToString(Int64)</a>	Converts the <a href="#">Int64</a> to its JSON string representation.
= ♡ S	<a href="#">ToString(UInt64)</a>	Converts the <a href="#">UInt64</a> to its JSON string representation.
= ♡ S	<a href="#">ToString(Single)</a>	Converts the <a href="#">Single</a> to its JSON string representation.
= ♡ S	<a href="#">ToString(Double)</a>	Converts the <a href="#">Double</a> to its JSON string representation.
= ♡ S	<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(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="#"><u>Undefined</u></a>	Represents JavaScript's undefined as a string. This field is read-only.
--	----------------------------------	--

## **[-] Inheritance Hierarchy**

[Object](#)

└ **JsonConvert**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# Documentation DeserializeAnonymousType<T> Method (value, anonymousTypeObject)

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

C# ▾

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)**

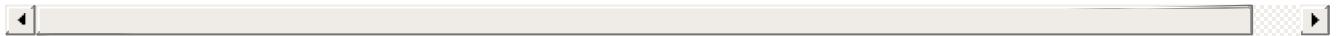
The anonymous type object.

**▀ Return Value**

The deserialized anonymous type from the JSON string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# DeserializeObject Method

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

**DeserializeObject()**

C# ▾

## Members

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

	<u>Type</u> , <u>JsonSerializerSettings</u> )	specified .NET type.
--	--	----------------------

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# DeserializeObject<T>

## Method (value)

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

**DeserializeObject<T>(String)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# 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: 4.0.2.0  
(4.0.2.13623)

# DeserializeObject<T>

## Method (value, converters)

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

**DeserializeObject<T>(String, JsonConverter[])**

C# ▾

Deserializes the JSON to the specified .NET type.

### Declaration Syntax

C#

Visual Basic

Visual C++

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

```
Public Shared Function DeserializeObject(Of T) ( _
    value As String, _
    ParamArray converters As JsonConverter() _
) 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: 4.0.2.C  
(4.0.2.13623)

# DeserializeObject<T>

## Method (value, settings)

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

**DeserializeObject<T>(String, JsonSerializerSettings)**

C# ▾

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 used.

## Return Value

The deserialized object from the JSON string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# DeserializeObject Method (value, settings)

[Namespaces](#) ► [Newtonsoft.Json](#) ►  [JsonConvert](#) ► **DeserializeObject(String, JsonSerializerSettings)**

C# ▾

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 used.

## Return Value

The deserialized object from the JSON string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# 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: 4.0.2.C  
(4.0.2.13623)

# DeserializeObject Method (value, type, converters)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **DeserializeObject(String, Type, JsonConverter[])**

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: 4.0.2.C  
(4.0.2.13623)

# DeserializeObject Method (value, type, settings)

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

**DeserializeObject(String, Type, JsonSerializerSettings)**

C# ▾

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 used.

## Return Value

The deserialized object from the JSON string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# DeserializeXmlNode

## Method

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

**DeserializeXmlNode()**

C# ▾

### Members

Icon	Member	Description
= S	<a href="#">DeserializeXmlNode(String)</a>	Deserializes the XmlNode from a JSON string.
= S	<a href="#">DeserializeXmlNode(String, String)</a>	Deserializes the XmlNode from a JSON string nested in a root element
= S	<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: 4.0.2.C  
(4.0.2.13623)

# DeserializeXmlNode

## Method (value)

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

**DeserializeXmlNode(String)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# DeserializeXmlNode

## Method (value, deserializeRootElementName)

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

**DeserializeXmlNode(String, String)**

C#

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
)
```

```
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: 4.0.2.C  
(4.0.2.13623)

# DeserializeXmlNode

## Method (value, deserializeRootElementName, writeArrayAttribute)

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

**DeserializeXmlNode(String, String, Boolean)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

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

C# ▾

**Members**

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

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# DeserializeXNode Method (value)

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

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++

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

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

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

## Parameters

### **value (String)**

The JSON string.

## Return Value

The deserialized XNode

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# DeserializeXNode Method (value, deserializeRootElementName)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **DeserializeXNode(String, String)**

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static XDocument DeserializeXNode(  
    string value,  
    string deserializeRootElementName  
)
```

```
Public Shared Function DeserializeXNode ( _  
    value As String, _  
    deserializeRootElementName As String _  
) As XDocument
```

```
public:  
static XDocument^ DeserializeXNode(  
    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 XNode

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# DeserializeXNode Method (value, deserializeRootElementName, writeArrayAttribute)

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

**DeserializeXNode(String, String, Boolean)**

C#

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

## - Declaration Syntax

C#

Visual Basic

Visual C++

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

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

```
public:
static XDocument^ DeserializeXNode(
    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: 4.0.2.C  
(4.0.2.13623)

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++  
C++

```
public static readonly string False
```

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

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

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

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

## 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: 4.0.2.0  
(4.0.2.13623)

# NegativeInfinity Field

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

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

# PopulateObject Method

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

C# ▾

## - Members

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

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# PopulateObject Method (value, target)

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

**PopulateObject(String, Object)**

C# ▾

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)**

C# ▾

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 used.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# PositiveInfinity Field

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

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

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

# 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: 4.0.2.0  
(4.0.2.13623)

# SerializeObject Method (value, formatting)

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

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# SerializeObject Method (value, formatting, converters)

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

**SerializeObject(Object, Formatting, JsonConverter[])**

C# ▾

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 JsonConverter[] converters
)
```

```
Public Shared Function SerializeObject ( _
    value As Object, _
    formatting As Formatting, _
    ParamArray converters As JsonConverter() ) _
) 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: 4.0.2.0  
(4.0.2.13623)

# SerializeObject Method (value, formatting, settings)

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

**SerializeObject(Object, Formatting, JsonSerializerSettings)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# SerializeObject Method (value, converters)

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

**SerializeObject(Object, JsonConverter[])**

C# ▾

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 JsonConverter[] converters  
)
```

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

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

## Parameters

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

The object to serialize.

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

A collection converters used while serializing.

## Return Value

A JSON string representation of the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

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

C# ▾

**[-] Members**

Icon	Member	Description
= S	<a href="#">SerializeXmlNode(XmlNode)</a>	Serializes the XML node to a JSON string.
= S	<a href="#">SerializeXmlNode(XmlNode, Formatting)</a>	Serializes the XML node to a JSON string.
= S	<a href="#">SerializeXmlNode(XmlNode, Formatting, Boolean)</a>	Serializes the XML node to a JSON string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# SerializeXmlNode Method (node)

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

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# 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: 4.0.2.C  
(4.0.2.13623)

# SerializeXmlNode Method (node, formatting, omitRootObject)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **SerializeXmlNode(XmlNode, Formatting, Boolean)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

C# ▾

## - Members

Icon	Member	Description
= S	<a href="#">SerializeXNode(XObject)</a>	Serializes the <a href="#">XNode</a> to a JSON string.
= S	<a href="#">SerializeXNode(XObject, Formatting)</a>	Serializes the <a href="#">XNode</a> to a JSON string.
= S	<a href="#">SerializeXNode(XObject, Formatting, Boolean)</a>	Serializes the <a href="#">XNode</a> to a JSON string.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# SerializeXNode Method (node)

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

C# ▾

Serializes the [XNode](#) to a JSON string.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string SerializeXNode(  
    XObject node  
)
```

```
Public Shared Function SerializeXNode ( _  
    node As XObject _  
) As String
```

```
public:  
static String^ SerializeXNode(  
    XObject^ node  
)
```

## Parameters

### **node ([XObject](#))**

The node to convert to JSON.

## Return Value

A JSON string of the XNode.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# SerializeXNode Method (node, formatting)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **SerializeXNode(XObject, Formatting)**

C# ▾

Serializes the [XNode](#) to a JSON string.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string SerializeXNode(
    XObject node,
    Formatting formatting
)
```

```
Public Shared Function SerializeXNode ( _
    node As XObject, _
    formatting As Formatting _ 
) As String
```

```
public:
static String^ SerializeXNode(
    XObject^ node,
    Formatting formatting
)
```

## 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: 4.0.2.C  
(4.0.2.13623)

# SerializeXNode Method (node, formatting, omitRootObject)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **SerializeXNode(XObject, Formatting, Boolean)**

C# ▾

Serializes the [XNode](#) to a JSON string.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string SerializeXNode(
    XObject node,
    Formatting formatting,
    bool omitRootObject
)
```

```
Public Shared Function SerializeXNode ( _
    node As XObject, _
    formatting As Formatting, _
    omitRootObject As Boolean _
) As String
```

```
public:
static String^ SerializeXNode(
    XObject^ node,
    Formatting formatting,
    bool omitRootObject
)
```

## Parameters

### **node ([XObject](#))**

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: 4.0.2.C  
(4.0.2.13623)

# ToString Method

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

C# ▾

## - Members

Icon	Member	Description
= 	<a href="#">ToString(DateTime)</a>	Converts the <a href="#">DateTime</a> to its JSON string representation.
= 	<a href="#">ToString(DateTimeOffset)</a>	Converts the <a href="#">DateTimeOffset</a> to its JSON string representation.
= 	<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.

≡•S	<a href="#">ToString(String)</a>	Converts the <a href="#">String</a> to its JSON string representation.
≡•S	<a href="#">ToString(String, Char)</a>	Converts the <a href="#">String</a> to its JSON string representation.
≡•S	<a href="#">ToString(Object)</a>	Converts the <a href="#">Object</a> to its JSON string representation.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# ToString Method (value)

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

## ToString(Boolean)

C# ▾

Converts the [Boolean](#) to its JSON string representation.

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static string ToString(  
    bool value  
)
```

```
Public Shared Function ToString ( _  
    value As Boolean _  
) As String
```

```
public:  
static String^ ToString(  
    bool value  
)
```

### Parameters

#### value ([Boolean](#))

The value to convert.

### Return Value

A JSON string representation of the [Boolean](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

C# ▾

Converts the [Byte](#) to its JSON string representation.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    byte value  
)
```

```
Public Shared Function ToString ( _  
    value As Byte _  
) As String
```

```
public:  
static String^ ToString(  
    unsigned char value  
)
```

## Parameters

### **value (Byte)**

The value to convert.

## Return Value

A JSON string representation of the [Byte](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Converts the [Char](#) to its JSON string representation.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    char value  
)
```

```
Public Shared Function ToString ( _  
    value As Char _  
) As String
```

```
public:  
static String^ ToString(  
    wchar_t value  
)
```

## Parameters

### **value ([Char](#))**

The value to convert.

## Return Value

A JSON string representation of the [Char](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# ToString Method (value)

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

## ToString(DateTime)

C# ▾

Converts the [DateTime](#) to its JSON string representation.

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static string ToString(  
    DateTime value  
)
```

```
Public Shared Function ToString ( _  
    value As DateTime _  
) As String
```

```
public:  
static String^ ToString(  
    DateTime value  
)
```

### Parameters

#### **value** ([DateTime](#))

The value to convert.

### Return Value

A JSON string representation of the [DateTime](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# ToString Method (value)

[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: 4.0.2.0 (4.0.2.13623)

# ToString Method (value)

[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: 4.0.2.0 (4.0.2.13623)

C# ▾

Converts the [Double](#) to its JSON string representation.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    double value  
)
```

```
Public Shared Function ToString ( _  
    value As Double _  
) As String
```

```
public:  
static String^ ToString(  
    double value  
)
```

## Parameters

### **value ([Double](#))**

The value to convert.

## Return Value

A JSON string representation of the [Double](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Converts the [Enum](#) to its JSON string representation.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    Enum value  
)
```

```
Public Shared Function ToString ( _  
    value As Enum _  
) As String
```

```
public:  
static String^ ToString(  
    Enum^ value  
)
```

## Parameters

### **value (Enum)**

The value to convert.

## Return Value

A JSON string representation of the [Enum](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

C# ▾

Converts the [Guid](#) to its JSON string representation.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    Guid value  
)
```

```
Public Shared Function ToString ( _  
    value As Guid _  
) As String
```

```
public:  
static String^ ToString(  
    Guid value  
)
```

## Parameters

### **value ([Guid](#))**

The value to convert.

## Return Value

A JSON string representation of the [Guid](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

C# ▾

Converts the [Int16](#) to its JSON string representation.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    short value  
)
```

```
Public Shared Function ToString ( _  
    value As Short _  
) As String
```

```
public:  
static String^ ToString(  
    short value  
)
```

## Parameters

### **value** ([Int16](#))

The value to convert.

## Return Value

A JSON string representation of the [Int16](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Converts the [Int32](#) to its JSON string representation.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    int value  
)
```

```
Public Shared Function ToString ( _  
    value As Integer _  
) As String
```

```
public:  
static String^ ToString(  
    int value  
)
```

## Parameters

### **value** ([Int32](#))

The value to convert.

## Return Value

A JSON string representation of the [Int32](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Converts the [Int64](#) to its JSON string representation.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    long value  
)
```

```
Public Shared Function ToString ( _  
    value As Long _  
) As String
```

```
public:  
static String^ ToString(  
    long long value  
)
```

## Parameters

### **value** ([Int64](#))

The value to convert.

## Return Value

A JSON string representation of the [Int64](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Converts the [Object](#) to its JSON string representation.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    object value  
)
```

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

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

## Parameters

### **value (Object)**

The value to convert.

## Return Value

A JSON string representation of the [Object](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Converts the [SByte](#) to its JSON string representation.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    sbyte value  
)
```

```
Public Shared Function ToString ( _  
    value As SByte _  
) As String
```

```
public:  
static String^ ToString(  
    signed char value  
)
```

## Parameters

### **value ([SByte](#))**

The value to convert.

## Return Value

A JSON string representation of the [SByte](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Converts the [Single](#) to its JSON string representation.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    float value  
)
```

```
Public Shared Function ToString ( _  
    value As Single _  
) As String
```

```
public:  
static String^ ToString(  
    float value  
)
```

## Parameters

### **value ([Single](#))**

The value to convert.

## Return Value

A JSON string representation of the [Single](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Converts the [String](#) to its JSON string representation.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    string value  
)
```

```
Public Shared Function ToString ( _  
    value As String _  
) As String
```

```
public:  
static String^ ToString(  
    String^ value  
)
```

## Parameters

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

The value to convert.

## Return Value

A JSON string representation of the [String](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# ToString Method (value, delimiter)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConvert](#) ► **ToString(String, Char)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

C# ▾

Converts the [UInt16](#) to its JSON string representation.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    ushort value  
)
```

```
Public Shared Function ToString ( _  
    value As UShort _  
) As String
```

```
public:  
static String^ ToString(  
    unsigned short value  
)
```

## Parameters

### **value ([UInt16](#))**

The value to convert.

## Return Value

A JSON string representation of the [UInt16](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Converts the [UInt32](#) to its JSON string representation.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    uint value  
)
```

```
Public Shared Function ToString ( _  
    value As UInteger _  
) As String
```

```
public:  
static String^ ToString(  
    unsigned int value  
)
```

## Parameters

### **value ([UInt32](#))**

The value to convert.

## Return Value

A JSON string representation of the [UInt32](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Converts the [UInt64](#) to its JSON string representation.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static string ToString(  
    ulong value  
)
```

```
Public Shared Function ToString ( _  
    value As ULong _  
) As String
```

```
public:  
static String^ ToString(  
    unsigned long long value  
)
```

## Parameters

### **value ([UInt64](#))**

The value to convert.

## Return Value

A JSON string representation of the [UInt64](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

C# ▾

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

## 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: 4.0.2.0  
(4.0.2.13623)

# JsonConverter Class

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

C# ▾

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

 Public Protected Instance Static

Declared

Inherited

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 JsonConverter.
=♪	<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="#"><u>WriteJson(JsonWriter, Object, JsonSerializer)</u></a>	Writes the JSON representation of the object.
--	--	---

## [-] Inheritance Hierarchy

[Object](#)

└ [JsonConverter](#)

- └ [BinaryConverter](#)
- └ [BsonObjectIdConverter](#)
- └ [CustomCreationConverter<T>](#)
- └ [DataSetConverter](#)
- └ [DataTableConverter](#)
- └ [DateTimeConverterBase](#)
- └ [EntityKeyMemberConverter](#)
- └ [ExpandoObjectConverter](#)
- └ [KeyValuePairConverter](#)
- └ [RegexConverter](#)
- └ [StringEnumConverter](#)
- └ [XmlNodeConverter](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

## Constructor

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

**JsonConverter()**

C# ▾

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

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected JsonConverter()
```

```
Protected Sub New
```

```
protected:  
JsonConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# CanConvert Method (*objectType*)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConverter](#) ► [CanConvert\(Type\)](#)

C# ▾

Determines whether this instance can convert the specified object type.

## 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.

# CanRead Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConverter](#) ► **CanRead**

C# ▾

Gets a value indicating whether this [JsonConverter](#) can read JSON.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual bool CanRead { get; }
```

```
Public Overridable ReadOnly Property CanRead As Boolean  
    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: 4.0.2.0 (4.0.2.13623)

# CanWrite Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConverter](#) ► **CanWrite**

C# ▾

Gets a value indicating whether this [JsonConverter](#) can write JSON.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual bool CanWrite { get; }
```

```
Public Overridable ReadOnly Property CanWrite As Boolean  
    Get
```

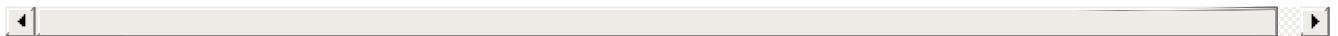
```
public:  
    virtual property bool CanWrite {  
        bool get ();  
    }
```

## Value

true if this [JsonConverter](#) can write JSON; otherwise, false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



C# ▾

Gets the [JsonSchema](#) of the JSON produced by the JsonConverter.

## 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: 4.0.2.C  
(4.0.2.13623)

# 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: 4.0.2.C  
(4.0.2.13623)

# 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: 4.0.2.0  
(4.0.2.13623)

# JsonConverterAttribute

## Class

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

C# ▾

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 : pub
```

### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherit

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: 4.0.2.0  
(4.0.2.13623)



# JsonConverterAttribute

## Constructor (converterType)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConverterAttribute](#) ► [JsonConverterAttribute\(Type\)](#)

C# ▾

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

### 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: 4.0.2.0 (4.0.2.13623)

# ConverterType Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonConverterAttribute](#) ► **ConverterType**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# JsonConverterCollection

## 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

Declare

Inherit

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>	Gets a <a href="#">IList&lt;T&gt;</a> wrapper around the <a href="#">Collection&lt;T&gt;</a> .  (Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a> .)

	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Remove(T)</a>	Removes the first occurrence of a specific object from the <a href="#">Collection&lt;T&gt;</a> .  (Inherited from <a href="#">Collection&lt;JsonConverter&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;JsonConverter&gt;</a> .)
	<a href="#">RemoveItem(Int32)</a>	Removes the element at the specified index of the <a href="#">Collection&lt;T&gt;</a> .  (Inherited from <a href="#">Collection&lt;JsonConverter&gt;</a> .)
	<a href="#">SetItem(Int32, T)</a>	Replaces the element at the specified index.  (Inherited from <a href="#">Collection&lt;JsonConverter&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<JsonConverter>](#)

- └ [JsonConverterCollection](#)

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



# JsonConverterCollection Constructor

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

C#

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonConverterCollection()
```

```
Public Sub New
```

```
public:  
JsonConverterCollection()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonIgnoreAttribute

## Class

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

C#

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	Declare
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static		<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Inherit

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>	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> .)
扫一	<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](#)

  └ **JsonIgnoreAttribute**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



# JsonIgnoreAttribute Constructor

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

**JsonIgnoreAttribute()**

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonIgnoreAttribute()
```

```
Public Sub New
```

```
public:  
    JsonIgnoreAttribute()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# JsonObjectAttribute Class

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

C# ▾

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

Public

Protected

Instance

Static

Declared

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 specific 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>	Gets the <a href="#">Type</a> of the current instance.

		(Inherited from <a href="#">Object</a> )
	<a href="#"><u>Id</u></a>	Gets or sets the id.
	<a href="#"><u>IsDefaultAttribute()</u></a>	(Inherited from <a href="#">JsonContainerAttribut</a> ) When overridden in a derived class, indicate whether the value of this instance is the default value for the derived class.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#"><u>IsReference</u></a>	Gets or sets a value that indicates whether to preserve object reference data.  (Inherited from <a href="#">JsonContainerAttribut</a> )
	<a href="#"><u>Match(Object)</u></a>	When overridden in a derived class, returns value that indicates whether this instance equals a specified object.  (Inherited from <a href="#">Attribute</a> .)
	<a href="#"><u>MemberSerialization</u></a>	Gets or sets the member serialization.
	<a href="#"><u>MemberwiseClone()</u></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="#">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 the <a href="#">Attribute</a> .  (Inherited from <a href="#">Attribute</a> .)

## Inheritance Hierarchy

```

Object
  ↘ Attribute
    ↘ JsonContainerAttribute
      ↘ JsonObjectAttribute

```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# JsonObjectAttribute Constructor

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

**JsonObjectAttribute()**

C# ▾

## 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: 4.0.2.C  
(4.0.2.13623)

# JsonObjectAttribute Constructor

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

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonObjectAttribute()
```

```
Public Sub New
```

```
public:  
JsonObjectAttribute()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonObjectAttribute Constructor (memberSerialization)

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

**JsonObjectAttribute(MemberSerialization)**

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonObjectAttribute Constructor (id)

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

**JsonObjectAttribute(String)**

C#

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonObjectAttribute(  
    string id  
)
```

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

```
public:  
JsonObjectAttribute(  
    String^ id  
)
```

## Parameters

*id* ([String](#))

The container Id.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

## Property

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

### MemberSerialization

C# ▾

Gets or sets the member serialization.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public MemberSerialization MemberSerialization { get
```

```
Public Property MemberSerialization As MemberSerialization
    Get
    Set
```

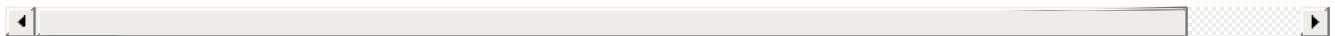
```
public:
property MemberSerialization MemberSerialization {
    MemberSerialization get ();
    void set (MemberSerialization value);
}
```

#### Value

The member serialization.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# JsonPropertyAttribute Class

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

C#

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 : public
```

## Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherit

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="#">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="#">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: 4.0.2.0  
(4.0.2.13623)



# JsonPropertyAttribute Constructor

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

**JsonPropertyAttribute()**

C# ▾

## ▀ 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: 4.0.2.0  
(4.0.2.13623)

# JsonPropertyAttribute Constructor

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

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonPropertyAttribute()
```

```
Public Sub New
```

```
public:  
JsonPropertyAttribute()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonPropertyAttribute Constructor (propertyName)

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

**JsonPropertyAttribute(String)**

C#

Initializes a new instance of the [JsonPropertyAttribute](#) class with the specified name.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonPropertyAttribute(  
    string propertyName  
)
```

```
Public Sub New ( _  
    propertyName As String _  
)
```

```
public:  
JsonPropertyAttribute(  
    String^ propertyName  
)
```

## Parameters

**propertyName (String)**

Name of the property.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# DefaultValueHandling

## Property

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

### DefaultValueHandling

C#

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: 4.0.2.0  
(4.0.2.13623)



# IsReference Property

[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: 4.0.2.0  
(4.0.2.13623)

# NullValueHandling

## Property

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

### NullValueHandling

C# ▾

Gets or sets the null value handling used when serializing this property.

#### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public NullValueHandling NullValueHandling { get; se
```

```
Public Property NullValueHandling As NullValueHandling
    Get
    Set
```

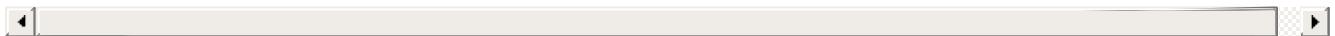
```
public:
property NullValueHandling NullValueHandling {
    NullValueHandling get ();
    void set (NullValueHandling value);
}
```

#### Value

The null value handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



## Property

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

### ObjectCreationHandling

C#

Gets or sets the object creation handling used when deserializing this property.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public ObjectCreationHandling ObjectCreationHandling
```

```
Public Property ObjectCreationHandling As ObjectCreationHandling
    Get
    Set
```

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

#### Value

The object creation handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# PropertyName Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonPropertyAttribute](#) ► **PropertyName**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

## Property

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

### ReferenceLoopHandling

C# ▾

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: 4.0.2.0  
(4.0.2.13623)



# Required Property

[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: 4.0.2.0  
(4.0.2.13623)

## 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: 4.0.2.0 (4.0.2.13623)

# 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"/> Inherit

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="#">CurrentState</a>	Gets the current reader state.
	<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="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)

	<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="#">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="#">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: 4.0.2.0  
(4.0.2.13623)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► [JsonReader\(\)](#)

C# ▾

Initializes a new instance of the [JsonReader](#) class with the specified [TextReader](#).

## - Declaration Syntax

C#

Visual Basic

Visual C++

```
protected JsonReader()
```

```
Protected Sub New
```

```
protected:  
JsonReader()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

▾

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: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

C# ▾

Gets the current reader state.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
protected JsonReader.State CurrentState { get; }
```

```
Protected Readonly Property CurrentState As JsonRead
Get
```

```
protected:
property JsonReader.State CurrentState {
    JsonReader.State get ();
}
```

## Value

The current reader state.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



C#  
▼

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 Integer  
    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: 4.0.2.0 (4.0.2.13623)

# Dispose Method (disposing)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **Dispose(Boolean)**

C#

Releases unmanaged and - optionally - managed resources

## 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: 4.0.2.0 (4.0.2.13623)

# QuoteChar Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **QuoteChar**

C# ▾

Gets the quotation mark character used to enclose the value of a string.

## 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: 4.0.2.0 (4.0.2.13623)



# Read Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **Read()**

C# ▾

Reads the next JSON token from the stream.

## Declaration Syntax

C#

Visual Basic

Visual C++  
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: 4.0.2.C  
(4.0.2.13623)

# ReadAsBytes Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► [ReadAsBytes\(\)](#)

C# ▾

Reads the next JSON token from the stream as a **Byte[]**.

## 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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# ReadAsDateTimeOffset

## Method

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

**ReadAsDateTimeOffset()**

C# ▾

Reads the next JSON token from the stream as a [Nullable<T>](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract Nullable<DateTimeOffset> ReadAsDateTimeOffset()
```

```
Public MustOverride Function ReadAsDateTimeOffset As Nullable<DateTimeOffset>
```

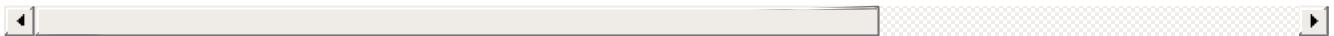
```
public:  
virtual Nullable<DateTimeOffset> ReadAsDateTimeOffset()
```

### Return Value

A [Nullable<T>](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# ReadAsDecimal Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► [ReadAsDecimal\(\)](#)

C# ▾

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 Nullable<Decimal>
```

```
public:  
virtual Nullable<Decimal> ReadAsDecimal() abstract
```

## Return Value

A [Nullable<T>](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



# SetStateBasedOnCurrent Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **SetStateBasedOnCurrent()**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

C# ▾

## - 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: 4.0.2.0  
(4.0.2.13623)

# SetToken Method (newToken)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **SetToken(JsonToken)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# SetToken Method (newToken, value)

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

**SetToken(JsonToken, Object)**

C# ▾

Sets the current token and value.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual void SetToken(  
    JsonToken newToken,  
    object value  
)
```

```
Protected Overridable Sub SetToken ( _  
    newToken As JsonToken, _  
    value As Object _  
)
```

```
protected:  
virtual void SetToken(  
    JsonToken newToken,  
    object^ value  
)
```

## Parameters

### **newToken (JsonToken)**

The new token.

### **value (Object)**

The value.

▾

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: 4.0.2.0  
(4.0.2.13623)

C# ▾

Gets the type of the current Json token.

## Declaration Syntax

C#

Visual Basic

Visual C++

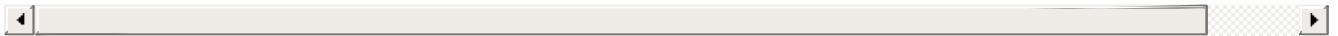
```
public virtual JsonToken TokenType { get; }
```

```
Public Overridable ReadOnly Property TokenType As Js
    Get
```

```
public:
virtual property JsonToken TokenType {
    JsonToken get ();
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



Gets the text value of the current Json token.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual Object Value { get; }
```

```
Public Overridable ReadOnly Property Value As Object
    Get
```

```
public:
virtual property Object^ Value {
    Object^ get ();
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# ValueType Property

[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: 4.0.2.0 (4.0.2.13623)

## Enumeration

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReader](#) ► **JsonReader.State**

C# ▾

Specifies the state of the reader.

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
protected enum State
```

```
Protected Enumeration State
```

```
protected 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: 4.0.2.C  
(4.0.2.13623)

# 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++

```
public class JsonReaderException : Exception
```

```
Public Class JsonReaderException _
    Inherits Exception
```

```
public ref class JsonReaderException : public Except
```

### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherit

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="#">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="#">LineNumber</a>	Gets the line number indicating where the error occurred.
	<a href="#">LinePosition</a>	Gets the line position indicating where the error occurred.
	<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
  └ JsonReaderException
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonReaderException Constructor

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

**JsonReaderException()**

C# ▾

## 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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# JsonReaderException Constructor

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

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonReaderException()
```

```
Public Sub New
```

```
public:  
JsonReaderException()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# JsonReaderException

## Constructor (message)

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

**JsonReaderException(String)**

C#

Initializes a new instance of the [JsonReaderException](#) class with a specified error message.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonReaderException(  
    string message  
)
```

```
Public Sub New ( _  
    message As String _  
)
```

```
public:  
JsonReaderException(  
    String^ message  
)
```

### Parameters

#### **message (String)**

The error message that explains the reason for the exception.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonReaderException

## Constructor (message, innerException)

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

**JsonReaderException(String, Exception)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# LineNumber Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReaderException](#) ► **LineNumber**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# LinePosition Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonReaderException](#) ► **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: 4.0.2.0  
(4.0.2.13623)

# 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++ ▾

```
public class JsonSerializationException : Exception
```

```
Public Class JsonSerializationException _  
    Inherits Exception
```

```
public ref class JsonSerializationException : public
```

## Members

All Members

Constructors

Methods

Properties

 Public Protected Instance Static

Declared

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 with a specified error message.

	<a href="#">JsonSerializationException(String, Exception)</a>	Initializes a new instance of the <b>JsonSerializationException</b> class with a specified error message and a reference to the inner exception that is the cause of this exception.
	<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 coded 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](#)

  └ **JsonSerializationException**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



C# ▾

## 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 specified error message.
≡	<a href="#">JsonSerializationException(String, Exception)</a>	Initializes a new instance of the <a href="#">JsonSerializationException</a> class with a specified error message and a reference to the inner exception that is the cause of this exception.

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSerializationException()
```

```
Public Sub New
```

```
public:  
JsonSerializationException()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# 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: 4.0.2.0 (4.0.2.13623)

# 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: 4.0.2.C  
(4.0.2.13623)

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++

```
public class JsonSerializer
```

```
Public Class JsonSerializer
```

```
public ref class JsonSerializer
```

## Members

All Members

Constructors

Methods

Properties

Events

 Public Instance Declared Protected Static 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="#">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.
 <b>S</b>	<a href="#">Create(JsonSerializerSettings)</a>	Creates a new <b>JsonSerializer</b> instance using the specified <a href="#">JsonSerializerSettings</a> .
	<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="#">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="#">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: 4.0.2.C  
(4.0.2.13623)

## Constructor

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

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# Binder Property

[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: 4.0.2.0 (4.0.2.13623)



## 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: 4.0.2.0 (4.0.2.13623)



C# ▾

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: 4.0.2.0  
(4.0.2.13623)

## Property

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

**ContractResolver**

C# ▾

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++

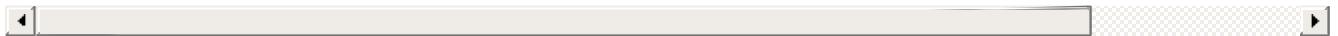
```
public virtual IContractResolver ContractResolver {
```

```
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: 4.0.2.0 (4.0.2.13623)



C# ▾

Gets a collection [JsonConverter](#) that will be used during serialization.

## 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: 4.0.2.0 (4.0.2.13623)



# Create Method (settings)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► **Create(JsonSerializerSettings)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# DefaultValueHandling

## Property

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

### DefaultValueHandling

C# ▾

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: 4.0.2.C  
(4.0.2.13623)



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.

# Deserialize Method (reader)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► [Deserialize\(JsonReader\)](#)

C# ▾

Deserializes the Json structure contained by the specified [JsonReader](#).

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public Object Deserialize(  
    JsonReader reader  
)
```

```
Public Function Deserialize ( _  
    reader As JsonReader _  
) As Object
```

```
public:  
Object^ Deserialize(  
    JsonReader^ reader  
)
```

## Parameters

### reader ([JsonReader](#))

The [JsonReader](#) that contains the JSON structure to deserialize.

## Return Value

The [Object](#) being deserialized.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# Deserialize<T> Method (reader)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► **Deserialize<T>(JsonReader)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# Deserialize Method (reader, objectType)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► **Deserialize(JsonReader, Type)**

C# ▾

Deserializes the Json structure contained by the specified [JsonReader](#) into an instance of the specified type.

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public Object Deserialize(  
    JsonReader reader,  
    Type objectType  
)
```

```
Public Function Deserialize ( _  
    reader As JsonReader, _  
    objectType As Type _  
) As Object
```

```
public:  
Object^ Deserialize(  
    JsonReader^ reader,  
    Type^ objectType  
)
```

## Parameters

### *reader* ([JsonReader](#))

The [JsonReader](#) containing the object.

### *objectType* ([Type](#))

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

## Return Value

The instance of *objectType* being deserialized.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# Deserialize Method (reader, objectType)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► [Deserialize\(TextReader, Type\)](#)

C# ▾

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 serialized.

## Return Value

The instance of *objectType* being deserialized.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

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: 4.0.2.0 (4.0.2.13623)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► [MissingMemberHandling](#)

C# ▾

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++ ▾

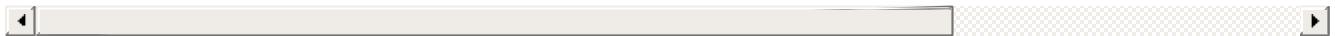
```
public virtual MissingMemberHandling MissingMemberHa
```

```
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: 4.0.2.0 (4.0.2.13623)



# NullValueHandling

## Property

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

### NullValueHandling

C# ▾

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 Nul
    Get
    Set
```

```
public:
virtual property NullValueHandling NullValueHandling
    NullValueHandling get ();
    void set (NullValueHandling value);
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



## 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 As ObjectCreationHandling
    Get
    Set
```

```
public:
virtual property ObjectCreationHandling ObjectCreationHandling {
    ObjectCreationHandling get ();
    void set (ObjectCreationHandling value);
}
```

### Value

The object creation handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# Populate Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► [Populate\(\)](#)

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: 4.0.2.0 (4.0.2.13623)

# 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.

# Populate Method (reader, target)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► **Populate(TextReader, Object)**

C# ▾

Populates the JSON values onto the target object.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Populate(  
    TextReader reader,  
    object target  
)
```

```
Public Sub Populate ( _  
    reader As TextReader, _  
    target As Object _  
)
```

```
public:  
void Populate(  
    TextReader^ reader,  
    Object^ target  
)
```

## Parameters

### **reader ([TextReader](#))**

The [TextReader](#) that contains the JSON structure to reader values from.

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

The target object to populate values onto.

# Documentation **PreserveReferencesHandling** Property

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

## PreserveReferencesHandling

C# ▾

Gets or sets how object references are preserved by the serializer.

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

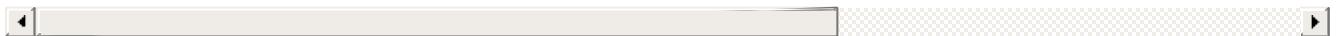
```
public virtual PreserveReferencesHandling PreserveRe
```

```
Public Overridable Property PreserveReferencesHandli
    Get
    Set
```

```
public:
virtual property PreserveReferencesHandling Preserve
    PreserveReferencesHandling get ();
    void set (PreserveReferencesHandling value);
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► [ReferenceLoopHandling](#)

Get or set how reference loops (e.g. a class referencing itself) is handled.

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: 4.0.2.0 (4.0.2.13623)



## Property

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

**ReferenceResolver**

C# ▾

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^ ReferenceResolver;
IReferenceResolver^ get ();
void set (IReferenceResolver^ value);
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# Serialize Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► **Serialize()**

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: 4.0.2.0 (4.0.2.13623)

# Serialize Method

## (**jsonWriter**, **value**)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► **Serialize(JsonWriter, Object)**

C# ▾

Serializes the specified [Object](#) and writes the Json structure to a [Stream](#) using the specified [JsonWriter](#).

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public void Serialize(
    JsonWriter jsonWriter,
    Object value
)
```

```
Public Sub Serialize ( _
    jsonWriter As JsonWriter, _
    value As Object _
)
```

```
public:
void Serialize(
    JsonWriter^ jsonWriter,
    Object^ value
)
```

### Parameters

#### **jsonWriter** ([JsonWriter](#))

The [JsonWriter](#) used to write the Json structure.

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

The [Object](#) to serialize.

# Serialize Method (*textWriter*, *value*)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializer](#) ► [Serialize\(TextWriter, Object\)](#)

C# ▾

Serializes the specified [Object](#) and writes the Json structure to a [Stream](#) using the specified [TextWriter](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Serialize(
    TextWriter textWriter,
    Object value
)
```

```
Public Sub Serialize ( _
    textWriter As TextWriter, _
    value As Object _
)
```

```
public:
void Serialize(
    TextWriter^ textWriter,
    Object^ value
)
```

## Parameters

### *textWriter* ([TextWriter](#))

The [TextWriter](#) used to write the Json structure.

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

The [Object](#) to serialize.

# TypeNameAssemblyFormat Property

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

## TypeNameAssemblyFormat

C# ▾

Gets or sets how a type name assembly is written and resolved by the serializer.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual FormatterAssemblyStyle TypeNameAssemb
```

```
Public Overridable Property TypeNameAssemblyFormat As FormatterAssemblyStyle
    Get
    Set
```

```
public:
virtual property FormatterAssemblyStyle TypeNameAssemblyFormat {
    FormatterAssemblyStyle get ();
    void set (FormatterAssemblyStyle value);
}
```

### Value

The type name assembly format.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



## Property

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

### TypeNameHandling

C# ▾

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: 4.0.2.0 (4.0.2.13623)



# JsonSerializerSettings

## Class

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

C# ▾

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



Declared



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="#">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="#">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

		<p><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="#">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="#">MissingMemberHandling</a>	<p>Gets or sets how missing members (e.g. JSON contains a property that isn't a member on the object) are handled during deserialization.</p>
📝	<a href="#">NullValueHandling</a>	<p>Gets or sets how null values are handled during serialization and deserialization.</p>
📝	<a href="#">ObjectCreationHandling</a>	<p>Gets or sets how objects are created during deserialization.</p>
📝	<a href="#">PreserveReferencesHandling</a>	<p>Gets or sets how object references are preserved by the serializer.</p>
📝	<a href="#">ReferenceLoopHandling</a>	<p>Gets or sets how reference loops</p>

		(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 writing and reading is handled by the serializer.

## Inheritance Hierarchy

[Object](#)

└ **JsonSerializerSettings**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# JsonSerializerSettings Constructor

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

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# Binder Property

[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: 4.0.2.0  
(4.0.2.13623)

## Property

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

[ConstructorHandling](#)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)



# Context Property

[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: 4.0.2.0  
(4.0.2.13623)

# ContractResolver

## Property

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

### ContractResolver

C# ▾

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 IContractResolver
    Get
    Set
```

```
public:
property IContractResolver^ ContractResolver {
    IContractResolver^ get ();
    void set (IContractResolver^ value);
}
```

#### Value

The contract resolver.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# Converters Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ► [Converters](#)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# DefaultValueHandling

## Property

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

### DefaultValueHandling

C#

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: 4.0.2.0  
(4.0.2.13623)



# Error Property

[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 ErrorEventArgs)
    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: 4.0.2.0  
(4.0.2.13623)



## Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonSerializerSettings](#) ► [MissingMemberHandling](#)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)



# NullValueHandling

## Property

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

### NullValueHandling

C#

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 NullValueHandling
    Get
    Set
```

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

#### Value

Null value handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



## Property

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

[ObjectCreationHandling](#)

C#

Gets or sets how objects are created during deserialization.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public ObjectCreationHandling ObjectCreationHandling
```

```
Public Property ObjectCreationHandling As ObjectCreationHandling
    Get
    Set
```

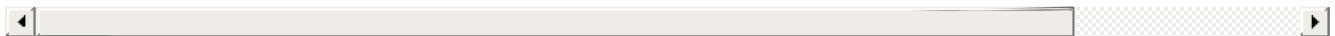
```
public:
property ObjectCreationHandling ObjectCreationHandling
    ObjectCreationHandling get ();
    void set (ObjectCreationHandling value);
}
```

### Value

The object creation handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# PreserveReferencesHandling Property

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

## PreserveReferencesHandling

C# ▾

Gets or sets how object references are preserved by the serializer.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public PreserveReferencesHandling PreserveReferences
```

```
Public Property PreserveReferencesHandling As Preser
    Get
    Set
```

```
public:
property PreserveReferencesHandling PreserveReference
    PreserveReferencesHandling get ();
    void set (PreserveReferencesHandling value);
}
```

### Value

The preserve references handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



## Property

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

[ReferenceLoopHandling](#)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)



## Property

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

**ReferenceResolver**

C# ▾

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 IReferenceResolver
    Get
    Set
```

```
public:
property IReferenceResolver^ ReferenceResolver {
    IReferenceResolver^ get ();
    void set (IReferenceResolver^ value);
}
```

### Value

The reference resolver.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



C# ▾

Gets or sets how a type name assembly is written and resolved by the serializer.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public FormatterAssemblyStyle TypeNameAssemblyFormat
```

```
Public Property TypeNameAssemblyFormat As FormatterA
    Get
    Set
```

```
public:
property FormatterAssemblyStyle TypeNameAssemblyFormat
    FormatterAssemblyStyle get ();
    void set (FormatterAssemblyStyle value);
}
```

## Value

The type name assembly format.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



## Property

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

### TypeNameHandling

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# JsonTextReader Class

[Namespaces](#) ► [Newtonsoft.Json](#) ► **JsonTextReader** 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 JsonTextReader : JsonReader,  

    IJsonLineInfo
```

```
Public Class JsonTextReader  
    Inherits JsonReader  
    Implements IJsonLineInfo
```

```
public ref class JsonTextReader : public JsonReader,  

    IJsonLineInfo
```

## Members

All Members

Constructors

Methods

Properties

 Public Protected Instance Static

Declared

Inherited

Icon	Member	Description
	<a href="#">JsonTextReader(TextReader)</a>	Initializes a new instance of the <a href="#">JsonReader</a> class with the specified <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 closed.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">CurrentState</a>	Gets the current reader state.  (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="#">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="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</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="#">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="#">SetStateBasedOnCurrent()</a>	Sets the state based on current token.

		<p>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 token.</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
	<a href="#">TokenType</a>	<p>Gets the type of the current Json token.</p> <p>(Inherited from <a href="#">JsonReader</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="#">Value</a>	<p>Gets the text value of the current Json token.</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>
	<a href="#">ValueType</a>	<p>Gets The Common Language Runtime (CLR) type for the current Json token.</p> <p>(Inherited from <a href="#">JsonReader</a>.)</p>

## Inheritance Hierarchy

[Object](#)

└ [JsonReader](#)

  └ [JsonTextReader](#)

(4.0.2.13623)



## Constructor (reader)

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

**JsonTextReader(TextReader)**

C# ▾

Initializes a new instance of the [JsonReader](#) class with the specified [TextReader](#).

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public JsonTextReader(  
    TextReader reader  
)
```

```
Public Sub New ( _  
    reader As TextReader _  
)
```

```
public:  
JsonTextReader(  
    TextReader^ reader  
)
```

### Parameters

**reader ([TextReader](#))**

The [TextReader](#) containing the XML data to read.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# Close Method

[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: 4.0.2.0  
(4.0.2.13623)

C# ▾

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

## 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: 4.0.2.0  
(4.0.2.13623)

C#  
▼

Gets the current line number.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public int LineNumber { get; }
```

```
Public ReadOnly Property LineNumber As Integer
    Get
```

```
public:
virtual property int LineNumber {
    int get () sealed;
}
```

## 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: 4.0.2.0  
(4.0.2.13623)

# LinePosition Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextReader](#) ► [LinePosition](#)

C# ▾

Gets the current line position.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public int LinePosition { get; }
```

```
Public ReadOnly Property LinePosition As Integer
    Get
```

```
public:
virtual property int LinePosition {
    int get () sealed;
}
```

## 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: 4.0.2.0  
(4.0.2.13623)

# Read Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextReader](#) ► **Read()**

C# ▾

Reads the next JSON token from the stream.

## Declaration Syntax

C#

Visual Basic

Visual C++  
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: 4.0.2.C  
(4.0.2.13623)

# ReadAsBytes Method

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

## ReadAsBytes()

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# ReadAsDateTimeOffset

## Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextReader](#) ► **ReadAsDateTimeOffset()**

C# ▾

Reads the next JSON token from the stream as a [Nullable<T>](#).

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public override Nullable<DateTimeOffset> ReadAsDateTimeOffset()
```

```
Public Overrides Function ReadAsDateTimeOffset As Nullable<DateTimeOffset>
```

```
public:  
    virtual Nullable<DateTimeOffset> ReadAsDateTimeOffset()
```

### Return Value

A [DateTimeOffset](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# ReadAsDecimal Method

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

## ReadAsDecimal()

C# ▾

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>](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



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"/> Declare <input checked="" type="checkbox"/> Inherit

Icon	Member	Description
	<a href="#">JsonTextWriter(TextWriter)</a>	Creates an instance of the <a href="#">JsonWriter</a> class using the specified <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 the underlying stream or the output buffer is closed.

		should be closed when the stream is closed.
		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">Equals(Object)</a>	Determines whether the current object 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 operations before the <a href="#">Object</a> is destroyed by garbage collection.
		(Inherited from <a href="#">Object</a> .)
≡	<a href="#">Flush()</a>	Flushes whatever is in the underlying streams and closes the underlying stream.
		(Overrides <a href="#">JsonWriter.Flush()</a> )
📋	<a href="#">Formatting</a>	Indicates how the output is formatted.
		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">GetHashCode()</a>	Serves as a hash function for the current 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="#">Indentation</a>	Gets or sets how many indentation characters to write for each level in the JSON output. If <a href="#">Formatting</a> is set to <a href="#">Formatting.Indented</a> , this property is ignored.
		(Inherited from <a href="#">JsonWriter</a> )
📋	<a href="#">IndentChar</a>	Gets or sets which character to use for indentation.

		indenting when <a href="#">Formatting</a> . Inden
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">QuoteChar</a>	Gets or sets which character is used 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.
	<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 specified end token.

		(Overrides <a href="#">JsonWriter.WriteEnd(JsonWriter)</a> )
≡	<a href="#">WriteEnd()</a>	Writes the end of the cursor 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 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.  (Overrides <a href="#">JsonWriter.WriteLine()</a> )
💡	<a href="#">WriteIndentSpace()</a>	Writes an indent space.  (Overrides <a href="#">JsonWriter.WriteLineSpace()</a> )
≡	<a href="#">WriteNull()</a>	Writes a null value.  (Overrides <a href="#">JsonWriter.WriteLine()</a> )
≡	<a href="#">WritePropertyName(String)</a>	Writes the property name/value pair on a Json object.  (Overrides <a href="#">JsonWriter.WriteProperty(String)</a> )
≡	<a href="#">WriteRaw(String)</a>	Writes raw JSON.  (Overrides <a href="#">JsonWriter.WriteLine()</a> )
≡	<a href="#">WriteRawValue(String)</a>	Writes raw JSON where expected and updates the current cursor.  (Overrides <a href="#">JsonWriter.WriteLine()</a> )

		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteStartArray()</a>	Writes the beginning of an array.  (Overrides <a href="#">JsonWriter.WriteStartArray()</a> )
≡	<a href="#">WriteStartConstructor(String)</a>	Writes the start of a constructor with the given name.  (Overrides <a href="#">JsonWriter.WriteStartConstructor()</a> )
≡	<a href="#">WriteStartObject()</a>	Writes the beginning of an object.  (Overrides <a href="#">JsonWriter.WriteStartObject()</a> )
	<a href="#">WriteState</a>	Gets the state of the writer.  (Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteToken(JsonReader)</a>	Writes the current <a href="#">JsonReader</a> token.  (Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteUndefined()</a>	Writes an undefined value.  (Overrides <a href="#">JsonWriter.WriteLine()</a> )
≡	<a href="#">WriteValue(String)</a>	Writes a <a href="#">String</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(String)</a> )
≡	<a href="#">WriteValue(Int32)</a>	Writes a <a href="#">Int32</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(Int32)</a> )
≡	<a href="#">WriteValue(UInt32)</a>	Writes a <a href="#">UInt32</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(UInt32)</a> )
≡	<a href="#">WriteValue(Int64)</a>	Writes a <a href="#">Int64</a> value.

		(Overrides <a href="#">JsonWriter.WriteLine</a> )
≡	<a href="#">WriteValue(UInt64)</a>	Writes a <a href="#">UInt64</a> value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
≡	<a href="#">WriteValue(Single)</a>	Writes a <a href="#">Single</a> value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
≡	<a href="#">WriteValue(Double)</a>	Writes a <a href="#">Double</a> value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
≡	<a href="#">WriteValue(Boolean)</a>	Writes a <a href="#">Boolean</a> value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
≡	<a href="#">WriteValue(Int16)</a>	Writes a <a href="#">Int16</a> value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
≡	<a href="#">WriteValue(UInt16)</a>	Writes a <a href="#">UInt16</a> value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
≡	<a href="#">WriteValue(Char)</a>	Writes a <a href="#">Char</a> value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
≡	<a href="#">WriteValue(Byte)</a>	Writes a <a href="#">Byte</a> value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
≡	<a href="#">WriteValue(SByte)</a>	Writes a <a href="#">SByte</a> value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
≡	<a href="#">WriteValue(Decimal)</a>	Writes a <a href="#">Decimal</a> value.

		(Overrides <a href="#">JsonWriter.WriteLine(D</a>
≡	<a href="#">WriteValue(DateTime)</a>	Writes a <a href="#">DateTime</a> value  (Overrides <a href="#">JsonWriter.WriteLine(D</a>
≡	<a href="#">WriteValue(Byte[])</a>	Writes a <b>Byte[]</b> value.  (Overrides <a href="#">JsonWriter.WriteLine(B</a>
≡	<a href="#">WriteValue(DateTimeOffset)</a>	Writes a <a href="#">DateTimeOffset</a>  (Overrides <a href="#">JsonWriter.WriteLine(D</a>
≡	<a href="#">WriteValue(Nullable&lt;Int32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrit</a>
≡	<a href="#">WriteValue(Nullable&lt;UInt32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrit</a>
≡	<a href="#">WriteValue(Nullable&lt;Int64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrit</a>
≡	<a href="#">WriteValue(Nullable&lt;UInt64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrit</a>
≡	<a href="#">WriteValue(Nullable&lt;Single&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrit</a>
≡	<a href="#">WriteValue(Nullable&lt;Double&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrit</a>
≡	<a href="#">WriteValue(Nullable&lt;Boolean&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val  (Inherited from <a href="#">JsonWrit</a>

≡	<a href="#">WriteValue(Nullable&lt;Int16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value. (Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;UInt16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value. (Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;Char&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value. (Inherited from <a href="#">JsonWriter</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(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. (Overrides <a href="#">JsonWriter.WriteValueDelimiter()</a> )
≡	<a href="#">WriteWhitespace(String)</a>	Writes out the given whitespace character.

(Overrides  
[JsonWriter.WriteWhiteSpace](#)

## **[-] Inheritance Hierarchy**

[Object](#)

└ [JsonWriter](#)

  └ **JsonTextWriter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



## Constructor (textWriter)

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

[JsonTextWriter\(TextWriter\)](#)

C#

Creates an instance of the `JsonWriter` class using the specified `TextWriter`.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonTextWriter(  
    TextWriter textWriter  
)
```

```
Public Sub New ( _  
    textWriter As TextWriter _  
)
```

```
public:  
JsonTextWriter(  
    TextWriter^ textWriter  
)
```

### Parameters

**`textWriter (TextWriter)`**

The `TextWriter` to write to.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# Close Method

[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: 4.0.2.0  
(4.0.2.13623)

# Flush Method

[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: 4.0.2.C  
(4.0.2.13623)

# Indentation Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► [Indentation](#)

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: 4.0.2.C  
(4.0.2.13623)

# IndentChar Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **IndentChar**

C# ▾

Gets or sets which character to use for indenting when [Formatting](#) is set to [Formatting.Indented](#).

## 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: 4.0.2.C  
(4.0.2.13623)

# QuoteChar Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **QuoteChar**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# QuoteName Property

[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: 4.0.2.0  
(4.0.2.13623)

## WriteComment Method

### (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)

C# ▾

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: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

# WriteIndentSpace

## Method

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

### WriteIndentSpace()

C# ▾

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: 4.0.2.0  
(4.0.2.13623)



Writes a null value.

## Declaration Syntax

```
public override void WriteNull()
```

```
Public Overrides Sub WriteNull
```

```
public:  
virtual void WriteNull() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# WritePropertyName

## Method (name)

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

### WritePropertyName(String)

C# ▾

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: 4.0.2.0 (4.0.2.13623)

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

## WriteRaw(String)

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteStartArray Method

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

## WriteStartArray()

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# WriteStartConstructor

## Method (name)

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

**WriteStartConstructor(String)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteStartObject Method

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

## WriteStartObject()

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **WriteUndefined()**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

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.

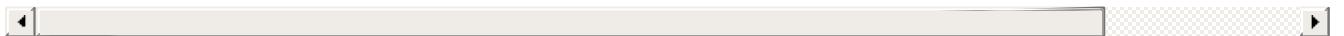
		(Overrides <a href="#">JsonWriter.WriteLine(B)</a> )
≡ ♦	<a href="#">WriteValue(Int16)</a>	Writes a <a href="#">Int16</a> value.
		(Overrides <a href="#">JsonWriter.WriteLine(In)</a> )
≡ ♦	<a href="#">WriteValue(UInt16)</a>	Writes a <a href="#">UInt16</a> value.
		(Overrides <a href="#">JsonWriter.WriteLine(U)</a> )
≡ ♦	<a href="#">WriteValue(Char)</a>	Writes a <a href="#">Char</a> value.
		(Overrides <a href="#">JsonWriter.WriteLine(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.WriteLine(S)</a> )
≡ ♦	<a href="#">WriteValue(Decimal)</a>	Writes a <a href="#">Decimal</a> value.
		(Overrides <a href="#">JsonWriter.WriteLine(D)</a> )
≡ ♦	<a href="#">WriteValue(DateTime)</a>	Writes a <a href="#">DateTime</a> value
		(Overrides <a href="#">JsonWriter.WriteLine(D)</a> )
≡ ♦	<a href="#">WriteValue(Byte[])</a>	Writes a <a href="#">Byte[]</a> value.
		(Overrides <a href="#">JsonWriter.WriteLine(B)</a> )
≡ ♦	<a href="#">WriteValue(DateTimeOffset)</a>	Writes a <a href="#">DateTimeOffset</a>
		(Overrides <a href="#">JsonWriter.WriteLine(DT)</a> )

	<a href="#">JsonWriter.WriteLine(D)</a>	
≡	<a href="#">JsonValue(Nullable&lt;Int32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">JsonValue(Nullable&lt;UInt32&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">JsonValue(Nullable&lt;Int64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">JsonValue(Nullable&lt;UInt64&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">JsonValue(Nullable&lt;Single&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">JsonValue(Nullable&lt;Double&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">JsonValue(Nullable&lt;Boolean&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">JsonValue(Nullable&lt;Int16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">JsonValue(Nullable&lt;UInt16&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">JsonValue(Nullable&lt;Char&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡	<a href="#">JsonValue(Nullable&lt;Byte&gt;)</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> val
≡		

	<a href="#"><u>WriteValue(Nullable&lt;SByte&gt;)</u></a>	Writes a <a href="#"><u>Nullable&lt;T&gt;</u></a> value.
≡	<a href="#"><u>WriteValue(Nullable&lt;Decimal&gt;)</u></a>	Writes a <a href="#"><u>Nullable&lt;T&gt;</u></a> value.
≡	<a href="#"><u>WriteValue(Nullable&lt;DateTime&gt;)</u></a>	Writes a <a href="#"><u>Nullable&lt;T&gt;</u></a> value.
≡	<a href="#"><u>WriteValue(Nullable&lt;DateTimeOffset&gt;)</u></a>	Writes a <a href="#"><u>Nullable&lt;T&gt;</u></a> value.
≡	<a href="#"><u>WriteValue(Object)</u></a>	Writes a <a href="#"><u>Object</u></a> value. An exception is raised if the value cannot be converted to a single JSON token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
 (4.0.2.13623)



# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **WriteValue(Boolean)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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(  
    unsigned char value  
) override
```

## Parameters

### **value (Byte)**

The [Byte](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **WriteValue(Char)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **WriteValue(DateTime)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ►  
**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: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **WriteValue(Int64)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **WriteValue(SByte)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **WriteValue(String)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **WriteValue(UInt32)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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: 4.0.2.0  
(4.0.2.13623)

## Method

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

**WriteValueDelimiter()**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# WriteWhitespace Method (ws)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonTextWriter](#) ► **WriteWhitespace(String)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

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 interger.
<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: 4.0.2.C  
(4.0.2.13623)

# 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

Instance

Declared

Protected

Static

Inherited

Icon	Member	Description
	<a href="#">JsonValidatingReader(JsonReader)</a>	Initializes a new instance of <b>JsonValidatingReader</b> class that validates the content returned by the given <a href="#">JsonReader</a> .
	<a href="#">Close()</a>	Changes the <a href="#">JsonReader.State</a> 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="#">CurrentState</a>	Gets the current reader state.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">Depth</a>	Gets the depth of the current node in the JSON document.  (Overrides <a href="#">JsonReader.Depth</a> .)
	<a href="#">Dispose(Boolean)</a>	Releases unmanaged and 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 instance.  (Inherited from <a href="#">Object</a> .)
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to free resources and perform other 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 this <a href="#">Object</a> .

		(Inherited from <a href="#">Object</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>	Reads the next JSON token from the stream as a <b>Byte[]</b> .  (Overrides <a href="#">JsonReader.ReadAsBytes</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="#">Reader</a>	Gets the <a href="#">JsonReader</a> used to validate 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 node. (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 JsonToken. (Overrides <a href="#">JsonReader.ValueType</a> .)

## [-] Inheritance Hierarchy

[Object](#)

└ [JsonReader](#)

  └ **JsonValidatingReader**

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



# JsonValidatingReader

## Constructor (reader)

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

[JsonValidatingReader\(JsonReader\)](#)

C#

Initializes a new instance of the [JsonValidatingReader](#) class that validates the content returned from the given [JsonReader](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonValidatingReader(  
    JsonReader reader  
)
```

```
Public Sub New ( _  
    reader As JsonReader _  
)
```

```
public:  
JsonValidatingReader(  
    JsonReader^ reader  
)
```

### Parameters

**reader ([JsonReader](#))**

The [JsonReader](#) to read from while validating.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# Depth Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ► **Depth**

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: 4.0.2.0 (4.0.2.13623)

# QuoteChar Property

[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 int
```

```
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) o
}
```

## Value

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# Read Method

[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: 4.0.2.C  
(4.0.2.13623)

# ReadAsBytes Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ► **ReadAsBytes()**

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# ReadAsDateTimeOffset

## Method

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

**ReadAsDateTimeOffset()**

C#

Reads the next JSON token from the stream as a [Nullable<T>](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<DateTimeOffset> ReadAsDateTimeOffset()
```

```
Public Overrides Function ReadAsDateTimeOffset As Nullable<DateTimeOffset>
```

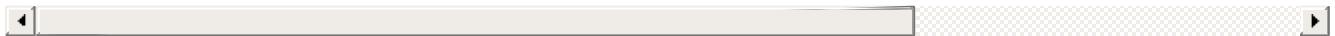
```
public:  
virtual Nullable<DateTimeOffset> ReadAsDateTimeOffset();
```

### Return Value

A [Nullable<T>](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# ReadAsDecimal Method

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonValidatingReader](#) ► [ReadAsDecimal\(\)](#)

C# ▾

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>](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# Reader Property

[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: 4.0.2.0 (4.0.2.13623)

C#  
▼

Gets or sets the schema.

## Declaration Syntax

C#

Visual Basic

Visual C++  
▼

```
public JsonSchema Schema { get; set; }
```

```
Public Property Schema As JsonSchema
    Get
    Set
```

```
public:
property JsonSchema^ Schema {
    JsonSchema^ get ();
    void set (JsonSchema^ value);
}
```

## Value

The schema.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# TokenType Property

[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: 4.0.2.0 (4.0.2.13623)



# ValidationEventHandler

## Event

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

### ValidationEventHandler

C#

Sets an event handler for receiving schema validation errors.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public event ValidationEventHandler ValidationEventH
```

```
Public Event ValidationEventHandler As ValidationEve
```

```
public:  
    event ValidationEventHandler^ ValidationEventHandler;  
        void add (ValidationEventHandler^ value);  
        void remove (ValidationEventHandler^ value);  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)





Gets the text value of the current Json token.

## Declaration Syntax

```
public override Object Value { get; }
```

```
Public Overrides ReadOnly Property Value As Object
    Get
```

```
public:
virtual property Object^ Value {
    Object^ get () override;
}
```

## Value

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

## ValueType

C#  
▼

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: 4.0.2.0 (4.0.2.13623)

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"/> Inherit

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

		<p><a href="#">TextReader</a> should be closed when the writer is closed.</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="#">Flush()</a>	Flushes whatever is in the buffer to the underlying streams and also flushes the underlying stream.
📝	<a href="#">Formatting</a>	Indicates how the output is formatted.
≡	<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="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<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 containing the specified text.  <b>C#</b> <a href="#">Copy</a> /* . . . */
	<a href="#">WriteEnd()</a>	Writes the end of the current Json object or array.
	<a href="#">WriteEnd(JsonToken)</a>	Writes the specified end token.
	<a href="#">WriteEndArray()</a>	Writes the end of an array.
	<a href="#">WriteEndConstructor()</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="#"><u>WriteRaw(String)</u></a>	Writes raw JSON without changing the writer's state.
≡	<a href="#"><u>WriteRawValue(String)</u></a>	Writes raw JSON where value is expected and updates the writer's state
≡	<a href="#"><u>WriteStartArray()</u></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="#"><u>WriteValue(Single)</u></a>	Writes a <a href="#"><u>Single</u></a> value.
≡ ♡	<a href="#"><u>WriteValue(Double)</u></a>	Writes a <a href="#"><u>Double</u></a> value.
≡ ♡	<a href="#"><u>WriteValue(Boolean)</u></a>	Writes a <a href="#"><u>Boolean</u></a> value.
≡ ♡	<a href="#"><u>WriteValue(Int16)</u></a>	Writes a <a href="#"><u>Int16</u></a> value.
≡ ♡	<a href="#"><u>WriteValue(UInt16)</u></a>	Writes a <a href="#"><u>UInt16</u></a> value.
≡ ♡	<a href="#"><u>WriteValue(Char)</u></a>	Writes a <a href="#"><u>Char</u></a> value.
≡ ♡	<a href="#"><u>WriteValue(Byte)</u></a>	Writes a <a href="#"><u>Byte</u></a> value.
≡ ♡	<a href="#"><u>WriteValue(SByte)</u></a>	Writes a <a href="#"><u>SByte</u></a> value.
≡ ♡	<a href="#"><u>WriteValue(Decimal)</u></a>	Writes a <a href="#"><u>Decimal</u></a> value.
≡ ♡	<a href="#"><u>WriteValue(DateTime)</u></a>	Writes a <a href="#"><u>DateTime</u></a> value
≡ ♡	<a href="#"><u>WriteValue(DateTimeOffset)</u></a>	Writes a <a href="#"><u>DateTimeOffset</u></a> value.

=♪	<a href="#">WriteValue&lt;Nullable&lt;Int32&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
=♪	<a href="#">WriteValue&lt;Nullable&lt;UInt32&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
=♪	<a href="#">WriteValue&lt;Nullable&lt;Int64&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
=♪	<a href="#">WriteValue&lt;Nullable&lt;UInt64&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
=♪	<a href="#">WriteValue&lt;Nullable&lt;Single&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
=♪	<a href="#">WriteValue&lt;Nullable&lt;Double&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
=♪	<a href="#">WriteValue&lt;Nullable&lt;Boolean&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
=♪	<a href="#">WriteValue&lt;Nullable&lt;Int16&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
=♪	<a href="#">WriteValue&lt;Nullable&lt;UInt16&gt;&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(Byte[])</a>	Writes a <b>Byte[]</b> 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 whitespace.

## [-] Inheritance Hierarchy

[Object](#)

└ **JsonWriter**

  └ [BsonWriter](#)

  └ [JsonTextWriter](#)

  └ [JTokenWriter](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

# CloseOutput Property

[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: 4.0.2.0  
(4.0.2.13623)

# Flush Method

[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: 4.0.2.C  
(4.0.2.13623)

# Formatting Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► [Formatting](#)

C# ▾

Indicates how the 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: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)

# WriteComment Method (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.

## - 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: 4.0.2.0 (4.0.2.13623)

▾

Writes the end of the current Json object or array.

## Declaration Syntax

[C#](#)[Visual Basic](#)[Visual C++](#)

```
public void WriteEnd()
```

```
Public Sub WriteEnd
```

```
public:  
void WriteEnd()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteEnd(JsonToken)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)



Writes the end of an array.

## Declaration Syntax

```
public void WriteEndArray()
```

```
Public Sub WriteEndArray
```

```
public:  
void WriteEndArray()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# WriteEndConstructor

## Method

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

### WriteEndConstructor()

C# ▾

Writes the end constructor.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public void WriteEndConstructor()
```

```
Public Sub WriteEndConstructor
```

```
public:  
void WriteEndConstructor()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

▼

Writes the end of a Json object.

## Declaration Syntax

[C#](#)[Visual Basic](#)[Visual C++](#)

```
public void WriteEndObject()
```

```
Public Sub WriteEndObject
```

```
public:  
void WriteEndObject()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

# WriteIndentSpace

## 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: 4.0.2.0  
(4.0.2.13623)



Writes a null value.

## Declaration Syntax

```
public virtual void WriteNull()
```

```
Public Overridable Sub WriteNull
```

```
public:  
virtual void WriteNull()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# WritePropertyName

## Method (name)

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

### WritePropertyName(String)

C# ▾

Writes the property name of a name/value pair on a Json object.

#### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public virtual void WritePropertyName(  
    string name  
)
```

```
Public Overridable Sub WritePropertyName ( _  
    name As String _  
)
```

```
public:  
virtual void WritePropertyName(  
    String^ name  
)
```

#### Parameters

##### **name (String)**

The name of the property.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

C# ▾

Writes raw JSON without changing the writer's state.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteRaw(  
    string json  
)
```

```
Public Overridable Sub WriteRaw ( _  
    json As String _  
)
```

```
public:  
virtual void WriteRaw(  
    String^ json  
)
```

## Parameters

### *json* ([String](#))

The raw JSON to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# WriteRawValue Method (json)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteRawValue(String)**

C#

Writes raw JSON where a value is expected and updates the writer's state.

## 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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

# WriteStartConstructor

## Method (name)

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

**WriteStartConstructor(String)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# WriteState Property

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteState**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteToken Method (reader)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteToken(JsonReader)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)



Writes an undefined value.

## Declaration Syntax

```
public virtual void WriteUndefined()
```

```
Public Overridable Sub WriteUndefined
```

```
public:  
virtual void WriteUndefined()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

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(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&lt;Nullable&lt;Single&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
≡ ♦	<a href="#">WriteValue&lt;Nullable&lt;Double&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
≡ ♦	<a href="#">WriteValue&lt;Nullable&lt;Boolean&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
≡ ♦	<a href="#">WriteValue&lt;Nullable&lt;Int16&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
≡ ♦	<a href="#">WriteValue&lt;Nullable&lt;UInt16&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
≡ ♦	<a href="#">WriteValue&lt;Nullable&lt;Char&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
≡ ♦	<a href="#">WriteValue&lt;Nullable&lt;Byte&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
≡ ♦	<a href="#">WriteValue&lt;Nullable&lt;SByte&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.
≡ ♦	<a href="#">WriteValue&lt;Nullable&lt;Decimal&gt;&gt;</a>	Writes a <a href="#">Nullable&lt;T&gt;</a> value.

≡	<a href="#"><u>WriteValue(Nullable&lt;DateTime&gt;)</u></a>	Writes a <a href="#"><u>Nullable&lt;T&gt;</u></a> value.
≡	<a href="#"><u>WriteValue(Nullable&lt;DateTimeOffset&gt;)</u></a>	Writes a <a href="#"><u>Nullable&lt;T&gt;</u></a> value.
≡	<a href="#"><u>WriteValue(Byte[])</u></a>	Writes a <b>Byte[]</b> value.
≡	<a href="#"><u>WriteValue(Object)</u></a>	Writes a <a href="#"><u>Object</u></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: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Boolean)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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: 4.0.2.C  
(4.0.2.13623)

# WriteValue Method (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: 4.0.2.C  
(4.0.2.13623)

# WriteValue Method (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: 4.0.2.C  
(4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(DateTime)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ►  
**WriteValue(DateTimeOffset)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Decimal)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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: 4.0.2.C  
(4.0.2.13623)

# WriteValue Method (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: 4.0.2.C  
(4.0.2.13623)

# WriteValue Method (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: 4.0.2.C  
(4.0.2.13623)

# WriteValue Method (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: 4.0.2.C  
(4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Nullable<Boolean>)**

C# ▾

Writes a [Nullable<T>](#) value.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<bool> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable(Of Boolean) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<bool> value  
)
```

## Parameters

### **value** ([Nullable<Boolean>](#))

The [Nullable<T>](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Nullable<Byte>)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

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

**WriteValue(Nullable<Char>)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Nullable<DateTime>)**

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

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

**WriteValue(Nullable<DateTimeOffset>)**

C# ▾

Writes a [Nullable<T>](#) value.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public virtual void WriteValue(  
    Nullable<DateTimeOffset> value  
)
```

```
Public Overridable Sub WriteValue ( _  
    value As Nullable(Of DateTimeOffset) _  
)
```

```
public:  
virtual void WriteValue(  
    Nullable<DateTimeOffset> value  
)
```

## Parameters

**value ([Nullable<DateTimeOffset>](#))**

The [Nullable<T>](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (value)

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

**WriteValue(Nullable<Decimal>)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (value)

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

**WriteValue(Nullable<Double>)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (value)

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

**WriteValue(Nullable<Int16>)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (value)

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

**WriteValue(Nullable<Int32>)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (value)

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

**WriteValue(Nullable<Int64>)**

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (value)

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

**WriteValue(Nullable<SByte>)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (value)

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

**WriteValue(Nullable<Single>)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (value)

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

**WriteValue(Nullable<UInt16>)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (value)

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

**WriteValue(Nullable<UInt32>)**

C# ▾

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 UInteger) _  
)
```

```
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: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (value)

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

**WriteValue(Nullable<UInt64>)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json](#) ► [JsonWriter](#) ► **WriteValue(Object)**

C# ▾

Writes a [Object](#) value. An error will 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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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: 4.0.2.C  
(4.0.2.13623)

# WriteValue Method (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: 4.0.2.C  
(4.0.2.13623)

# WriteValue Method (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: 4.0.2.C  
(4.0.2.13623)

# WriteValue Method (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: 4.0.2.C  
(4.0.2.13623)

# WriteValue Method (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: 4.0.2.C  
(4.0.2.13623)

# WriteValue Method (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: 4.0.2.C  
(4.0.2.13623)

# WriteValueDelimiter

## Method

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

### WriteValueDelimiter()

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# WriteWhitespace Method (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: 4.0.2.0  
(4.0.2.13623)

# 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++

```
public class JsonWriterException : Exception
```

```
Public Class JsonWriterException _
    Inherits Exception
```

```
public ref class JsonWriterException : public Except
```

### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherit

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="#">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



---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonWriterException Constructor

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

**JsonWriterException()**

C# ▾

## 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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# JsonWriterException Constructor

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

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonWriterException()
```

```
Public Sub New
```

```
public:  
JsonWriterException()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonWriterException

## Constructor (message)

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

**JsonWriterException(String)**

C#

Initializes a new instance of the [JsonWriterException](#) class with a specified error message.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonWriterException(  
    string message  
)
```

```
Public Sub New ( _  
    message As String _  
)
```

```
public:  
JsonWriterException(  
    String^ message  
)
```

### Parameters

#### **message (String)**

The error message that explains the reason for the exception.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonWriterException

## Constructor (message, innerException)

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

**JsonWriterException(String, Exception)**

C# ▾

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++

```
public JsonWriterException(
    string message,
    Exception innerException
)
```

```
Public Sub New ( _
    message As String, _
    innerException As Exception _
)
```

```
public:
JsonWriterException(
    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: 4.0.2.C  
(4.0.2.13623)

# MemberSerialization

## Enumeration

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

C# ▾

Specifies the member serialization options for the [JsonSerializer](#).

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public enum MemberSerialization
```

```
Public Enumeration MemberSerialization
```

```
public enum class MemberSerialization
```

### Members

Member	Description
<b>OptOut</b>	All members are serialized by default. Members can be excluded using the <a href="#">JsonIgnoreAttribute</a> .
<b>OptIn</b>	Only members must be marked with the <a href="#">JsonPropertyAttribute</a> are serialized.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

## 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: 4.0.2.0  
(4.0.2.13623)

## 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: 4.0.2.0 (4.0.2.13623)

## Enumeration

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

C#

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 objects when needed.
<b>Reuse</b>	Only reuse existing objects.
<b>Replace</b>	Always create new objects.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# Documentation **PreserveReferencesHandling** Enumeration

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

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

## Enumeration

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

C#

Specifies reference loop handling options for the [JsonSerializer](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public enum ReferenceLoopHandling
```

```
Public Enumeration ReferenceLoopHandling
```

```
public enum class ReferenceLoopHandling
```

### Members

Member	Description
Error	Throw a <a href="#">JsonSerializationException</a> where a loop is encountered.
Ignore	Ignore loop references and do not serialize.
Serialize	Serialize loop references.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

## Enumeration

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

C# ▾

Specifies type name handling options for the [JsonSerializer](#).

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
[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>Auto</b>	Include the .NET type name when the type of the object being serialized is not the same as its declared type.
<b>All</b>	Always include the .NET type name when serializing.

(4.0.2.13623)

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
Error	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.
Closed	The <a href="#">Close()</a> method has been called.
Object	An object is being written.
Array	A array is being written.
Constructor	A constructor is being written.
Property	A property is being written.
Start	A write method has not been called.

## 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.

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

Declared

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 <u>Object</u> is equal to the current <u>Object</u> .  (Inherited from <u>Object</u> .)
💡	<a href="#">Finalize()</a>	Allows an <u>Object</u> to attempt to free resources and perform other cleanup operations before the <u>Object</u> is reclaimed by garbage collection.  (Inherited from <u>Object</u> .)

	<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 Oid.

## Inheritance Hierarchy

[Object](#)

  └ **BsonObjectId**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C (4.0.2.13623)

# BsonObjectId

## Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonObjectId](#) ►

**BsonObjectId(Byte[])**

C# ▾

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

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public BsonObjectId(  
    byte[] value  
)
```

```
Public Sub New (  
    value As Byte() _  
)
```

```
public:  
BsonObjectId(  
    array<unsigned char>^ value  
)
```

### Parameters

#### **value (Byte[])**

The Oid value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



Gets or sets the value of the Oid.

## Declaration Syntax

```
public byte[] Value { get; private set; }
```

```
Public Property Value As Byte()
    Get
    Private Set
```

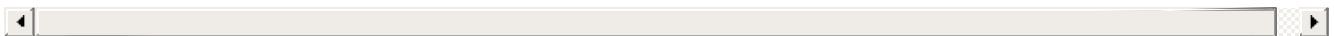
```
public:
property array<unsigned char>^ Value {
    array<unsigned char>^ get ();
    private: void set (array<unsigned char>^ val
}
```

## Value

The value of the Oid.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



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"/> 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="#">BsonReader(Stream)</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="#">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 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="#">CurrentState</a>	Gets the current reader state.  (Inherited from <a href="#">JsonReader</a> .)
	<a href="#">DateTimeKindHandling</a>	Gets or sets the <a href="#">DateTimeKind</a> used when reading <a href="#">DateTime</a> values from BSON.
	<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 comply with incorrect Json.NET 3.5 writing of binary.
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <a href="#">Object</a> . (Inherited from <a href="#">Object</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 stream. (Overrides <a href="#">JsonReader.Read()</a> .)
	<a href="#">ReadAsBytes()</a>	Reads the next JSON token from stream as a <a href="#">Byte[]</a> . (Overrides <a href="#">JsonReader.ReadAsBytes()</a> .)
	<a href="#">ReadAsDateTimeOffset()</a>	Reads the next JSON token from 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 stream as a <a href="#">Nullable&lt;T&gt;</a> . (Overrides <a href="#">JsonReader.ReadAsDecimal()</a> .)

	<a href="#">ReadRootValueAsArray</a>	Gets or sets a value indicating whether the root object will be read as a JavaScript array.
	<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 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: 4.0.2.0  
(4.0.2.13623)



# BsonReader Constructor

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

[BsonReader\(\)](#)

C# ▾

## Members

Icon	Member	Description
	<a href="#">BsonReader(Stream)</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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# BsonReader Constructor (*stream*)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ► [BsonReader\(Stream\)](#)

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# 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: 4.0.2.0  
(4.0.2.13623)

## Close Method

[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: 4.0.2.0  
(4.0.2.13623)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

### DateTimeKindHandling

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

C# ▾

Gets or sets a value indicating whether binary data reading should be 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: 4.0.2.0  
(4.0.2.13623)

# Read Method

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ► **Read()**

C# ▾

Reads the next JSON token from the stream.

## Declaration Syntax

C#

Visual Basic

Visual C++  
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: 4.0.2.C  
(4.0.2.13623)

# ReadAsBytes Method

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

## ReadAsBytes()

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# ReadAsDateTimeOffset

## Method

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

[ReadAsDateTimeOffset\(\)](#)

C#

Reads the next JSON token from the stream as a [Nullable<T>](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<DateTimeOffset> ReadAsDateTimeOffset()
```

```
Public Overrides Function ReadAsDateTimeOffset As Nullable<DateTimeOffset>
```

```
public:  
virtual Nullable<DateTimeOffset> ReadAsDateTimeOffset();
```

### Return Value

A [Nullable<T>](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# ReadAsDecimal Method

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ►

## ReadAsDecimal()

C# ▾

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>](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



## Property

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonReader](#) ► [ReadRootValueAsArray](#)

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

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"/> Inherit

Icon	Member	Description
	<a href="#">BsonWriter(Stream)</a>	Initializes a new instance of the <b>BsonWriter</b> class.
	<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 the underlying stream or its output should be closed when the <b>BsonWriter</b> object is closed.

		(Inherited from <a href="#">JsonWriter</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> , an exception will occur.
	<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 operations before the <a href="#">Object</a> is released by garbage collection.  (Inherited from <a href="#">Object</a> .)
	<a href="#">Flush()</a>	Flushes whatever is in the underlying streams and closes the underlying stream.  (Overrides <a href="#">JsonWriter.Flush()</a> .)
	<a href="#">Formatting</a>	Indicates how the output is formatted.  (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">GetHashCode()</a>	Serves as a hash function for a <a href="#">Type</a> object of a specific 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="#">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.WriteLine</a> )
	<a href="#">WriteEnd()</a>	Writes the end of the current cursor 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 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.WriteLine()</a> )
💡	<a href="#">WriteObjectId(Byte[])</a>	Writes a <b>Byte[]</b> value that represents the BSON object id.
💡	<a href="#">WritePropertyName(String)</a>	Writes the property name and value pair on a JsonWriter.  (Overrides <a href="#">JsonWriter.WriteProperty()</a> )
💡	<a href="#">WriteRaw(String)</a>	Writes raw JSON.  (Overrides <a href="#">JsonWriter.WriteLine()</a> )
💡	<a href="#">WriteRawValue(String)</a>	Writes raw JSON where the type is expected and updates the current position.  (Overrides <a href="#">JsonWriter.WriteRawValue()</a> )
💡	<a href="#">WriteRegex(String, String)</a>	Writes a BSON regex.
💡	<a href="#">WriteStartArray()</a>	Writes the beginning of an array.  (Overrides <a href="#">JsonWriter.WriteLine()</a> )
💡	<a href="#">WriteStartConstructor(String)</a>	Writes the start of a constructor with the given name.  (Overrides <a href="#">JsonWriter.WriteStartConstructor()</a> )

	<a href="#">WriteStartObject()</a>	Writes the beginning of an object.  (Overrides <a href="#">JsonWriter.WriteStartObject</a> )
	<a href="#">WriteState</a>	Gets the state of the writer.  (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteToken(JsonReader)</a>	Writes the current <a href="#">JsonReader</a> token.  (Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteUndefined()</a>	Writes an undefined value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
	<a href="#">WriteValue(String)</a>	Writes a <a href="#">String</a> value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
	<a href="#">WriteValue(Int32)</a>	Writes a <a href="#">Int32</a> value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
	<a href="#">WriteValue(UInt32)</a>	Writes a <a href="#">UInt32</a> value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
	<a href="#">WriteValue(Int64)</a>	Writes a <a href="#">Int64</a> value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
	<a href="#">WriteValue(UInt64)</a>	Writes a <a href="#">UInt64</a> value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
	<a href="#">WriteValue(Single)</a>	Writes a <a href="#">Single</a> value.  (Overrides <a href="#">JsonWriter.WriteLine</a> )
		

	<a href="#">WriteValue(Double)</a>	Writes a <a href="#">Double</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(D)</a> )
≡	<a href="#">WriteValue(Boolean)</a>	Writes a <a href="#">Boolean</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(B)</a> )
≡	<a href="#">WriteValue(Int16)</a>	Writes a <a href="#">Int16</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(W)</a> )
≡	<a href="#">WriteValue(UInt16)</a>	Writes a <a href="#">UInt16</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(U)</a> )
≡	<a href="#">WriteValue(Char)</a>	Writes a <a href="#">Char</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(W)</a> )
≡	<a href="#">WriteValue(Byte)</a>	Writes a <a href="#">Byte</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(W)</a> )
≡	<a href="#">WriteValue(SByte)</a>	Writes a <a href="#">SByte</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(S)</a> )
≡	<a href="#">WriteValue(Decimal)</a>	Writes a <a href="#">Decimal</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(D)</a> )
≡	<a href="#">WriteValue(DateTime)</a>	Writes a <a href="#">DateTime</a> value  (Overrides <a href="#">JsonWriter.WriteLine(D)</a> )
≡	<a href="#">WriteValue(DateTimeOffset)</a>	Writes a <a href="#">DateTimeOffset</a>  (Overrides

	<a href="#">JsonWriter.WriteLine(D)</a>
≡	<a href="#">WriteValue(Byte[])</a>
	Writes a <b>Byte[]</b> value.  (Overrides <a href="#">JsonWriter.WriteLine(Byte[])</a> )
≡	<a href="#">WriteValue(Nullable&lt;Int32&gt;)</a>
	Writes a <a href="#">Nullable&lt;T&gt;</a> value.  (Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;UInt32&gt;)</a>
	Writes a <a href="#">Nullable&lt;T&gt;</a> value.  (Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;Int64&gt;)</a>
	Writes a <a href="#">Nullable&lt;T&gt;</a> value.  (Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;UInt64&gt;)</a>
	Writes a <a href="#">Nullable&lt;T&gt;</a> value.  (Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;Single&gt;)</a>
	Writes a <a href="#">Nullable&lt;T&gt;</a> value.  (Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;Double&gt;)</a>
	Writes a <a href="#">Nullable&lt;T&gt;</a> value.  (Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;Boolean&gt;)</a>
	Writes a <a href="#">Nullable&lt;T&gt;</a> value.  (Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;Int16&gt;)</a>
	Writes a <a href="#">Nullable&lt;T&gt;</a> value.  (Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;UInt16&gt;)</a>
	Writes a <a href="#">Nullable&lt;T&gt;</a> value.  (Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;Char&gt;)</a>
	Writes a <a href="#">Nullable&lt;T&gt;</a> value.  (Inherited from <a href="#">JsonWriter</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(Object)</a>	Writes a <a href="#">Object</a> value. As 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**



# BsonWriter Constructor (*stream*)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ► [BsonWriter\(Stream\)](#)

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# Close Method

[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: 4.0.2.0  
(4.0.2.13623)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### DateTimeKindHandling

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# Flush Method

[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: 4.0.2.C  
(4.0.2.13623)

# WriteComment Method (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.

## 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: 4.0.2.0 (4.0.2.13623)

▼

Writes a null value.

## Declaration Syntax

```
public override void WriteNull()
```

```
Public Overrides Sub WriteNull
```

```
public:  
virtual void WriteNull() override
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# WriteObjectId Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►  
**WriteObjectId(Byte[])**

C# ▾

Writes a **Byte[]** value that represents a BSON object id.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public void WriteObjectId(  
    byte[] value  
)
```

```
Public Sub WriteObjectId ( _  
    value As Byte() _  
)
```

```
public:  
void WriteObjectId(  
    array<unsigned char>^ value  
)
```

## Parameters

**value (Byte[])**

[Missing <param name="value"/> documentation for  
"M:Newtonsoft.Json.Bson.BsonWriter.WriteObjectId(System.B

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# WritePropertyName

## Method (name)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

### WritePropertyName(String)

C#

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: 4.0.2.0 (4.0.2.13623)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

## WriteRaw(String)

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteRawValue Method (json)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

**WriteRawValue(String)**

C#

Writes raw JSON where a value is expected and updates the writer's state.

## 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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# WriteRegex Method (pattern, options)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

**WriteRegex(String, String)**

C# ▾

Writes a BSON regex.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public void WriteRegex(  
    string pattern,  
    string options  
)
```

```
Public Sub WriteRegex ( _  
    pattern As String, _  
    options As String _  
)
```

```
public:  
void WriteRegex(  
    String^ pattern,  
    String^ options  
)
```

## Parameters

### **pattern ([String](#))**

The regex pattern.

### **options ([String](#))**

The regex options.

# WriteStartArray Method

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

## WriteStartArray()

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# WriteStartConstructor

## Method (name)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

**WriteStartConstructor(String)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteStartObject Method

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

## WriteStartObject()

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# WriteUndefined Method

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

## WriteUndefined()

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

C# ▾

**Members**

Icon	Member	Description
≡	<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.

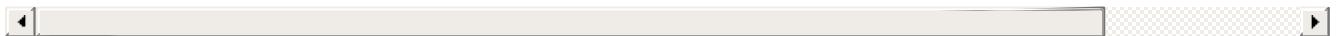
		(Overrides <a href="#">JsonWriter.WriteLine(B</a> )
≡	<a href="#">WriteValue(Int16)</a>	Writes a <a href="#">Int16</a> value.
≡	<a href="#">WriteValue(UInt16)</a>	(Overrides <a href="#">JsonWriter.WriteLine(In</a> )
≡	<a href="#">WriteValue(Char)</a>	Writes a <a href="#">UInt16</a> value.
≡	<a href="#">WriteValue(Byte)</a>	(Overrides <a href="#">JsonWriter.WriteLine(U</a> )
≡	<a href="#">WriteValue(SByte)</a>	Writes a <a href="#">Char</a> value.
≡	<a href="#">WriteValue(Decimal)</a>	(Overrides <a href="#">JsonWriter.WriteLine(C</a> )
≡	<a href="#">WriteValue(DateTime)</a>	Writes a <a href="#">Byte</a> value.
≡	<a href="#">WriteValue(DateTimeOffset)</a>	(Overrides <a href="#">JsonWriter.WriteLine(W</a> )
≡	<a href="#">WriteValue(Byte[])</a>	Writes a <a href="#">SByte</a> value.
≡		(Overrides <a href="#">JsonWriter.WriteLine(S</a> )
≡		Writes a <a href="#">Decimal</a> value.
≡		(Overrides <a href="#">JsonWriter.WriteLine(D</a> )
≡		Writes a <a href="#">DateTime</a> value.
≡		(Overrides <a href="#">JsonWriter.WriteLine(D</a> )
≡		Writes a <a href="#">DateTimeOffset</a> value.
≡		(Overrides <a href="#">JsonWriter.WriteLine(D</a> )
≡		Writes a <a href="#">Byte[]</a> value.
		(Overrides <a href="#">JsonWriter.WriteLine(B</a> )

	<a href="#">JsonWriter.WriteLine(B)</a>	
≡	<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="#"><u>WriteValue(Nullable&lt;SByte&gt;)</u></a>	Writes a <a href="#"><u>Nullable&lt;T&gt;</u></a> value.
≡	<a href="#"><u>WriteValue(Nullable&lt;Decimal&gt;)</u></a>	Writes a <a href="#"><u>Nullable&lt;T&gt;</u></a> value.
≡	<a href="#"><u>WriteValue(Nullable&lt;DateTime&gt;)</u></a>	Writes a <a href="#"><u>Nullable&lt;T&gt;</u></a> value.
≡	<a href="#"><u>WriteValue(Nullable&lt;DateTimeOffset&gt;)</u></a>	Writes a <a href="#"><u>Nullable&lt;T&gt;</u></a> value.
≡	<a href="#"><u>WriteValue(Object)</u></a>	Writes a <a href="#"><u>Object</u></a> value. An exception is raised if the value cannot be converted to a single JSON token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
 (4.0.2.13623)



# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ► **WriteValue(Boolean)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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(  
    unsigned char value  
) override
```

## Parameters

### **value (Byte)**

The [Byte](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ► **WriteValue(Char)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ► **WriteValue(DateTime)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ► **WriteValue(Int64)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ► **WriteValue(SByte)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ► **WriteValue(String)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ► **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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Bson](#) ► [BsonWriter](#) ►

**WriteValue(UInt32)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# WriteValue Method (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: 4.0.2.0  
(4.0.2.13623)

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

Enumerations

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="#"><b>DataTableConverter</b></a>	Converts a <a href="#">DataTable</a> to and from JSON.
	<a href="#"><b>DateTimeConverterBase</b></a>	Provides a base class for converting a <a href="#">DateTime</a> to and from JSON.
	<a href="#"><b>EntityKeyMemberConverter</b></a>	Converts an Entity Framework EntityKey to and from JSON.
	<a href="#"><b>ExpandoObjectConverter</b></a>	Converts an ExpandoObject to and from JSON.
	<a href="#"><b>IsoDateTimeConverter</b></a>	Converts a <a href="#">DateTime</a> to and from the ISO 8601 date format (e.g. 2008-04-12T12:53Z).
	<a href="#"><b>JavaScriptDateTimeConverter</b></a>	Converts a <a href="#">DateTime</a> to and from a JavaScript date constructor (e.g. new Date(52231943)).
	<a href="#"><b>JsonDateTimeSerializationMode</b></a>	Specifies whether a DateTime object represents a local time, a Coordinated Universal Time (UTC), or is not specified as either local time or UTC.

	<a href="#"><b>KeyValuePairConverter</b></a>	Converts a <a href="#"><b>KeyValuePair&lt;TKey, TValue&gt;</b></a> to and from JSON.
	<a href="#"><b>RegexConverter</b></a>	Converts a <a href="#"><b>Regex</b></a> to and from JSON and BSON.
	<a href="#"><b>StringEnumConverter</b></a>	Converts an <a href="#"><b>Enum</b></a> to and from its name string value. Converts an <a href="#"><b>Enum</b></a> to and from its name string value.
	<a href="#"><b>XmlNodeConverter</b></a>	Converts XML to and from JSON.

# BinaryConverter Class

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► **BinaryConverter**

C# ▾

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 JsonConverter
```

## 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="#">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 JsonConverter.  (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="#"><u>WriteJson(JsonWriter, Object, JsonSerializer)</u></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: 4.0.2.0  
(4.0.2.13623)

# BinaryConverter

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [BinaryConverter](#) ► [BinaryConverter\(\)](#)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# CanConvert Method (*objectType*)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [BinaryConverter](#) ► [CanConvert\(Type\)](#)

C#

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.

# ReadJson Method (*reader*, *objectType*, *existingValue*, *serializer*)

[Namespaces](#) ► [Newtonsoft.Json.Converts](#) ► [BinaryConverter](#) ►

**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: 4.0.2.C  
(4.0.2.13623)

# WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [BinaryConverter](#) ► **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: 4.0.2.0  
(4.0.2.13623)

# BsonObjectIdConverter

## Class

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

### BsonObjectIdConverter

C# ▾

Converts a [BsonObjectId](#) to and from JSON and BSON.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class BsonObjectIdConverter : JsonConverter
```

```
Public Class BsonObjectIdConverter _
    Inherits JsonConverter
```

```
public ref class BsonObjectIdConverter : public Json
```

#### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static



Declare



Inherit

Icon

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

		<p><a href="#">JsonConverter</a> can read JSON.</p> <p>(Inherited from <a href="#">JsonConverter</a>.)</p>
	<a href="#"><b>CanWrite</b></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="#"><b>Equals(Object)</b></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="#"><b>Finalize()</b></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="#"><b>GetHashCode()</b></a>	<p>Serves as a hash function for a particular type.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#"><b>GetSchema()</b></a>	<p>Gets the <a href="#">JsonSchema</a> of the JSON produced by the JsonConverter.</p> <p>(Inherited from <a href="#">JsonConverter</a>.)</p>
	<a href="#"><b>GetType()</b></a>	<p>Gets the <a href="#">Type</a> of the current instance.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#"><b>MemberwiseClone()</b></a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#"><b>ReadJson(JsonReader, Type, Object, JsonSerializer)</b></a>	<p>Reads the JSON representation of the object.</p> <p>(Overrides</p>

		<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: 4.0.2.0 (4.0.2.13623)



# BsonObjectIdConverter

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►  
[BsonObjectIdConverter](#) ► **BsonObjectIdConverter()**

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

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public BsonObjectIdConverter()
```

```
Public Sub New
```

```
public:  
BsonObjectIdConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# CanConvert Method (*objectType*)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►  
[BsonObjectIdConverter](#) ► **CanConvert(Type)**

C#

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.

# ReadJson Method (**reader**, **objectType**, **existingValue**, **serializer**)

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

[BsonObjectIdConverter](#) ► **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: 4.0.2.C  
(4.0.2.13623)

# WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►  
[BsonObjectIdConverter](#) ► **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: 4.0.2.0  
(4.0.2.13623)

# Documentation **CustomCreationConverter<T> Class**

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

## CustomCreationConverter<T>

C# ▾

Create a custom object

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract class CustomCreationConverter<T> : JsonConverter
```

```
Public MustInherit Class CustomCreationConverter(Of  
    Inherits JsonConverter)
```

```
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	Declare
<input checked="" type="checkbox"/> Public		<input checked="" type="checkbox"/> Instance		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Static		<input checked="" type="checkbox"/> Inherit

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>	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 <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.  (Overrides <a href="#">JsonConverter.CanV</a> )
 	<a href="#">Create(Type)</a>	Creates an object which will be populated by the serializer.
 	<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 JsonConverter.  (Inherited from <a href="#">JsonConverter</a> .)
 	<a href="#">GetType()</a>	Gets the <a href="#">Type</a> of the current inst

		(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: 4.0.2.0 (4.0.2.13623)



## Constructor

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

[CustomCreationConverter<T>](#) ► **CustomCreationConverter<T>()**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# CanConvert Method (*objectType*)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►  
[CustomCreationConverter<T>](#) ► **CanConvert(*Type*)**

C# ▾

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.

# CanWrite Property

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

[CustomCreationConverter<T>](#) ► **CanWrite**

C# ▾

Gets a value indicating whether this [JsonConverter](#) can write JSON.

## 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: 4.0.2.C  
(4.0.2.13623)

# Create Method (*objectType*)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [CustomCreationConverter<T>](#) ► **Create(Type)**

C# ▾

Creates an object which will then be populated by the serializer.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract T Create(  
    Type objectType  
)
```

```
Public MustOverride Function Create ( _  
    objectType As Type _  
) As T
```

```
public:  
virtual T Create(  
    Type^ objectType  
) abstract
```

## Parameters

### *objectType* ([Type](#))

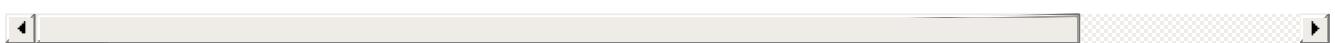
Type of the object.

## Return Value

[Missing <returns> documentation for  
"M:Newtonsoft.Json.Converters.CustomCreationConverter`1.Create"]

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



◀

▶

# ReadJson Method (**reader**, **objectType**, **existingValue**, **serializer**)

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

[CustomCreationConverter<T>](#) ► **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: 4.0.2.C  
(4.0.2.13623)

# WriteJson Method (writer, value, serializer)

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

[CustomCreationConverter<T>](#) ► **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: 4.0.2.0  
(4.0.2.13623)

# DataSetConverter Class

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► **DataSetConverter**

C# ▾

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"/> 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="#">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 JsonConverter.  (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

	<p>current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<p><a href="#"><u>WriteJson(JsonWriter, Object, JsonSerializer)</u></a></p> <p>Writes the JSON representation of the object.</p> <p>(Overrides <a href="#">JsonConverter.WriteJson(JsonWriter, Object, JsonSerializer)</a>.)</p>

## ▀ Inheritance Hierarchy

[Object](#)  
└ [JsonConverter](#)  
  └ **DataSetConverter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# DataSetConverter Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [DataSetConverter](#) ► [DataSetConverter\(\)](#)

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# CanConvert Method (*valueType*)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [DataSetConverter](#) ► [CanConvert\(Type\)](#)

C# ▾

Determines whether this instance can convert the specified value type.

## 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.Converts](#) ► [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: 4.0.2.C  
(4.0.2.13623)

# 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: 4.0.2.0  
(4.0.2.13623)

# DataTableConverter

## Class

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► **DataTableConverter**

C# ▾

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

Public

Protected

Instance

Static

Declare

Inherit

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 <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 JsonConverter.
		(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

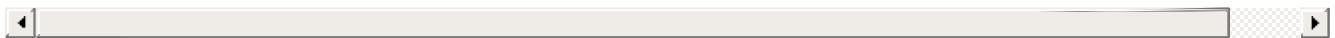
[Object](#)

└ [JsonConverter](#)

  └ [DataTableConverter](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



# DataTableConverter Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [DataTableConverter](#) ► **DataTableConverter()**

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public DataTableConverter()
```

```
Public Sub New
```

```
public:  
DataTableConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# CanConvert Method (*valueType*)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [DataTableConverter](#) ► [CanConvert\(Type\)](#)

C# ▾

Determines whether this instance can convert the specified value type.

## 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.Converts](#) ► [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: 4.0.2.C  
(4.0.2.13623)

# WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [DataTableConverter](#) ►

**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: 4.0.2.0  
(4.0.2.13623)

# 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 : JsonCo
```

```
Public MustInherit Class DateTimeConverterBase _
    Inherits JsonConverter
```

```
public ref class DateTimeConverterBase abstract : pu
```

## Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherit

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 JsonConverter.  (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.  (Inherited from <a href="#">JsonConverter</a> .)

 <a href="#">ToString()</a>  <a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</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> <p>Writes the JSON representation of the object.</p> <p>(Inherited from <a href="#">JsonConverter</a>.)</p>
---	---

## ▀ Inheritance Hierarchy

[Object](#)

└ [JsonConverter](#)

  └ **DateTimeConverterBase**

    └ [IsoDateTimeConverter](#)

    └ [JavaScriptDateTimeConverter](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
 (4.0.2.13623)



# DateTimeConverterBase Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►  
[DateTimeConverterBase](#) ► **DateTimeConverterBase()**

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
protected DateTimeConverterBase()
```

```
Protected Sub New
```

```
protected:  
DateTimeConverterBase()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# CanConvert Method (*objectType*)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►  
[DateTimeConverterBase](#) ► **CanConvert(Type)**

C# ▾

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.

# EntityKeyMemberConverter Class

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

## EntityKeyMemberConverter

C#

Converts an Entity Framework EntityKey to and from JSON.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class EntityKeyMemberConverter : JsonConverter
```

```
Public Class EntityKeyMemberConverter _  
    Inherits JsonConverter
```

```
public ref class EntityKeyMemberConverter : public J
```

### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherit

Icon	Member	Description
	<a href="#">EntityKeyMemberConverter()</a>	Initializes a new instance of the <b>EntityKeyMemberConverter</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 <b>EntityKeyMemberConverter</b> 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 JsonConverter.  (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(JsonReader, Type, Object, JsonSerializer)</a> )

		<a href="#">Type</a> , <a href="#">Object</a> , <a href="#">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 object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonWriter, Object, JsonSerializer)</a> .)

## ▀ Inheritance Hierarchy

[Object](#)  
 └ [JsonConverter](#)  
 └ **EntityKeyMemberConverter**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# CanConvert Method (*objectType*)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [EntityKeyMemberConverter](#) ► **CanConvert(Type)**

C# ▾

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.

# ReadJson Method (**reader**, **objectType**, **existingValue**, **serializer**)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►   
[EntityKeyMemberConverter](#) ► **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: 4.0.2.C  
(4.0.2.13623)

# WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [EntityKeyMemberConverter](#) ► **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: 4.0.2.0  
(4.0.2.13623)

# ExpandoObjectConverter Class

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► **ExpandoObjectConverter**

C# ▾

Converts an ExpandoObject to and from JSON.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public class ExpandoObjectConverter : JsonConverter
```

```
Public Class ExpandoObjectConverter _
    Inherits JsonConverter
```

```
public ref class ExpandoObjectConverter : public Json
```

## Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherit

Icon	Member	Description
	<a href="#">ExpandoObjectConverter()</a>	Initializes a new instance of the <b>ExpandoObjectConverter</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

		<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>(Overrides <a href="#">JsonConverter.CanWrite</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 JsonConverter.</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> <p>(Overrides</p>

		<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](#)

  └ [ExpandoObjectConverter](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# ExpandoObjectConverter Constructor

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

[ExpandoObjectConverter](#) ► **ExpandoObjectConverter()**

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public ExpandoObjectConverter()
```

```
Public Sub New
```

```
public:  
    ExpandoObjectConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# CanConvert Method (*objectType*)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►  
[ExpandoObjectConverter](#) ► **CanConvert(Type)**

C#

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.

# CanWrite Property

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

[ExpandoObjectConverter](#) ► **CanWrite**

C# ▾

Gets a value indicating whether this [JsonConverter](#) can write JSON.

## 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: 4.0.2.C  
(4.0.2.13623)

# ReadJson Method (**reader**, **objectType**, **existingValue**, **serializer**)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►  
[ExpandoObjectConverter](#) ► **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: 4.0.2.C  
(4.0.2.13623)

# WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►  
[ExpandoObjectConverter](#) ► **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: 4.0.2.0  
(4.0.2.13623)

# IsoDateTimeConverter

## 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 : DateTimeConverterBase
```

```
Public Class IsoDateTimeConverter
    Inherits DateTimeConverterBase
```

```
public ref class IsoDateTimeConverter : public DateTimeConverterBase
```

### Members

All Members

Constructors

Methods

Properties

Public

Instance

Declared

Protected

Static

Inherited

Icon

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>	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="#">Culture</a>	Gets or sets the culture used when converting a date to and from JSON.
	<a href="#">DateTimeFormat</a>	Gets or sets the date time format used when converting a date to and from JSON.
	<a href="#">DateTimeStyles</a>	Gets or sets the date time styles used when converting a date to and from 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 JsonConverter.  (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: 4.0.2.0  
(4.0.2.13623)



# IsoDateTimeConverter

## 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: 4.0.2.0  
(4.0.2.13623)

# Culture Property

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [IsoDateTimeConverter](#)

► **Culture**

C#

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: 4.0.2.0  
(4.0.2.13623)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [IsoDateTimeConverter](#)

► **DateTimeFormat**

C#

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: 4.0.2.0  
(4.0.2.13623)

# DateTimeStyles Property

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [IsoDateTimeConverter](#)

► **DateTimeStyles**

C#

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: 4.0.2.0  
(4.0.2.13623)

# ReadJson Method (*reader*, *objectType*, *existingValue*, *serializer*)

[Namespaces](#) ► [Newtonsoft.Json.Convertisers](#) ► [IsoDateTimeConverter](#)

► **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: 4.0.2.C  
(4.0.2.13623)

# WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [IsoDateTimeConverter](#)

► [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: 4.0.2.0  
(4.0.2.13623)

# Documentation **JavaScriptDateTimeConverter Class**

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

## JavaScriptDateTimeConverter

C# ▾

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 : public
```

### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declared  
 Inherited

Icon	Member	Description
	<a href="#">JavaScriptDateTimeConverter()</a>	Initializes a new instance of the <b>JavaScriptDateTimeConverter</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>	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 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 JsonConverter.  (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="#">Object</a> .)

		<a href="#">JsonConverter.ReadJson(JsonType, 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(JsonObject, JsonSerializer)</a> .)

## [-] Inheritance Hierarchy

```

Object
  ↘ JsonConverter
    ↘ DateTimeConverterBase
      ↘ JavaScriptDateTimeConverter

```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



# Documentation **JavaScriptDateTimeConverter**

## **Constructor**

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

[JavaScriptDateTimeConverter](#) ► **JavaScriptDateTimeConverter()**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# ReadJson Method (**reader**, **objectType**, **existingValue**, **serializer**)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [JavaScriptDateTimeConverter](#) ► **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 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: 4.0.2.C  
(4.0.2.13623)

# WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [JavaScriptDateTimeConverter](#) ► **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: 4.0.2.0  
(4.0.2.13623)

# Documentation **JsonDateTimeSerializationMode** Enumeration

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

## JsonDateTimeSerializationMode

C# ▾

Specifies whether a DateTime object represents a local time, a Coordinated Universal Time (UTC), or is not specified as either local time or UTC.

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public enum JsonDateTimeSerializationMode
```

```
Public Enumeration JsonDateTimeSerializationMode
```

```
public enum class JsonDateTimeSerializationMode
```

### Members

Member	Description
<b>Local</b>	The time represented is local time.
<b>Utc</b>	The time represented is UTC.
<b>Unspecified</b>	The time represented is not specified as either local time or Coordinated Universal Time (UTC).
<b>RoundtripKind</b>	Preserves the DateTimeKind field of a date when a DateTime object is converted to a string and the string is then converted back to a DateTime object.

# KeyValuePairConverter

## Class

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

### KeyValuePairConverter

C#

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"/> 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="#">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

		<p><a href="#">JsonConverter</a> can read JSON.</p> <p>(Inherited from <a href="#">JsonConverter</a>.)</p>
	<a href="#"><b>CanWrite</b></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="#"><b>Equals(Object)</b></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="#"><b>Finalize()</b></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="#"><b>GetHashCode()</b></a>	<p>Serves as a hash function for a particular type.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#"><b>GetSchema()</b></a>	<p>Gets the <a href="#">JsonSchema</a> of the JSON produced by the JsonConverter.</p> <p>(Inherited from <a href="#">JsonConverter</a>.)</p>
	<a href="#"><b>GetType()</b></a>	<p>Gets the <a href="#">Type</a> of the current instance.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#"><b>MemberwiseClone()</b></a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
	<a href="#"><b>ReadJson(JsonReader, Type, Object, JsonSerializer)</b></a>	<p>Reads the JSON representation of the object.</p> <p>(Overrides</p>

		<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: 4.0.2.0 (4.0.2.13623)



# KeyValuePairConverter

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [KeyValuePairConverter](#) ► **KeyValuePairConverter()**

C#

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

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public KeyValuePairConverter()
```

```
Public Sub New
```

```
public:  
KeyValuePairConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# CanConvert Method (*objectType*)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ►  
[KeyValuePairConverter](#) ► **CanConvert(*Type*)**

C#

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.

# ReadJson Method (**reader**, **objectType**, **existingValue**, **serializer**)

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

[KeyValuePairConverter](#) ► **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: 4.0.2.C  
(4.0.2.13623)

# WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [KeyValuePairConverter](#) ► **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: 4.0.2.0  
(4.0.2.13623)

C# ▾

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
<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="#">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 JsonConverter.  (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="#"><u>WriteJson(JsonWriter, Object, JsonSerializer)</u></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: 4.0.2.0  
(4.0.2.13623)

# RegexConverter

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [RegexConverter](#) ►

**RegexConverter()**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# CanConvert Method (*objectType*)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [RegexConverter](#) ► [CanConvert\(Type\)](#)

C# ▾

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.

# ReadJson Method (*reader*, *objectType*, *existingValue*, *serializer*)

[Namespaces](#) ► [Newtonsoft.Json.Converts](#) ► [RegexConverter](#) ►

**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: 4.0.2.C  
(4.0.2.13623)

# WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [RegexConverter](#) ► **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: 4.0.2.0  
(4.0.2.13623)

# StringEnumConverter

## Class

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► **StringEnumConverter**

C# ▾

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++

```
public class StringEnumConverter : JsonConverter
```

```
Public Class StringEnumConverter _
    Inherits JsonConverter
```

```
public ref class StringEnumConverter : public JsonCo
```

### - Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherit

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 JsonConverter.  (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="#"><u>ReadJson(JsonReader, Type, Object, JsonSerializer)</u></a>	Reads the JSON representation of the object.  (Overrides <a href="#">JsonConverter.ReadJson(JsonReader, Type, Object, JsonSerializer)</a> .)
≡	<a href="#"><u>ToString()</u></a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
≡	<a href="#"><u>WriteJson(JsonWriter, Object, JsonSerializer)</u></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: 4.0.2.C  
(4.0.2.13623)



# StringEnumConverter

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [StringEnumConverter](#)  
► **StringEnumConverter()**

C#

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

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public StringEnumConverter()
```

```
Public Sub New
```

```
public:  
StringEnumConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# CamelCaseText Property

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [StringEnumConverter](#)

► **CamelCaseText**

C# ▾

Gets or sets a value indicating whether the written enum text should be camel case.

## 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: 4.0.2.0  
(4.0.2.13623)

# CanConvert Method (*objectType*)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [StringEnumConverter](#)  
► **CanConvert(Type)**

C#

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.

# ReadJson Method (*reader*, *objectType*, *existingValue*, *serializer*)

[Namespaces](#) ► [Newtonsoft.Json.Convertisers](#) ► [StringEnumConverter](#)

► **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: 4.0.2.C  
(4.0.2.13623)

# WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [StringEnumConverter](#)  
 ► **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: 4.0.2.0  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# XXmlNodeConverter Class

[Namespaces](#) ► [Newtonsoft.Json.Convertisers](#) ► [XXmlNodeConverter](#)

C# ▾

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

Public  
 Protected

Instance  
 Static

Declare  
 Inherit

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 <b>XmlNodeConverter</b> 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 JsonConverter.  (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 the specified 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 arrays when converting the written XML to JSON.
≡	<a href="#">WriteJson(JsonWriter, Object, JsonSerializer)</a>	Writes the JSON representation of the specified object.  (Overrides <a href="#">JsonConverter.WriteJson(JsonWriter, Object, JsonSerializer)</a> .)

## [-] Inheritance Hierarchy

```

Object
└ JsonConverter
    └ XmlNodeConverter

```

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)





# XXmlNodeConverter

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [XmlNodeConverter](#) ► **XmlNodeConverter()**

C# ▾

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

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public XmlNodeConverter()
```

```
Public Sub New
```

```
public:  
XmlNodeConverter()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# CanConvert Method (valueType)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [XmlNodeConverter](#) ► [CanConvert\(Type\)](#)

C# ▾

Determines whether this instance can convert the specified value type.

## 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.

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

C# ▾

Gets or sets a value indicating whether to write the root JSON object.

## 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: 4.0.2.0  
(4.0.2.13623)

# ReadJson Method (*reader*, *objectType*, *existingValue*, *serializer*)

[Namespaces](#) ► [Newtonsoft.Json.Converts](#) ► [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: 4.0.2.C  
(4.0.2.13623)

# WriteArrayAttribute

## Property

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [XmlNodeConverter](#) ► [WriteArrayAttribute](#)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# WriteJson Method (writer, value, serializer)

[Namespaces](#) ► [Newtonsoft.Json.Converters](#) ► [XmlNodeConverter](#) ►

**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: 4.0.2.0  
(4.0.2.13623)

## 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++ ▾

```
namespace Newtonsoft.Json.Linq
```

```
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="#">IJEnumerable&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="#"><b>JContainer</b></a>	Represents a token that can contain other tokens.
◆	<a href="#"><b>JEnumerable&lt;T&gt;</b></a>	Represents a collection of <a href="#"><b>JToken</b></a> objects.
◆	<a href="#"><b>JObject</b></a>	Represents a JSON object.
◆	<a href="#"><b>JProperty</b></a>	Represents a JSON property.
◆	<a href="#"><b>JRaw</b></a>	Represents a raw JSON string.
◆	<a href="#"><b>JToken</b></a>	Represents an abstract JSON token
◆	<a href="#"><b>JTokenEqualityComparer</b></a>	Compares tokens to determine whether they are equal.
◆	<a href="#"><b>JTokenReader</b></a>	Represents a reader that provides fast, non-cached, forward-only access to serialized Json data.
◆	<a href="#"><b>JTokenType</b></a>	Specifies the type of token.
◆	<a href="#"><b>JTokenWriter</b></a>	Represents a writer that provides a fast, non-cached, forward-only way of generating Json data.

	<a href="#"><b>JValue</b></a>	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

 Public Instance Declared Protected Static Inherited

Icon

Member

Description

[Ancestors<T>\(IEnumerable<T>\)](#)

Returns a collection of tokens that contains the ancestors of every token in the source collection.

[AsJEnumerable\(IEnumerable<JToken>\)](#)

Returns the input typed as [IJEnumerable<T>](#).

[AsJEnumerable<T>\(IEnumerable<T>\)](#)

Returns the input typed as [IJEnumerable<T>](#).

= 	<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.
= 	<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.
= 	<a href="#">Properties(IEnumerable&lt;JObject&gt;)</a>	Returns a collection of child properties of every object in the source collection.
= 	<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.
= 	<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: 4.0.2.0 (4.0.2.13623)

# Ancestors<T> Method (source)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ► **Ancestors<T>(IEnumerable<T>)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

▾**▀ Members**

Icon	Member	Description
	<a href="#">AsJEnumerable(IEnumerable&lt;JToken&gt;)</a>	Returns the input typed as <a href="#">IJEnumerable&lt;T&gt;</a> .
	<a href="#">AsJEnumerable&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns the input typed as <a href="#">IJEnumerable&lt;T&gt;</a> .

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# AsJEnumerable Method (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: 4.0.2.C  
(4.0.2.13623)

# AsJEnumerable<T>

## Method (source)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ►

`AsJEnumerable<T>(IEnumerable<T>)`

C# ▾

Returns the input typed as [IJEnumerable<T>](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public static IJEnumerable<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 IJEnumerable(Of T)
```

```
[ExtensionAttribute]
public:
generic<typename T>
where T : JToken
static IJEnumerable<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**

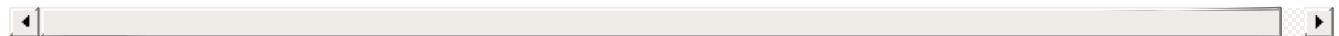
The input typed as [IJEnumerable<T>](#).

## **❑ 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: 4.0.2.C  
(4.0.2.13623)



# Children Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ► [Children\(\)](#)

C# ▾

## ‑ Members

Icon	Member	Description
= 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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# Children<T> Method (source)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ► **Children<T>(IEnumerable<T>)**

C# ▾

Returns a collection of child tokens of every array in the source collection.

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static IJEnumerable<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 IJEnumerable(Of JToken)
```

```
[ExtensionAttribute]
public:
generic<typename T>
where T : JToken
static IJEnumerable<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: 4.0.2.C  
(4.0.2.13623)

# Children<T, U> Method (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: 4.0.2.C  
(4.0.2.13623)

# Descendants<T> Method (source)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ►   
**Descendants<T>(IEnumerable<T>)**

C# ▾

Returns a collection of tokens that contains the descendants of every token in the source collection.

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static IJEnumerable<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 IJEnumerable(Of JToken)
```

```
[ExtensionAttribute]
public:
generic<typename T>
where T : JContainer
static IJEnumerable<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

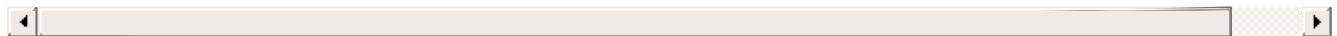
An [IEnumerable<T>](#) of [JToken](#) that contains the descendants of every node in the source collection.

## ▀ 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: 4.0.2.C  
(4.0.2.13623)



# Properties Method

## (source)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ►

**Properties(IEnumerable< JObject >)**

C# ▾

Returns a collection of child properties of every object in the source collection.

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static IJEnumerable<JProperty> Properties(
    this IEnumerable< JObject > source
)
```

```
<ExtensionAttribute> _
Public Shared Function Properties ( _
    source As IEnumerable(Of JObject) ) _
As IJEnumerable(Of JProperty)
```

```
[ExtensionAttribute]
public:
static IJEnumerable<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: 4.0.2.C  
(4.0.2.13623)

C# ▾

## - Members

Icon	Member	Description
= 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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Value<U> Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ► **Value<U>**  
**(IEnumerable<JToken>)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# Value<*T*, *U*> Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ► **Value<*T*, *U*>(IEnumerable<*T*>)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

C# ▾

## 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.

C#

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: 4.0.2.C  
(4.0.2.13623)

# Values<U> Method

## (source)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ► **Values<U>**  
**(IEnumerable<JToken>)**

C#

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: 4.0.2.C  
(4.0.2.13623)

# Values<U> Method

## (source, key)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ► [Values<U>](#)  
 (IEnumerable<JToken>, Object)

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# Values Method (source, key)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [Extensions](#) ►

**Values(IEnumerable<JToken>, Object)**

C# ▾

Returns a collection of child values of every object in the source collection with the given key.

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static IJEnumerable<JToken> Values(
    this IEnumerable<JToken> source,
    object key
)
```

```
<ExtensionAttribute> _
Public Shared Function Values ( _
    source As IEnumerable(Of JToken), _
    key As Object _
) As IJEnumerable(Of JToken)
```

```
[ExtensionAttribute]
public:
static IJEnumerable<JToken>^ Values(
    IEnumerable<JToken>^ source,
    Object^ key
)
```

## 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: 4.0.2.C  
(4.0.2.13623)

# IJEnumerable<T>

## Interface

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► **IJEnumerable<T>**

C# ▾

Represents a collection of [JToken](#) objects.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public interface IJEnumerable<out T> : IEnumerable<T>
    IEnumerable
where T : JToken
```

```
Public Interface IJEnumerable(Of Out T As JToken) _
    Inherits IEnumerable\(Of T\), IEnumerable
```

```
generic<typename T>
where T : JToken
public interface class IJEnumerable : IEnumerable<T>
    IEnumerable
```

### Generic Template Parameters

*T*

The type of token

### Members

All Members

Methods

Properties

Public

Protected

Instance

Static

Declared

Inherited

Icon

Member

Description



[Ancestors<T>\(IEnumerable<T>\)](#)

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="#"><u>Value&lt;T, U&gt;(IEnumerable&lt;T&gt;)</u></a>	Converts the value.  (Inherited from <a href="#"><u>Extensions</u></a> .)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# Item Property (key)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [IJEnumerable<T>](#) ► **Item[Object]**

C# ▾

Gets the [IJEnumerable<T>](#) with the specified key.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
IJEnumerable<JToken> this[
    Object key
] { get; }
```

```
ReadOnly Default Property Item ( _
    key As Object
) As IJEnumerable(Of JToken)
    Get
```

```
property IJEnumerable<JToken^>^ default[Object^ key]
    IJEnumerable<JToken^>^ get (Object^ key);
}
```

## Parameters

**key (Object)**

## Value

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

Represents a JSON array.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JArray : JContainer, IList<JToken>,  

    ICollection<JToken>, IEnumerable<JToken>, IE
```

```
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^>, IEnumerable
```

## Members

All Members

Constructors

Methods

Properties

Events

 Public Instance Declared Protected Static Inherited

Icon

Member

Description

[JArray\(\)](#)

Initializes a new instance of the JArray class.

[JArray\(JArray\)](#)

Initializes a new instance of the JArray class from another JArray object.

	<a href="#"><u>JArray(Object[])</u></a>	Initializes the array with the specified elements.
≡	<a href="#"><u>JArray(Object)</u></a>	Initializes the array with the specified elements.
≡	<a href="#"><u>Add(JToken)</u></a>	Adds the specified value to the collection.
≡	<a href="#"><u>Add(Object)</u></a>	Adds the specified value as an object to the collection.
≡	<a href="#"><u>AddAfterSelf(Object)</u></a>	Adds the specified value immediately after the current token.
≡	<a href="#"><u>AddBeforeSelf(Object)</u></a>	Adds the specified value immediately before the current token.
≡	<a href="#"><u>AddFirst(Object)</u></a>	Adds the specified value as the first item in the collection.
⚡	<a href="#"><u>AddingNew</u></a>	Occurs when a new item is added to the collection.

		addes to the current token stream.
		(Inherited from <a href="#">JToken</a> )
≡	<a href="#">AfterSelf()</a>	Returns the sibling tokens after the current token.
		(Inherited from <a href="#">JToken</a> )
≡	<a href="#">Ancestors()</a>	Returns an enumerable of all ancestor tokens.
		(Inherited from <a href="#">JToken</a> )
≡	<a href="#">Ancestors&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns an enumerable of all ancestor tokens.
		(Inherited from <a href="#">ExtensibleObject</a> )
≡	<a href="#">AsJEnumerable(IEnumerable&lt;JToken&gt;)</a>	Returns an enumerable of the current token.
		(Inherited from <a href="#">ExtensibleObject</a> )
≡	<a href="#">AsJEnumerable&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns an enumerable of the current token.
		(Inherited from <a href="#">ExtensibleObject</a> )
≡	<a href="#">BeforeSelf()</a>	Returns the sibling tokens before the current token.
		(Inherited from <a href="#">JToken</a> )

	<a href="#">Children()</a>	Retu child in do
≡	<a href="#">Children&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	(Inhe JCor
≡	<a href="#">Children&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Retu child in the (Inhe Extende
≡	<a href="#">Clear()</a>	Retu conv every colla (Inhe Extende
⚡	<a href="#">CollectionChanged</a>	Rem IColle
⚡	<a href="#">Contains(JToken)</a>	Occu of the chan is res (Inhe JCor
⊕	<a href="#">Count</a>	Dete IColle spec

	<a href="#">IColl</a>
≡	<a href="#">CreateReader()</a>
	Create this t  (Inhe
≡	<a href="#">CreateWriter()</a>
	Create can I to the  (Inhe <a href="#">JCor</a>
≡	<a href="#">DeepClone()</a>
	Create the <a href="#">J</a> are r  (Inhe
≡	<a href="#">Descendants()</a>
	Retu desc token  (Inhe <a href="#">JCor</a>
≡	<a href="#">Equals(Object)</a>
	Dete spec the c  (Inhe
💡	<a href="#">Finalize()</a>
	Allow atten and p oper <a href="#">Obje</a> garb  (Inhe

	<a href="#">First</a>	Get the first child of this token.
	<a href="#">FromObject(Object)</a>	Create an object from the specified JSON object.
	<a href="#">FromObject(Object, JsonSerializer)</a>	Create an object from the specified JSON object using the specified JsonSerializer.
	<a href="#">GetHashCode()</a>	Serves as the default hash function.
	<a href="#">GetMetaObject(Expression)</a>	Returns the dynamic metaobject corresponding to the specified expression.
	<a href="#">GetType()</a>	Gets the current type.
	<a href="#">HasValues</a>	Gets whether this token has child tokens.
	<a href="#">IndexOf(JToken)</a>	Determines the index of the specified token.

	<a href="#">IList&lt; T &gt;</a>
	<a href="#">Insert(Int32, JToken)</a>
	Inserts the specified value at the specified index.
	<a href="#">IsValid(JToken, JsonSchema)</a>
	Determines whether the specified JSON token is valid according to the specified schema.
	<a href="#">Item[Object]</a>
	Gets the item at the specified index.
	<a href="#">Item[Int32]</a>
	Gets the item at the specified index.
	<a href="#">Last</a>
	Gets the last item.
	<a href="#">ListChanged</a>
	Occurs when the list changes.
	<a href="#">Load(JsonReader)</a>
	Loads the list from the specified JSON reader.

	<a href="#">MemberwiseClone()</a>	Create the c  (Inhe
	<a href="#">Next</a>	Gets of thi  (Inhe
	<a href="#">OnAddingNew(AddingNewEventArgs)</a>	Rais even  (Inhe JCor
	<a href="#">OnCollectionChanged(NotifyCollectionChangedEventArgs)</a>	Rais Colle  (Inhe JCor
	<a href="#">OnListChanged(ListChangedEventArgs)</a>	Rais even  (Inhe JCor
	<a href="#">Parent</a>	Gets  (Inhe
	<a href="#">Parse(String)</a>	Load string  
	<a href="#">Previous</a>	Gets toke  (Inhe
		

	<a href="#">Remove(JToken)</a>	Removes the specified object from the collection.
≡	<a href="#">Remove()</a>	Removes the current node from its parent.
≡	<a href="#">RemoveAll()</a>	Removes all nodes from the collection.
≡	<a href="#">RemoveAt(Int32)</a>	Removes the node at the specified index.
≡	<a href="#">Replace(JToken)</a>	Replaces the current node with the specified node.
≡	<a href="#">ReplaceAll(Object)</a>	Replaces all nodes in the collection with the specified node.
网站地图	<a href="#">Root</a>	Gets the root node of the document.
≡	<a href="#">SelectToken(String)</a>	Selects the first matching node.

		(Inhe
≡	<a href="#">SelectToken(String, Boolean)</a>	Select match
		(Inhe
≡	<a href="#">ToString()</a>	Retu JSON
		(Inhe
≡	<a href="#">ToString(Formatting, JsonConverter[])</a>	Retu token form
		(Inhe
	<a href="#">Type</a>	Gets <a href="#">JTok</a>
		(Over
≡	<a href="#">Validate(JToken, JsonSchema)</a>	Valid <a href="#">JTok</a>
		(Inhe Exte
≡	<a href="#">Validate(JToken, JsonSchema, ValidationEventHandler)</a>	Valid <a href="#">JTok</a>
		(Inhe Exte
≡	<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>	Extends Converts (Inherited from ExtensibleObject)
≡	<a href="#">Values&lt;T&gt;()</a>	Returns child objects in document order. (Inherited from JToken)
≡	<a href="#">Values(IEnumerable&lt;JToken&gt;, Object)</a>	Returns child objects in the order specified with the key. (Inherited from ExtensibleObject)
≡	<a href="#">Values(IEnumerable&lt;JToken&gt;)</a>	Returns child objects in the order specified. (Inherited from ExtensibleObject)
≡	<a href="#">Values&lt;U&gt;(IEnumerable&lt;JToken&gt;, Object)</a>	Returns converted enumerable collection key. (Inherited from ExtensibleObject)
≡	<a href="#">WriteTo(JsonWriter, JsonConverter[])</a>	Writes JSON. (Overrides JToken.WriteTo(JsonWriter, JsonConverter[]))

## Inheritance Hierarchy

[Object](#)

└ [JToken](#)

  └ [JContainer](#)

    └ [JArray](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



## - Members

Icon	Member	Description
	<a href="#">JArray()</a>	Initializes a new instance of the <a href="#">JArra</a> class.
	<a href="#">JArray(JArray)</a>	Initializes a new instance of the <a href="#">JArra</a> class from another <a href="#">JArray</a> object.
	<a href="#">JArray(Object[])</a>	Initializes a new instance of the <a href="#">JArra</a> class with the specified content.
	<a href="#">JArray(Object)</a>	Initializes a new instance of the <a href="#">JArra</a> class with the specified content.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

# JArray Constructor (other)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► [JArray\(JArray\)](#)

C#

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: 4.0.2.C  
(4.0.2.13623)

# JArray Constructor (content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► [JArray\(Object\)](#)

C#

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: 4.0.2.C  
(4.0.2.13623)

# JArray Constructor (content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► [JArray\(Object\[\]\)](#)

C#

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: 4.0.2.C  
(4.0.2.13623)

# 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.

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: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)

# Count Property

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **Count**

C# ▾

Gets the number of elements contained in the [ICollection<T>](#).

## 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

## Return Value

The number of elements contained in the [ICollection<T>](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# FromObject Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **FromObject()**

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: 4.0.2.0  
(4.0.2.13623)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# 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: 4.0.2.C  
(4.0.2.13623)

Determines the index of a specific item in the [IList<T>](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

---

```
public int IndexOf(  
    JToken item  
)
```

```
Public Function IndexOf ( _  
    item As JToken _  
) As Integer
```

```
public:  
virtual int IndexOf(  
    JToken^ item  
) sealed
```

## Parameters

### *item* ([JToken](#))

The object to locate in the [IList<T>](#).

## Return Value

The index of *item* if found in the list; otherwise, -1.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# Insert Method (index, item)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **Insert(Int32, JToken)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

## [-] 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: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.0 (4.0.2.13623)

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) override;
```

## Parameters

[key \(Object\)](#)

## Value

The [JToken](#) with the specified key.

# Load Method (reader)

[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: 4.0.2.0 (4.0.2.13623)

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# Remove Method (item)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **Remove(JToken)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# RemoveAt Method

## (index)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ► **RemoveAt(Int32)**

C# ▾

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>	<code>index</code> is not a valid index in the <code>IList&lt;T&gt;</code> .
<a href="#">NotSupportedException</a>	The <code>IList&lt;T&gt;</code> is read-only.

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: 4.0.2.C  
(4.0.2.13623)

# WriteTo Method (writer, converters)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JArray](#) ►

**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.

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

 Public Instance Declared Protected Static Inherited

Icon

Member

Description

[JConstructor\(\)](#)Initializes a new instance of the **JConstructor** class.[JConstructor\(JConstructor\)](#)Initializes a new instance of the **JConstructor** class from another **JConstructor** object.[JConstructor\(String, Object\[\]\)](#)Initializes a new instance of the **JConstructor** class with the specified string and array of objects.

	 <a href="#">JConstructor(String, Object)</a>	Initializes the object with the specified name and value.
	 <a href="#">JConstructor(String)</a>	Initializes the object with the specified name.
	 <a href="#">Add(Object)</a>	Adds the specified object as the child of the current token.
	 <a href="#">AddAfterSelf(Object)</a>	Adds the specified object immediately after the current token.
	 <a href="#">AddBeforeSelf(Object)</a>	Adds the specified object immediately before the current token.
	 <a href="#">AddFirst(Object)</a>	Adds the specified object as the first child of the current token.
	<a href="#">AddingNew</a>	Occurs when a new object is added.

		(Inhe JCor
≡	<a href="#">AfterSelf()</a>	Retu siblir tokei
		(Inhe
≡	<a href="#">Ancestors()</a>	Retu ance tokei
		(Inhe
≡	<a href="#">BeforeSelf()</a>	Retu siblir tokei
		(Inhe
≡	<a href="#">Children()</a>	Retu child in do
		(Inhe JCor
⚡	<a href="#">CollectionChanged</a>	Occu of the chan is res
		(Inhe JCor
≡	<a href="#">CreateReader()</a>	Cre this t
		(Inhe
≡	<a href="#">CreateWriter()</a>	Cre can I

		to the
		(Inhe JCor
≡	<a href="#">DeepClone()</a>	Create the <a href="#">J</a> are r
		(Inhe
≡	<a href="#">Descendants()</a>	Retu desc toker
		(Inhe JCor
≡	<a href="#">Equals(Object)</a>	Dete spec the c
		(Inhe
💡	<a href="#">Finalize()</a>	Allow atten and p oper <a href="#">Obj</a> garb
		(Inhe
📄	<a href="#">First</a>	Get t this t
		(Inhe JCor
≡	<a href="#">GetHashCode()</a>	Serv for a
		(Inhe

	<a href="#">GetMetaObject(Expression)</a>	Returns the meta object for the current object. Dynamic response operator this context (Inherited from <a href="#">Object</a> )
	<a href="#">GetType()</a>	Gets the runtime type of the current object. (Inherited from <a href="#">Object</a> )
	<a href="#">HasValues</a>	Gets whether the current object has child objects. (Inherited from <a href="#">JCorObject</a> )
	<a href="#">IsValid(JToken, JsonSchema)</a>	Determines whether the current object is valid according to the specified schema. Determines if the current token is valid. (Inherited from <a href="#">ExtensibleObject</a> )
	<a href="#">Item[Object]</a>	Gets the specified item. (Overrides <a href="#">JToken.Item[Object]</a> )
	<a href="#">Last</a>	Gets the last item. This token is the last item in the list. (Inherited from <a href="#">JCorObject</a> )
	<a href="#">ListChanged</a>	Occurs when the list changes.

		(Inhe JCor
	<a href="#">Load(JsonReader)</a>	Load from
	<a href="#">MemberwiseClone()</a>	Create 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 JCor
	<a href="#">OnCollectionChanged(NotifyCollectionChangedEventArgs)</a>	Rais Colle (Inhe JCor
	<a href="#">OnListChanged(ListChangedEventArgs)</a>	Rais even (Inhe JCor
	<a href="#">Parent</a>	Gets (Inhe

	<a href="#">Previous</a>	Gets the previous token in the stream. (Inherited from <a href="#">JToken</a> )
	<a href="#">Remove()</a>	Removes its parent node. (Inherited from <a href="#">JToken</a> )
	<a href="#">RemoveAll()</a>	Removes all child nodes from this node. (Inherited from <a href="#">JToken</a> )
	<a href="#">Replace(JToken)</a>	Replaces the selected node with the specified node. (Inherited from <a href="#">JToken</a> )
	<a href="#">ReplaceAll(Object)</a>	Replaces all child nodes of this node with the specified object. (Inherited from <a href="#">JToken</a> )
	<a href="#">Root</a>	Gets this node's root node. (Inherited from <a href="#">JToken</a> )
	<a href="#">SelectToken(String)</a>	Selects the first matching node. (Inherited from <a href="#">JToken</a> )
	<a href="#">SelectToken(String, Boolean)</a>	Selects the first matching node. (Inherited from <a href="#">JToken</a> )
	<a href="#">ToString()</a>	Returns a string representation of this node.

	<a href="#">ToString(Formatting, JsonConverter[])</a>	JSON (Inherited from <a href="#">Object</a> )
≡	<a href="#">Type</a>	Returns the type information for the current <a href="#">JsonToken</a> .  (Overridden from <a href="#">Object</a> )
≡	<a href="#">Validate(JsonSchema)</a>	Validates the current <a href="#">JsonToken</a> against the specified <a href="#">JsonSchema</a> .  (Inherited from <a href="#">Object</a> )
≡	<a href="#">Validate(JsonSchema, ValidationEventHandler)</a>	Validates the current <a href="#">JsonToken</a> against the specified <a href="#">JsonSchema</a> , using the specified <a href="#">ValidationEventHandler</a> .  (Inherited from <a href="#">Object</a> )
≡	<a href="#">Value&lt;T&gt;(Object)</a>	Gets the value of the current <a href="#">JsonToken</a> as the specified type.  (Inherited from <a href="#">Object</a> )
≡	<a href="#">Values&lt;T&gt;()</a>	Returns an <a href="#">Object</a> containing the child tokens in document order.  (Inherited from <a href="#">Object</a> )
≡	<a href="#">WriteTo(JsonWriter, JsonConverter[])</a>	Writes the current <a href="#">JsonToken</a> to the specified <a href="#">JsonWriter</a> using the specified <a href="#">JsonConverter</a> s.

(Ove  
JTok  
Json

## **[-] Inheritance Hierarchy**

[Object](#)

└ [JToken](#)

  └ [JContainer](#)

    └ **JConstructor**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# JConstructor

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ► [JConstructor\(\)](#)

C# ▾

### 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.

# JConstructor Constructor

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ► [JConstructor\(\)](#)

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# JConstructor

## Constructor (other)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ► [JConstructor\(JConstructor\)](#)

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

## Constructor (name)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ►

**JConstructor(String)**

C#

Initializes a new instance of the [JConstructor](#) class with the specified name.

### 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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JConstructor

## Constructor (name, content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ► [JConstructor\(String, Object\)](#)

C# ▾

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,
    object content
)
```

```
Public Sub New ( _
    name As String, _
    content As Object _
)
```

```
public:
JConstructor(
    String^ name,
    Object^ content
)
```

### Parameters

#### **name ([String](#))**

The constructor name.

#### **content ([Object](#))**

The contents of the constructor.

# JConstructor

## Constructor (name, content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ►

**JConstructor(String, Object[])**

C#

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.

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) override;
```

## Parameters

[key \(Object\)](#)

## Value

The [JToken](#) with the specified key.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

▾

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](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.0 (4.0.2.13623)

# WriteTo Method (writer, converters)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JConstructor](#) ► [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.

Represents a token that can contain other tokens.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract class JContainer : JToken,  

    IList<JToken>, ICollection<JToken>, IEnumerable  

ITypedList, IBindingList, IList, ICollection  

INotifyCollectionChanged
```

```
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^>, IEnumerable  

ITypedList, IBindingList, IList, ICollection  

INotifyCollectionChanged
```

## Members

All Members

Methods

Properties

Events

 Public Instance

Declared

 Protected Static

Inherited

Icon

Member

Desc

[Add\(Object\)](#)Adds  
content  
to this

	 <a href="#">AddAfterSelf(Object)</a>	Adds content after the current token.  (Inherited from <a href="#">JTok</a> )
	 <a href="#">AddBeforeSelf(Object)</a>	Adds content before the current token.  (Inherited from <a href="#">JTok</a> )
	 <a href="#">AddFirst(Object)</a>	Adds content as the first child of the current token.  (Inherited from <a href="#">JTok</a> )
	<a href="#">AddingNew</a>	Occurs when adding new items to a collection.
	 <a href="#">AfterSelf()</a>	Returns the content of the node immediately after the current document node.  (Inherited from <a href="#">JTok</a> )
	 <a href="#">Ancestors()</a>	Returns the ancestors of the current token.  (Inherited from <a href="#">JTok</a> )

≡	<a href="#">Ancestors&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns an <a href="#">IEnumerable&lt;T&gt;</a> containing the ancestors of the current token. This token is included in the returned collection.
≡	<a href="#">AsJEnumerable(IEnumerable&lt;JToken&gt;)</a>	Returns the current token converted to the specified type. This token is included in the returned collection.
≡	<a href="#">AsJEnumerable&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns the current token converted to the specified type. This token is included in the returned collection.
≡	<a href="#">BeforeSelf()</a>	Returns the previous sibling of the current token, before the current document starts.
≡	<a href="#">Children()</a>	Returns the children of the current token. This token is included in the returned collection.
≡	<a href="#">Children&lt;T&gt;(IEnumerable&lt;T&gt;)</a>	Returns the children of the current token.

		of ch every sourc
		(Inhe Extende
✳	<a href="#">Children&lt;T, U&gt;(IEnumerable&lt;T&gt;)</a>	Retu of co toker in the colle
		(Inhe Extende
⚡	<a href="#">CollectionChanged</a>	Occu items colle chan colle
✳	<a href="#">CreateReader()</a>	Crea <u>Json</u> tokene
		(Inhe <u>JTok</u>
✳	<a href="#">CreateWriter()</a>	Crea <u>Json</u> be us tokene
✳	<a href="#">DeepClone()</a>	Crea insta <u>JTok</u> tokene

		recu
		(Inhe JTok
≡	<a href="#">Descendants()</a>	Retu of the token in do
≡	<a href="#">Equals(Object)</a>	Dete the s is eq curre
		(Inhe Obje
💡	<a href="#">Finalize()</a>	Allow atten reso perf clear befor recla garb
		(Inhe Obje
📁	<a href="#">First</a>	Get t toke
		(Ove JTok
≡	<a href="#">GetHashCode()</a>	Serv funct partic

		(Inhe Obje
	<a href="#">GetMetaObject(Expression)</a>	Returns a <a href="#">DynamicObject</a> representing the current binding context.
		(Inhe JTok
	<a href="#">GetType()</a>	Gets the current type.
		(Inhe Obje
	<a href="#">HasValues</a>	Gets a value indicating whether this token has child tokens.
		(Ove JTok
	<a href="#">IsValid(JToken, JsonSchema)</a>	Determines if the <a href="#">JToken</a> is valid according to the specified <a href="#">JsonSchema</a> .
		(Inhe Exte
	<a href="#">Item[Object]</a>	Gets the single item in the sequence.
		(Inhe JTok
	<a href="#">Last</a>	Gets the last token in the sequence.
		(Ove

		<a href="#">JTok</a>
	<a href="#">ListChanged</a>	Occurs when changes occur in the collection.
	<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 token for the next item.
		(Inherited from <a href="#">JTok</a> )
	<a href="#">OnAddingNew(AddingNewEventArgs)</a>	Raises the <a href="#">AddingNew</a> event.
		(Inherited from <a href="#">CollectionBase</a> )
	<a href="#">OnCollectionChanged(NotifyCollectionChangedEventArgs)</a>	Raises the <a href="#">CollectionChanged</a> event.
		(Inherited from <a href="#">INotifyCollectionChanged</a> )
	<a href="#">OnListChanged(ListChangedEventArgs)</a>	Raises the <a href="#">ListChanged</a> event.
		(Inherited from <a href="#">JTok</a> )
	<a href="#">Parent</a>	Gets the parent item.
		(Inherited from <a href="#">JTok</a> )
	<a href="#">Previous</a>	Gets the token for the previous item.
		(Inherited from <a href="#">JTok</a> )

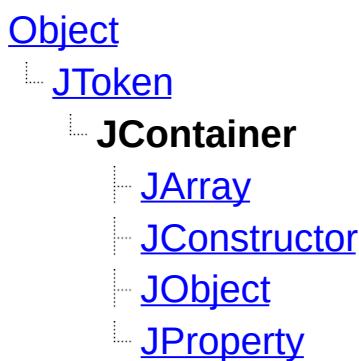
		sibling node
		(Inhe <a href="#">JTok</a>
≡	<a href="#">Remove()</a>	Rem from
		(Inhe <a href="#">JTok</a>
≡	<a href="#">RemoveAll()</a>	Rem node toke
≡	<a href="#">Replace(JToken)</a>	Repl with token
		(Inhe <a href="#">JTok</a>
≡	<a href="#">ReplaceAll(Object)</a>	Repl child token spec
📁	<a href="#">Root</a>	Gets of thi
		(Inhe <a href="#">JTok</a>
≡	<a href="#">SelectToken(String)</a>	Select that object
		(Inhe

		<a href="#">JTok</a>
≡	<a href="#">SelectToken(String, Boolean)</a>	Selects that object  (Inhe <a href="#">JTok</a>
≡	<a href="#">ToString()</a>	Retu JSO  (Inhe <a href="#">JTok</a>
≡	<a href="#">ToString(Formatting, JsonConverter[])</a>	Retu for th the g and c  (Inhe <a href="#">JTok</a>
	<a href="#">Type</a>	Gets for th  (Inhe <a href="#">JTok</a>
≡	<a href="#">Validate(JToken, JsonSchema)</a>	Valid spec  (Inhe <a href="#">Exte</a>
≡	<a href="#">Validate(JToken, JsonSchema, ValidationEventHandler)</a>	Valid spec  (Inhe <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 the this t docu (Ove <a href="#">JTok</a>
≡	<a href="#">Values(IEnumerable&lt;JToken&gt;, Object)</a>	Retu of ch ever sourc the g (Inhe <a href="#">Exte</a>
≡	<a href="#">Values(IEnumerable&lt;JToken&gt;)</a>	Retu of ch ever sourc (Inhe <a href="#">Exte</a>
≡	<a href="#">Values&lt;U&gt;(IEnumerable&lt;JToken&gt;, Object)</a>	Retu of co

		value
		object
		collection
		given
		(Inherited from <a href="#">Object</a> )
•	<a href="#">WriteTo(JsonWriter, JsonConverter[])</a>	Writes the object to the specified writer using the specified converter.
		(Inherited from <a href="#">JsonContainerToken</a> )

## ▀ Inheritance Hierarchy



---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



Adds the specified content as children of this [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public void Add(  
    object content  
)
```

```
Public Sub Add ( _  
    content As Object _  
)
```

```
public:  
void Add(  
    object^ content  
)
```

## Parameters

### **content (Object)**

The content to be added.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# AddFirst Method (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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

C# ▾

Occurs before an item is added to the collection.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public event AddingNewEventHandler AddingNew
```

```
Public Event AddingNew As AddingNewEventHandler
```

```
public:  
    event AddingNewEventHandler^ AddingNew {  
        void add (AddingNewEventHandler^ value);  
        void remove (AddingNewEventHandler^ value);  
    }
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

C# ▾

Returns a collection of the child tokens of this token, in document order.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public override JEnumerable<JToken> Children()
```

```
Public Overrides Function Children As JEnumerable(Of
```

```
public:  
virtual JEnumerable<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: 4.0.2.C  
(4.0.2.13623)



[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ► [CollectionChanged](#)

C# ▾

Occurs when the items list of the collection has changed, or the collection is reset.

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

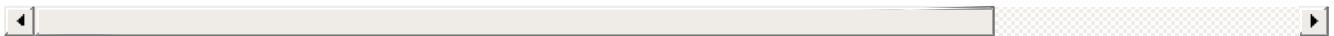
```
public event NotifyCollectionChangedEventHandler Col
```

```
Public Event CollectionChanged As NotifyCollectionch
```

```
public:  
virtual event NotifyCollectionChangedEventHandler^  
    void add (NotifyCollectionChangedEventHandler^  
    void remove (NotifyCollectionChangedEventHandler^  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



C# ▾

Creates an [JsonWriter](#) that can be used to add tokens to the [JToken](#).

## 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: 4.0.2.C  
(4.0.2.13623)

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

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: 4.0.2.0 (4.0.2.13623)

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 Boolean  
    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: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.0 (4.0.2.13623)

# ListChanged Event

[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 ListChangedEventArgs ListChanged
```

```
Public Event ListChanged As ListChangedEventArgs
```

```
public:  
virtual event ListChangedEventArgs^ ListChanged  
    void add (ListChangedEventArgs^ value);  
    void remove (ListChangedEventArgs^ value)  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ►

## OnAddingNew(AddingEventArgs)

C# ▾

Raises the [AddingNew](#) event.

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
protected virtual void OnAddingNew(  
    AddingEventArgs e  
)
```

```
Protected Overridable Sub OnAddingNew ( _  
    e As AddingEventArgs _  
)
```

```
protected:  
virtual void OnAddingNew(  
    AddingEventArgs^ e  
)
```

### Parameters

#### e ([AddingEventArgs](#))

The [AddingEventArgs](#) instance containing the event data.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# OnCollectionChanged

## Method (e)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ►

**OnCollectionChanged(NotifyCollectionChangedEventArgs)**

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# OnListChanged Method

(e)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ►

**OnListChanged(ListChangedEventArgs)**

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



Removes the child nodes from this token.

## Declaration Syntax

```
public void RemoveAll()
```

```
Public Sub RemoveAll
```

```
public:  
void RemoveAll()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# ReplaceAll Method (content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JContainer](#) ► **ReplaceAll(Object)**

C#

Replaces the children nodes of this token with the specified content.

## 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: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.0 (4.0.2.13623)



## Structure

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► **JEnumerable<T>**

C# ▾

Represents a collection of [JToken](#) objects.

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public struct JEnumerable<T> : IJEnumerable<T>,
    IEnumerable<T>, IEnumerable
where T : JToken
```

```
Public Structure JEnumerable(Of T As JToken) _
    Implements IJEnumerable(Of T), IEnumerable(O
IEnumerable
```

```
generic<typename T>
where T : JToken
public value class JEnumerable : IJEnumerable<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 of 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 href="#">IJEnumerable&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> )
 <b>S</b>	<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

		<p><a href="#">Object</a> is reclaimed by garbage collection.</p> <p>(Inherited from <a href="#">Object</a>.)</p>
≡	<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 <a href="#">Extension</a> )

[◀]

[▶]

## Constructor (enumerable)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JEnumerable<T>](#) ► **JEnumerable<T>(IEnumerable<T>)**

C#

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

# Equals Method (obj)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JEnumerable<T>](#) ► [Equals\(Object\)](#)

C#

Determines whether the specified [Object](#) is equal to this instance.

## 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 this instance.

## Return Value

true if the specified [Object](#) is equal to this instance; otherwise, false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

[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: 4.0.2.C  
(4.0.2.13623)

# GetHashCode Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JEnumerable<T>](#) ► [GetHashCode\(\)](#)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# Item Property (key)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JEnumerable<T>](#) ► **Item[Object]**

C# ▾

Gets the [IJEnumerable<T>](#) with the specified key.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public IJEnumerable<JToken> this[  
    Object key  
] { get; }
```

```
Public ReadOnly Default Property Item ( _  
    key As Object _  
) As IJEnumerable(Of JToken)  
    Get
```

```
public:  
virtual property IJEnumerable<JToken^>^ default[Object]  
    IJEnumerable<JToken^>^ get (Object^ key) sea  
}
```

## Parameters

**key (Object)**

## Value

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



Represents a JSON object.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JObject : JContainer, IDictionary<string, JToken>, ICollection<KeyValuePair<string, JToken>>, IEnumerable, INotifyPropertyChanged, ICustomTypeDescriptor
```

```
Public Class JObject
    Inherits JContainer
    Implements IDictionary(Of String, JToken),
    ICollection(Of KeyValuePair(Of String, JToken)),
    IEnumerable, INotifyPropertyChanged, ICustomTypeDescriptor
```

```
public ref class JObject : public JContainer,
    IDictionary<String^, JToken^>, ICollection<KeyValuePair<String^, JToken^>>,
    INotifyPropertyChanged, ICustomTypeDescriptor
```

## Members

All Members

Constructors

Methods

Properties

Events

 Public Instance Declared Protected Static Inherited

Icon

Member

Description

[JObject\(\)](#)

Initializes a new instance of the JObject class.

[JObject\(JObject\)](#)

Initializes a new instance of the JObject class from another JObject object.

≡ ♪	<a href="#">JObject(Object[])</a>	Initializes a new instance of the JObject class.
≡ ♪	<a href="#">JObject(Object)</a>	Initializes a new instance of the JObject class.
≡ ♪	<a href="#">Add(String, JToken)</a>	Adds a child token.
≡ ♪	<a href="#">Add(Object)</a>	Adds a child object.
≡ ♪	<a href="#">AddAfterSelf(Object)</a>	Adds an immediate child after self.
≡ ♪	<a href="#">AddBeforeSelf(Object)</a>	Adds an immediate child before self.
≡ ♪	<a href="#">AddFirst(Object)</a>	Adds an immediate child at the first position.
⚡	<a href="#">AddingNew</a>	Occurs when the current node is added.
≡ ♪	<a href="#">AfterSelf()</a>	Returns the current node.

			token order
		(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="#">CollectionChanged</a>	Occu colle	colle
		(Inhe	
💻	<a href="#">Count</a>	Gets conta	
≡	<a href="#">CreateReader()</a>	Create token	
		(Inhe	
≡	<a href="#">CreateWriter()</a>	Create used	
		(Inhe	
≡	<a href="#">DeepClone()</a>	Create	

		JTok
		recui
		(Inhe
≡	<a href="#">Descendants()</a>	Retu desc docu
		(Inhe
≡	<a href="#">Equals(Object)</a>	Dete <u>Obje</u>
		(Inhe
💡	<a href="#">Finalize()</a>	Allow reso oper recla
		(Inhe
📄	<a href="#">First</a>	Get t
		(Inhe
≡	<a href="#"><b>S</b> <u>FromObject(Object)</u></a>	Crea
≡	<a href="#"><b>S</b> <u>FromObject(Object, JsonSerializer)</u></a>	Crea
≡	<a href="#">GetEnumerator()</a>	Retu throu
≡	<a href="#">GetHashCode()</a>	Serv parti
		(Inhe

	<a href="#">GetMetaObject(Expression)</a>	Returns the meta object for the current item.
	<a href="#">GetType()</a>	Gets the type of the current item.
	<a href="#">HasValues</a>	Gets whether the current item has values.
	<a href="#">IsValid(JToken, JsonSchema)</a>	Determines if the current item is valid according to the specified schema.
	<a href="#">Item[Object]</a>	Gets the value of the current item as an object.
	<a href="#">Item[String]</a>	Gets the value of the current item as a string.
	<a href="#">Last</a>	Gets the last item in the list.
	<a href="#">ListChanged</a>	Occurs when the list changes.
	<a href="#">Load(JsonReader)</a>	Loads the list from a JSON reader.

	 <a href="#">MemberwiseClone()</a>	Create current
	 <a href="#">Next</a>	(Inherited) Gets node
	 <a href="#">OnAddingNew(AddingNewEventArgs)</a>	Raised
	 <a href="#">OnCollectionChanged(NotifyCollectionChangedEventArgs)</a>	(Inherited) Raises event
	 <a href="#">OnListChanged(ListChangedEventArgs)</a>	(Inherited) Raises
	 <a href="#">OnPropertyChanged(String)</a>	(Inherited) Raises with
	 <a href="#">OnPropertyChanging(String)</a>	(Inherited) Raises with
	 <a href="#">Parent</a>	Gets
	 <a href="#"><b>S</b> Parse(String)</a>	(Inherited) Loads content
	 <a href="#">Previous</a>	Gets

		this r
		(Inhe
≡	<a href="#"><u>Properties()</u></a>	Gets objec
≡	<a href="#"><u>Property(String)</u></a>	Gets name
⚡	<a href="#"><u>PropertyChanged</u></a>	Occu chan
⚡	<a href="#"><u>PropertyChanging</u></a>	Occu chan
≡	<a href="#"><u>PropertyValues()</u></a>	Gets objec
≡	<a href="#"><u>Remove(String)</u></a>	Rem spec
≡	<a href="#"><u>Remove()</u></a>	Rem
		(Inhe
≡	<a href="#"><u>RemoveAll()</u></a>	Rem toker
		(Inhe
≡	<a href="#"><u>Replace(JToken)</u></a>	Repl spec

		(Inhe
≡	<a href="#">ReplaceAll(Object)</a>	Repl tokei
		(Inhe
⊕	<a href="#">Root</a>	Gets
		(Inhe
≡	<a href="#">SelectToken(String)</a>	Sele objec
		(Inhe
≡	<a href="#">SelectToken(String, Boolean)</a>	Sele objec
		(Inhe
≡	<a href="#">ToString()</a>	Retu tokei
		(Inhe
≡	<a href="#">ToString(Formatting, JsonConverter[])</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>	Validates the current token against the specified schema.
		(Inherited from <a href="#">Object</a> )
≡	<a href="#">Value&lt;T&gt;(Object)</a>	Gets the value of the current token.
		(Inherited from <a href="#">Object</a> )
≡	<a href="#">Values&lt;T&gt;()</a>	Returns all values of the current token.
		(Inherited from <a href="#">Object</a> )
≡	<a href="#">WriteTo(JsonWriter, JsonConverter[])</a>	Writes the current token to the specified writer.
		(Overridden from <a href="#">JToken</a> )

## [-] Inheritance Hierarchy

[Object](#)  
 └ [JToken](#)  
   └ [JContainer](#)  
     └ **JObject**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



C# ▾

## - 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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

# JObject Constructor (other)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► [JObject\(JObject\)](#)

C#

Initializes a new instance of the [JObject](#) class from another [JObject](#) object.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JObject(  
    JObject other  
)
```

```
Public Sub New (  
    other As JObject  
)
```

```
public:  
JObject(  
    JObject^ other  
)
```

## Parameters

### *other* ([JObject](#))

A [JObject](#) object to copy from.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JObject Constructor (content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **JObject(Object)**

C#

Initializes a new instance of the [JObject](#) class with the specified content

## 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: 4.0.2.C  
(4.0.2.13623)

# JObject Constructor (content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **JObject(Object[])**

C#

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: 4.0.2.C  
(4.0.2.13623)

## Add Method (*propertyName*, *value*)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ►  [JObject](#) ► **Add(String, JToken)**

C# ▾

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.

# Count Property

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JObject](#) ► **Count**

C# ▾

Gets the number of elements contained in the [ICollection<T>](#).

## 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

## Return Value

The number of elements contained in the [ICollection<T>](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# FromObject Method

[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: 4.0.2.0  
(4.0.2.13623)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ►  [JObject](#) ►

## FromObject(Object)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# 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: 4.0.2.C  
(4.0.2.13623)

C# ▾

Returns an enumerator that iterates through the collection.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public IEnumerator<KeyValuePair<string, JToken>> GetEnumerator()
```

```
Public Function GetEnumerator As IEnumerator\(Of KeyValuePair\(Of String, JToken\)\)
```

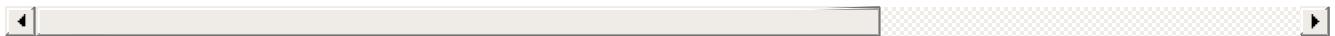
```
public:  
virtual IEnumerator<KeyValuePair<String^, JToken^>>^
```

## Return Value

A [IEnumerator<T>](#) that can be used to iterate through the collection.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



# GetMetaObject Method (parameter)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ►  [JObject](#) ►  
**GetMetaObject(Expression)**

C#

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# Item Property

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ►  [JObject](#) ► **Item**

C# ▾

## - 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: 4.0.2.0 (4.0.2.13623)

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) override;
```

## Parameters

[key \(Object\)](#)

## Value

The [JToken](#) with the specified key.

# Item Property (*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)
}
```

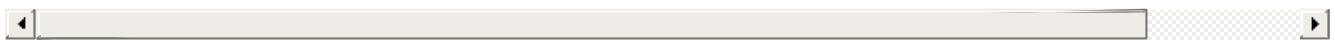
## Parameters

*propertyName* ([String](#))

## Value

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



C# ▾

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](#).

# OnPropertyChanged

## Method (propertyName)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ►  [JObject](#) ►

**OnPropertyChanged(String)**

C#

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: 4.0.2.0  
(4.0.2.13623)

# OnPropertyChanging Method (propertyName)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ►  [JObject](#) ►

**OnPropertyChanging(String)**

C# ▾

Raises the [PropertyChanging](#) event with the provided arguments.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual void OnPropertyChanging(  
    string propertyName  
)
```

```
Protected Overridable Sub OnPropertyChanging ( _  
    propertyName As String _  
)
```

```
protected:  
virtual void OnPropertyChanging(  
    String^ propertyName  
)
```

## Parameters

**propertyName (String)**

Name of the property.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# PropertyChanged Event

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ►  [JObject](#) ► [PropertyChanged](#)

C# ▾

Occurs when a property value changes.

## Declaration Syntax

C#

Visual Basic

Visual C++

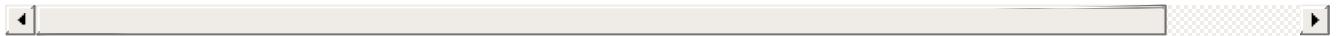
```
public event PropertyChangedEventHandler PropertyChanged
```

```
Public Event PropertyChanged As PropertyChangedEvent
```

```
public:  
virtual event PropertyChangedEventHandler^ PropertyChanged  
void add (PropertyChangedEventHandler^ value)  
void remove (PropertyChangedEventHandler^ value)  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



Json.NET - Quick Starts & API Documentation

# PropertyChanging Event

[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^ Property  
void add (PropertyChangingEventHandler^ value)  
void remove (PropertyChangingEventHandler^ value)  
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



Json.NET - Quick Starts & API Documentation

## PropertyValues Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ►  [JObject](#) ► [PropertyValues\(\)](#)

C# ▾

Gets an [JEnumerable<T>](#) of this object's property values.

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public JEnumerable<JToken> PropertyValues()
```

```
Public Function PropertyValues As JEnumerable(Of JTo
```

```
public:  
JEnumerable<JToken>^ PropertyValues()
```

### Return Value

An [JEnumerable<T>](#) of this object's property values.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



# 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: 4.0.2.0  
(4.0.2.13623)

# TryGetValue Method (*propertyName*, *value*)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ►  [JObject](#) ► **TryGetValue(String, JToken)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

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: 4.0.2.0 (4.0.2.13623)

# WriteTo Method (writer, converters)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ►  [JObject](#) ► **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.

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

 Public Instance Declared Protected Static Inherited

Icon

Member

Description

[JProperty\(JProperty\)](#)

Initializes a new instance of the `JProperty` class from another object.

[JProperty\(String, Object\[\]\)](#)

Initializes a new instance of the `JProperty` class.

[JProperty\(String, Object\)](#)

Initializes a new instance of the `JProperty` class.

	<a href="#">Add(Object)</a>	Adds as child  (Inherited from <a href="#">JCor</a> )
	<a href="#">AddAfterSelf(Object)</a>	Adds immediately after self  (Inherited from <a href="#">JCor</a> )
	<a href="#">AddBeforeSelf(Object)</a>	Adds immediately before self  (Inherited from <a href="#">JCor</a> )
	<a href="#">AddFirst(Object)</a>	Adds as the first token  (Inherited from <a href="#">JTok</a> )
	<a href="#">AddingNew</a>	Occurs when adding  (Inherited from <a href="#">JCor</a> )
	<a href="#">AfterSelf()</a>	Returns sibling tokens  (Inherited from <a href="#">JCor</a> )
	<a href="#">Ancestors()</a>	Returns ancestors tokens  (Inherited from <a href="#">JCor</a> )
		

	<a href="#">BeforeSelf()</a>	Retu siblir tokei
⚡	<a href="#">Children()</a>	Retu child in do
⚡	<a href="#">CollectionChanged</a>	(Over JCor
	<a href="#">CreateReader()</a>	Occu of the chan is res
⚡	<a href="#">CreateWriter()</a>	(Inhe JCor
	<a href="#">DeepClone()</a>	Create this t
⚡	<a href="#">Descendants()</a>	(Inhe

		(Inhe JCor
	<a href="#">Equals(Object)</a>	Dete spec the c
	<a href="#">Finalize()</a>	(Inhe Allow atten and j oper <u>Obje</u> garb
	<a href="#">First</a>	(Inhe Get t this t
	<a href="#">GetHashCode()</a>	(Inhe JCor
	<a href="#">GetMetaObject(Expression)</a>	Serv for a
	<a href="#">GetType()</a>	(Inhe Retu <u>Dyna</u> respo oper this c
	<a href="#">HasValues</a>	(Inhe Gets curre

		what child
		(Inhe JCor
⌚	<a href="#">IsValid(JToken, JsonSchema)</a>	Dete JTok
		(Inhe Exte
📝	<a href="#">Item[Object]</a>	Gets spec
		(Inhe
📝	<a href="#">Last</a>	Get t this t
		(Inhe JCor
⚡	<a href="#">ListChanged</a>	Occu chan list c
		(Inhe JCor
⌚ S	<a href="#">Load(JsonReader)</a>	Load a Js
⌚	<a href="#">MemberwiseClone()</a>	Create the c
		(Inhe
📝	<a href="#">Name</a>	Gets

	<a href="#">Next</a>	Gets of thi  (Inhe JCor
	<a href="#">OnAddingNew(AddingNewEventArgs)</a>	Rais even  (Inhe JCor
	<a href="#">OnCollectionChanged(NotifyCollectionChangedEventArgs)</a>	Rais Colle  (Inhe JCor
	<a href="#">OnListChanged(ListChangedEventArgs)</a>	Rais even  (Inhe JCor
	<a href="#">Parent</a>	Gets  (Inhe
	<a href="#">Previous</a>	Gets toke  (Inhe
	<a href="#">Remove()</a>	Rem its pa  (Inhe
	<a href="#">RemoveAll()</a>	Rem from  (Inhe JCor
	<a href="#">Replace(JToken)</a>	Repl

		the s
		(Inhe
≡	<a href="#">ReplaceAll(Object)</a>	Repl
		node
		spec
		(Inhe
		JCor
≡	<a href="#">Root</a>	Gets
		this .
		(Inhe
≡	<a href="#">SelectToken(String)</a>	Select
		match
		(Inhe
≡	<a href="#">SelectToken(String, Boolean)</a>	Select
		match
		(Inhe
≡	<a href="#">ToString()</a>	Retu
		JSOI
		(Inhe
≡	<a href="#">ToString(Formatting, JsonConverter[])</a>	Retu
		token
		form
		(Inhe
≡	<a href="#">Type</a>	Gets
		JTok
		(Ove
≡	<a href="#">Validate(JToken, JsonSchema)</a>	Valid
		JTok

		(Inhe Exte
≡	<a href="#">Validate(JToken, JsonSchema, ValidationEventHandler)</a>	Valid JTok
		(Inhe Exte
⊕	<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 in do
		(Inhe JCor
≡	<a href="#">WriteTo(JsonWriter, JsonConverter[])</a>	Write Json
		(Ove JTok Json

## [-] Inheritance Hierarchy

[Object](#)  
 └ [JToken](#)  
   └ [JContainer](#)  
     └ [JProperty](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



# JProperty Constructor

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JProperty](#) ► [JProperty\(\)](#)

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: 4.0.2.0  
(4.0.2.13623)

# JProperty Constructor (other)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JProperty](#) ► [JProperty\(JProperty\)](#)

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JProperty Constructor (name, content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JProperty](#) ► [JProperty\(String, Object\)](#)

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JProperty(  
    string name,  
    object content  
)
```

```
Public Sub New ( _  
    name As String, _  
    content As Object _  
)
```

```
public:  
JProperty(  
    String^ name,  
    Object^ content  
)
```

## Parameters

### **name** ([String](#))

The property name.

### **content** ([Object](#))

The property content.

# JProperty Constructor (name, content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JProperty](#) ► [JProperty\(String, Object\[\]\)](#)

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JProperty(  
    string name,  
    params Object[] content  
)
```

```
Public Sub New ( _  
    name As String, _  
    ParamArray content As Object() _  
)
```

```
public:  
JProperty(  
    String^ name,  
    ... array<Object^>^ content  
)
```

## Parameters

### **name** ([String](#))

The property name.

### **content** ([Object\[\]](#))

The property content.

C# ▾

Returns a collection of the child tokens of this token, in document order.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public override JEnumerable<JToken> Children()
```

```
Public Overrides Function Children As JEnumerable(Of
```

```
public:  
virtual JEnumerable<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: 4.0.2.C  
(4.0.2.13623)



▾

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: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

# WriteTo Method (writer, converters)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JProperty](#) ► **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.

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

 Public Protected Instance Static

Declared



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="#">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>JToken</a> .)
≡	<a href="#">Replace(JToken)</a>	Replaces this token with the specified token. (Inherited from <a>JToken</a> .)
⊕	<a href="#">Root</a>	Gets the root <a>JToken</a> of this <a>JToken</a> . (Inherited from <a>JToken</a> .)
≡	<a href="#">SelectToken(String)</a>	Selects the token that matches the object path. (Inherited from <a>JToken</a> .)
≡	<a href="#">SelectToken(String, Boolean)</a>	Selects the token that matches the object path. (Inherited from <a>JToken</a> .)
≡	<a href="#">ToString()</a>	Returns a <a>String</a> that represents this instance. (Inherited from <a>JValue</a> .)
≡	<a href="#">ToString(String)</a>	Returns a <a>String</a> that represents this instance. (Inherited from <a>JValue</a> .)
≡	<a href="#">ToString(IFormatProvider)</a>	Returns a <a>String</a> that represents this instance. (Inherited from <a>JValue</a> .)
≡	<a href="#">ToString(String, IFormatProvider)</a>	Returns a <a>String</a> that represents this instance. (Inherited from <a>JValue</a> .)
≡	<a href="#">ToString(Formatting)</a>	Returns the JSON for this token

	<a href="#">JsonConverter[])</a>	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, JsonConverter[])</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: 4.0.2.0  
(4.0.2.13623)

**[-] 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: 4.0.2.0  
(4.0.2.13623)

C# ▾

Initializes a new instance of the [JRaw](#) class from another [JRaw](#) object.

## Declaration Syntax

C#

Visual Basic

Visual C++  
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: 4.0.2.0 (4.0.2.13623)

# 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: 4.0.2.C  
(4.0.2.13623)

# Create Method (reader)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JRaw](#) ► [Create\(JsonReader\)](#)

C# ▾

Creates an instance of [JRaw](#) with the content of the reader's current token.

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static JRaw Create(  
    JsonReader reader  
)
```

```
Public Shared Function Create ( _  
    reader As JsonReader _  
) As JRaw
```

```
public:  
static JRaw^ Create(  
    JsonReader^ reader  
)
```

## 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: 4.0.2.0 (4.0.2.13623)

Represents an abstract JSON token.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public abstract class JToken : IJEnumerable<JToken>,  
    IEnumerable<JToken>, IEnumerable, IJsonLineInfo,  
    IDynamicMetaObjectProvider
```

```
Public MustInherit Class JToken  
    Implements IJEnumerable\(Of JToken\), IEnumerable,  
    IJsonLineInfo, ICloneable, IDyn
```

```
public ref class JToken abstract : IJEnumerable<JTok  
    IEnumerable<JToken^>, IEnumerable, IJsonLine  
    IDynamicMetaObjectProvider
```

## Members

All Members

Methods

Properties

 Public Protected Instance Static

Declared



Inherited

Icon

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="#">IJEnumerable&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="#">IJEnumerable&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.
≡ S	<a href="#">DeepEquals(JToken, JToken)</a>	Compares the values of two tokens, including the values of all descendant tokens.
≡ S	<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> .)
≡ S	<a href="#">Explicit(JToken to Boolean)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Boolean</a> .
≡ S	<a href="#">Explicit(JToken to DateTimeOffset)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">DateTimeOffset</a> .
≡ S	<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> .
≡ S	<a href="#">Explicit(JToken to Int64)</a>	Performs an explicit conversion from <a href="#">JToken</a> to <a href="#">Int64</a> .
≡ S	<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> .
≡ S	<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 <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 <a href="#">Byte[]</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.
 <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>		Serves as a hash function for a particular type.
		(Inherited from <a href="#">Object</a> .)
 <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 <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; to 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 <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&lt;Nullable&lt;Single&gt; to JToken&gt;</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .
 <a href="#">Implicit&lt;Decimal to JToken&gt;</a>	Performs an implicit conversion from <a href="#">Decimal</a> to <b>JToken</b> .
 <a href="#">Implicit&lt;Nullable&lt;Int16&gt; to JToken&gt;</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .
 <a href="#">Implicit&lt;Nullable&lt;UInt16&gt; to JToken&gt;</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .
 <a href="#">Implicit&lt;Nullable&lt;UInt32&gt; to JToken&gt;</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .
 <a href="#">Implicit&lt;Nullable&lt;UInt64&gt; to JToken&gt;</a>	Performs an implicit conversion from <a href="#">Nullable&lt;T&gt;</a> to <b>JToken</b> .
 <a href="#">Implicit&lt;Double to JToken&gt;</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 <a href="#">Byte[]</a> 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="#">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="#">ToString()</a>	Returns the indented JSON for this token.  (Overrides <a href="#">Object.ToString()</a> .)	
 <a href="#">ToString(Formatting, JsonConverter[])</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="#"><u>Values&lt;U&gt;(IEnumerable&lt;JToken&gt;, Object)</u></a>	Returns a collection of converted child values of every object in the source collection with the given key.  (Inherited from <a href="#">Extensions</a> .)
≡	<a href="#"><u>WriteTo(JsonWriter, JsonConverter[])</u></a>	Writes this token to a <a href="#">JsonWriter</a> .

## [-] Inheritance Hierarchy

[Object](#)

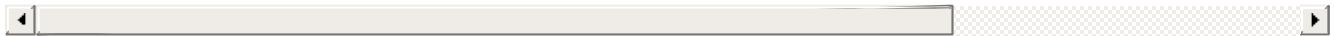
└ [JToken](#)

  └ [JContainer](#)

  └ [JValue](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



# AddAfterSelf Method (content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

[AddAfterSelf\(Object\)](#)

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

## AddBeforeSelf Method (content)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

**AddBeforeSelf(Object)**

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)

# Ancestors Method

[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: 4.0.2.C  
(4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)

C# ▾

## 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.
≡ ♀ 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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)



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 JEnumerable<T>
```

```
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: 4.0.2.0  
(4.0.2.13623)



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: 4.0.2.C  
(4.0.2.13623)

# DeepClone Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► [DeepClone\(\)](#)

C# ▾

Creates a new instance of the [JToken](#). All child tokens are recursively cloned.

## 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: 4.0.2.C  
(4.0.2.13623)

# DeepEquals Method (t1, t2)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **DeepEquals(JToken, JToken)**

C#

Compares the values of two tokens, including the values of all descendant tokens.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static bool DeepEquals(
    JToken t1,
    JToken t2
)
```

```
Public Shared Function DeepEquals ( _
    t1 As JToken, _
    t2 As JToken _
) As Boolean
```

```
public:
static bool DeepEquals(
    JToken^ t1,
    JToken^ t2
)
```

## 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: 4.0.2.C  
(4.0.2.13623)

# EqualityComparer

## Property

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **EqualityComparer**

C# ▾

Gets a comparer that can compare two tokens for value equality.

### 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: 4.0.2.C  
(4.0.2.13623)



# Explicit Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► [Explicit\(\)](#)

C# ▾

## 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 Nullable&lt;Double&gt;)</a>	Performs an explicit conversion from <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 <a href="#">Byte[]</a> .

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Decimal)**

Performs an explicit conversion from [JToken](#) to [Decimal](#).

C# ▾

## 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: 4.0.2.0 (4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<UInt32>)**

C# ▾

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static explicit operator Nullable<uint> ( JToken value )
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Nullable\(Of UInteger\)
```

```
static explicit operator Nullable<unsigned int> ( JToken^ value )
```

## Parameters

### **value (JToken)**

The value.

## Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## 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: 4.0.2.0 (4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to DateTime)**

C# ▾

Performs an explicit conversion from [JToken](#) to [DateTime](#).

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static explicit operator DateTime (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As DateTime
```

```
static explicit operator DateTime (  
    JToken^ value  
)
```

## Parameters

### ***value* ([JToken](#))**

The value.

## Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<Int64>)**

C# ▾

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static explicit operator Nullable<long> (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Nullable(Of Long)
```

```
static explicit operator Nullable<long long> (  
    JToken^ value  
)
```

## Parameters

### **value (JToken)**

The value.

## Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<Single>)**

C# ▾

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static explicit operator Nullable<float> (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Nullable(Of Single)
```

```
static explicit operator Nullable<float> (  
    JToken^ value  
)
```

## Parameters

### **value (JToken)**

The value.

## Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to UInt32)**

C# ▾

Performs an explicit conversion from [JToken](#) to [UInt32](#).

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static explicit operator uint (
    JToken value
)
```

```
Public Shared Narrowing Operator CType ( _
    value As JToken _
) As UInt32
```

```
static explicit operator unsigned int (
    JToken^ value
)
```

## Parameters

### **value (JToken)**

The value.

## Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to UInt64)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Byte[])**

C# ▾

Performs an explicit conversion from [JToken](#) to [Byte\[\]](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator byte[] (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Byte()
```

```
static explicit operator array<unsigned char>^ (  
    JToken^ value  
)
```

## Parameters

### **value (JToken)**

The value.

## Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Double)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Single)**

C# ▾

Performs an explicit conversion from [JToken](#) to [Single](#).

## Declaration Syntax

C#

Visual Basic

Visual C++  
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: 4.0.2.C  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to String)**

C# ▾

Performs an explicit conversion from [JToken](#) to [String](#).

## Declaration Syntax

C#

Visual Basic

Visual C++  
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: 4.0.2.C  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<DateTime>)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

C# ▾

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static explicit operator Nullable<DateTimeOffset>
    JToken value
)
```

```
Public Shared Narrowing Operator CType ( _
    value As JToken _
) As Nullable(Of DateTimeOffset)
```

```
static explicit operator Nullable<DateTimeOffset> (
    JToken^ value
)
```

## Parameters

### ***value* ([JToken](#))**

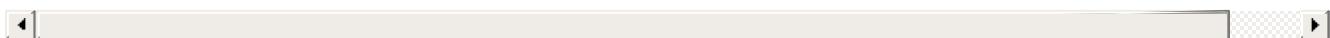
The value.

## Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<Decimal>)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Int64)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Boolean)**

C# ▾

Performs an explicit conversion from [JToken](#) to [Boolean](#).

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static explicit operator bool (
    JToken value
)
```

```
Public Shared Narrowing Operator CType ( _
    value As JToken _
) As Boolean
```

```
static explicit operator bool (
    JToken^ value
)
```

## Parameters

### **value (JToken)**

The value.

## Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<Boolean>)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<Int32>)**

C# ▾

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static explicit operator Nullable<int> (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Nullable(Of Integer)
```

```
static explicit operator Nullable<int> (  
    JToken^ value  
)
```

## Parameters

### **value (JToken)**

The value.

## Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## 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: 4.0.2.C (4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<UInt16>)**

C# ▾

Performs an explicit conversion from [JToken](#) to [Nullable<T>](#).

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static explicit operator Nullable<ushort> (  
    JToken value  
)
```

```
Public Shared Narrowing Operator CType ( _  
    value As JToken _  
) As Nullable\(of UShort\)
```

```
static explicit operator Nullable<unsigned short> (  
    JToken^ value  
)
```

## Parameters

### **value (JToken)**

The value.

## Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to UInt16)**

C# ▾

Performs an explicit conversion from [JToken](#) to [UInt16](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static explicit operator ushort (
    JToken value
)
```

```
Public Shared Narrowing Operator CType ( _
    value As JToken _
) As UShort
```

```
static explicit operator unsigned short (
    JToken^ value
)
```

## Parameters

### **value (JToken)**

The value.

## Return Value

The result of the conversion.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Int32)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Nullable<Double>)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# Explicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Explicit(JToken to Int16)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)

# FromObject Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **FromObject()**

C# ▾

## - 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: 4.0.2.0 (4.0.2.13623)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# FromObject Method (o, jsonSerializer)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►   
**FromObject(Object, JsonSerializer)**

Creates a [JToken](#) from an object using the specified [JsonSerializer](#).

C#

## 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: 4.0.2.C  
(4.0.2.13623)

# GetMetaObject Method (parameter)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

[GetMetaObject\(Expression\)](#)

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

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 Boolean  
    Get
```

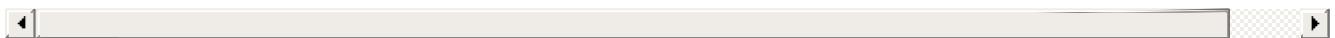
```
public:  
virtual property bool HasValues {  
    bool get () abstract;  
}
```

## Value

true if this token has child values; otherwise, false.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# Implicit Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► [Implicit\(\)](#)

C# ▾

## ‑ 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&lt;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&lt;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&lt;Int16 to JToken)</a>	Performs an implicit conversion from <a href="#">Int16</a> to <a href="#">JToken</a> .
	<a href="#">Implicit&lt;UInt16 to JToken)</a>	Performs an implicit conversion from <a href="#">UInt16</a> to <a href="#">JToken</a> .
	<a href="#">Implicit&lt;Int32 to JToken)</a>	Performs an implicit conversion from <a href="#">Int32</a> to <a href="#">JToken</a> .
	<a href="#">Implicit&lt;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&lt;DateTime to JToken)</a>	Performs an implicit conversion from <a href="#">DateTime</a> to <a href="#">JToken</a> .
	<a href="#">Implicit&lt;Nullable&lt;Int64&gt; to</a>	Performs an implicit

	<a href="#">JToken)</a>	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 <a href="#">Byte[]</a> to <a href="#">JToken</a> .

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# Implicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(Boolean to JToken)**

C# ▾

Performs an implicit conversion from [Boolean](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static implicit operator JToken (
    bool value
)
```

```
Public Shared Widening Operator CType ( _
    value As Boolean _
) As JToken
```

```
static implicit operator JToken^ (
    bool value
)
```

## Parameters

### **value ([Boolean](#))**

The value to create a [JValue](#) from.

## Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# Implicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(Byte[] to JToken)**

C# ▾

Performs an implicit conversion from **Byte[]** to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static implicit operator JToken (
    byte[] value
)
```

```
Public Shared Widening Operator CType ( _
    value As Byte() _
) As JToken
```

```
static implicit operator JToken^ (
    array<unsigned char>^ value
)
```

## 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: 4.0.2.C  
(4.0.2.13623)

# Implicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(DateTime to JToken)**

C# ▾

Performs an implicit conversion from [DateTime](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static implicit operator JToken (
    DateTime value
)
```

```
Public Shared Widening Operator CType ( _
    value As DateTime _
) As JToken
```

```
static implicit operator JToken^ (
    DateTime value
)
```

## Parameters

### **value ([DateTime](#))**

The value to create a [JValue](#) from.

## Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C#

Performs an implicit conversion from [DateTimeOffset](#) to [JToken](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public static implicit operator JToken (
    DateTimeOffset value
)
```

```
Public Shared Widening Operator CType ( _
    value As DateTimeOffset _
) As JToken
```

```
static implicit operator JToken^ (
    DateTimeOffset value
)
```

### Parameters

#### **value ([DateTimeOffset](#))**

The value to create a [JValue](#) from.

### Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# Implicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(Decimal to JToken)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# Implicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(Double to JToken)**

C# ▾

Performs an implicit conversion from [Double](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++  
C++

```
public static implicit operator JToken (
    double value
)
```

```
Public Shared Widening Operator CType ( _
    value As Double _
) As JToken
```

```
static implicit operator JToken^ (
    double value
)
```

## Parameters

### **value ([Double](#))**

The value to create a [JValue](#) from.

## Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# Implicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(Int16 to JToken)**

C# ▾

Performs an implicit conversion from [Int16](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++  
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: 4.0.2.C  
(4.0.2.13623)

C# ▾

Performs an implicit conversion from [Int32](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++  
C++

```
public static implicit operator JToken (
    int value
)
```

```
Public Shared Widening Operator CType ( _
    value As Integer _
) As JToken
```

```
static implicit operator JToken^ (
    int value
)
```

## 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: 4.0.2.C  
(4.0.2.13623)

C# ▾

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static implicit operator JToken (
    long value
)
```

```
Public Shared Widening Operator CType ( _
    value As Long _
) As JToken
```

```
static implicit operator JToken^ (
    long long value
)
```

## 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: 4.0.2.0 (4.0.2.13623)

C#

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static implicit operator JToken (  
    Nullable<bool> value  
)
```

```
Public Shared Widening Operator CType ( _  
    value As Nullable(Of Boolean) _  
) As JToken
```

```
static implicit operator JToken^ ( _  
    Nullable<bool> value  
)
```

## Parameters

### **value** ([Nullable<Boolean>](#))

The value to create a [JValue](#) from.

## Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# Implicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(Nullable<DateTime> to JToken)

C# ▾

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static implicit operator JToken (
    Nullable<DateTime> value
)
```

```
Public Shared Widening Operator CType ( _
    value As Nullable(Of DateTime) _
) As JToken
```

```
static implicit operator JToken^ (
    Nullable<DateTime> value
)
```

### Parameters

#### **value** ([Nullable<DateTime>](#))

The value to create a [JValue](#) from.

### Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(Nullable<DateTimeOffset> to JToken)

C# ▾

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static implicit operator JToken (
    Nullable<DateTimeOffset> value
)
```

```
Public Shared Widening Operator CType ( _
    value As Nullable(Of DateTimeOffset) _ 
) As JToken
```

```
static implicit operator JToken^ (
    Nullable<DateTimeOffset> value
)
```

### Parameters

#### **value ([Nullable<DateTimeOffset>](#))**

The value to create a [JValue](#) from.

### Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# Implicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit<Nullable<Decimal> to JToken)

C#

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public static implicit operator JToken (  
    Nullable<decimal> value  
)
```

```
Public Shared Widening Operator CType ( _  
    value As Nullable(Of Decimal) _  
) As JToken
```

```
static implicit operator JToken^ (  
    Nullable<Decimal> value  
)
```

### Parameters

#### **value** ([Nullable<Decimal>](#))

The value to create a [JValue](#) from.

### Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static implicit operator JToken (
    Nullable<double> value
)
```

```
Public Shared Widening Operator CType ( _
    value As Nullable(Of Double) _
) As JToken
```

```
static implicit operator JToken^ (
    Nullable<double> value
)
```

## Parameters

### **value ([Nullable<Double>](#))**

The value to create a [JValue](#) from.

## Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static implicit operator JToken (
    Nullable<short> value
)
```

```
Public Shared Widening Operator CType ( _
    value As Nullable\(Of Short\) ) _
) As JToken
```

```
static implicit operator JToken^ (
    Nullable<short> value
)
```

## Parameters

### **value ([Nullable<Int16>](#))**

The value to create a [JValue](#) from.

## Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# Implicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

## Implicit(Nullable<Int32> to JToken)

C# ▾

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public static implicit operator JToken (
    Nullable<int> value
)
```

```
Public Shared Widening Operator CType ( _
    value As Nullable(Of Integer) _
) As JToken
```

```
static implicit operator JToken^ (
    Nullable<int> value
)
```

### Parameters

#### **value** ([Nullable<Int32>](#))

The value to create a [JValue](#) from.

### Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static implicit operator JToken (
    Nullable<long> value
)
```

```
Public Shared Widening Operator CType ( _
    value As Nullable(Of Long) _
) As JToken
```

```
static implicit operator JToken^ (
    Nullable<long long> value
)
```

## Parameters

### **value** ([Nullable<Int64>](#))

The value to create a [JValue](#) from.

## Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static implicit operator JToken (
    Nullable<float> value
)
```

```
Public Shared Widening Operator CType ( _
    value As Nullable(Of Single) _
) As JToken
```

```
static implicit operator JToken^ (
    Nullable<float> value
)
```

## Parameters

### **value** ([Nullable<Single>](#))

The value to create a [JValue](#) from.

## Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static implicit operator JToken (
    Nullable<ushort> value
)
```

```
Public Shared Widening Operator CType ( _
    value As Nullable(Of UShort) _
) As JToken
```

```
static implicit operator JToken^ (
    Nullable<unsigned short> value
)
```

## Parameters

### **value** ([Nullable<UInt16>](#))

The value to create a [JValue](#) from.

## Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static implicit operator JToken (
    Nullable<uint> value
)
```

```
Public Shared Widening Operator CType ( _
    value As Nullable\(Of UInteger\) _
) As JToken
```

```
static implicit operator JToken^ (
    Nullable<unsigned int> value
)
```

## Parameters

### **value** ([Nullable<UInt32>](#))

The value to create a [JValue](#) from.

## Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

C# ▾

Performs an implicit conversion from [Nullable<T>](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public static implicit operator JToken (
    Nullable<ulong> value
)
```

```
Public Shared Widening Operator CType ( _
    value As Nullable\(Of ULong\) _
) As JToken
```

```
static implicit operator JToken^ (
    Nullable<unsigned long long> value
)
```

## Parameters

### **value** ([Nullable<UInt64>](#))

The value to create a [JValue](#) from.

## Return Value

The [JValue](#) initialized with the specified value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# Implicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(Single to JToken)**

C# ▾

Performs an implicit conversion from [Single](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++  
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: 4.0.2.C  
(4.0.2.13623)

# Implicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(String to JToken)**

C# ▾

Performs an implicit conversion from [String](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++  
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: 4.0.2.C  
(4.0.2.13623)

# Implicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(UInt16 to JToken)**

C# ▾

Performs an implicit conversion from [UInt16](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++  
C++

```
public static implicit operator JToken (
    ushort value
)
```

```
Public Shared Widening Operator CType ( _
    value As UShort _
) As JToken
```

```
static implicit operator JToken^ (
    unsigned short value
)
```

## 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: 4.0.2.C  
(4.0.2.13623)

# Implicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(UInt32 to JToken)**

C# ▾

Performs an implicit conversion from [UInt32](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++  
C++

```
public static implicit operator JToken (
    uint value
)
```

```
Public Shared Widening Operator CType ( _
    value As UInteger _
) As JToken
```

```
static implicit operator JToken^ (
    unsigned int value
)
```

## 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: 4.0.2.C  
(4.0.2.13623)

# Implicit Operator

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **Implicit(UInt64 to JToken)**

C# ▾

Performs an implicit conversion from [UInt64](#) to [JToken](#).

## Declaration Syntax

C#

Visual Basic

Visual C++  
C++

```
public static implicit operator JToken (
    ulong value
)
```

```
Public Shared Widening Operator CType ( _
    value As ULong _
) As JToken
```

```
static implicit operator JToken^ (
    unsigned long long value
)
```

## 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: 4.0.2.C  
(4.0.2.13623)

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.

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: 4.0.2.C  
(4.0.2.13623)

C# ▾

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.

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: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)

C# ▾

Gets the previous sibling token of this node.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JToken Previous { get; }
```

```
Public ReadOnly Property Previous As JToken
    Get
```

```
public:
property JToken^ Previous {
    JToken^ get ();
}
```

## Value

The [JToken](#) that contains the previous sibling token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# ReadFrom Method (reader)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►   
**ReadFrom(JsonReader)**

C# ▾

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.



Removes this token from its parent.

## Declaration Syntax

```
public void Remove()
```

```
Public Sub Remove
```

```
public:  
void Remove()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

C# ▾

Replaces this token with the specified token.

## Declaration Syntax

C#

Visual Basic

Visual C++  
C++

```
public void Replace(  
    JToken value  
)
```

```
Public Sub Replace ( _  
    value As JToken _  
)
```

```
public:  
void Replace(  
    JToken^ value  
)
```

## Parameters

### **value (JToken)**

The value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



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: 4.0.2.0  
(4.0.2.13623)

# SelectToken Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **SelectToken()**

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: 4.0.2.0 (4.0.2.13623)

# SelectToken Method (path)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ► **SelectToken(String)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# SelectToken Method (path, errorWhenNoMatch)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

[SelectToken\(String, Boolean\)](#)

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

C# ▾

## - 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, JsonConverter[])</a>	Returns the JSON for this token using the given formatting and converters.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)

# ToString Method (formatting, converters)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

**ToString(Formatting, JsonConverter[])**

C# ▾

Returns the JSON for this token using the given formatting and converters.

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public string ToString(  
    Formatting formatting,  
    params JsonConverter[] converters  
)
```

```
Public Function ToString ( _  
    formatting As Formatting, _  
    ParamArray converters As JsonConverter() _  
) As String
```

```
public:  
String^ ToString(  
    Formatting formatting,  
    ... array<JsonConverter>^ converters  
)
```

## Parameters

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

Indicates how the output is formatted.

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

A collection of [JsonConverter](#) 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: 4.0.2.C  
(4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)



C# ▾

Gets the [JToken](#) with the specified key converted to the specified type.

## 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.

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<T>
```

```
public:  
generic<typename T>  
virtual IEnumerable<T>^ Values()
```

## Generic Template Parameters

*T*

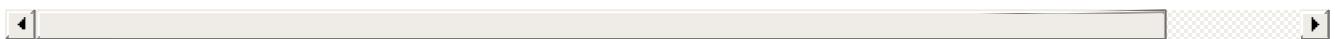
The type to convert the values to.

## Return Value

A [IEnumerable<T>](#) containing the child values of this [JToken](#), in document order.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# WriteTo Method (writer, converters)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JToken](#) ►

[WriteTo\(JsonWriter, JsonConverter\[\]\)](#)

C# ▾

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.

# JTokenEqualityComparer Class

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► **JTokenEqualityComparer**

C# ▾

Compares tokens to determine whether they are equal.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JTokenEqualityComparer : IEqualityComparer
```

```
Public Class JTokenEqualityComparer
    Implements IEqualityComparer(Of JToken)
```

```
public ref class JTokenEqualityComparer : IEqualityComparer
```

## Members

All Members

Constructors

Methods

Public

Protected

Instance

Static

Declared

Inherited

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\(\)](#)

C#

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JTokenEqualityComparer()
```

```
Public Sub New
```

```
public:  
JTokenEqualityComparer()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Equals Method (x, y)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenEqualityComparer](#) ► [Equals\(JToken, JToken\)](#)

C# ▾

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.

(4.0.2.13623)

# GetHashCode Method (obj)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenEqualityComparer](#) ► **GetHashCode(JToken)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

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"/> 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="#">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>	Gets or sets a value indicating whether	

		<p>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="#">CurrentState</a>	<p>Gets the current reader state.</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> <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="#">MemberwiseClone()</a>	<p>Creates a shallow copy of the current <a href="#">Object</a>.</p>

		(Inherited from <a href="#">Object</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="#">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="#">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="#"><u>TokenType</u></a>	Gets the type of the current Json token.
		(Inherited from <a href="#">JsonReader</a> .)
	<a href="#"><u>ToString()</u></a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .
		(Inherited from <a href="#">Object</a> .)
	<a href="#"><u>Value</u></a>	Gets the text value of the current Jsc token.
		(Inherited from <a href="#">JsonReader</a> .)
	<a href="#"><u>ValueType</u></a>	Gets The Common Language Runtir (CLR) type for the current Json toker
		(Inherited from <a href="#">JsonReader</a> .)

## Inheritance Hierarchy

```
Object
└ JsonReader
    └ JTokenReader
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

## Constructor (token)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenReader](#) ►

**JTokenReader(JToken)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# Read Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenReader](#) ► **Read()**

C# ▾

Reads the next JSON token from the stream.

## Declaration Syntax

C#

Visual Basic

Visual C++  
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: 4.0.2.C  
(4.0.2.13623)

# ReadAsBytes Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenReader](#) ►

## ReadAsBytes()

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# ReadAsDateTimeOffset

## Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenReader](#) ►

**ReadAsDateTimeOffset()**

C#

Reads the next JSON token from the stream as a [Nullable<T>](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public override Nullable<DateTimeOffset> ReadAsDateTimeOffset()
```

```
Public Overrides Function ReadAsDateTimeOffset As Nullable<DateTimeOffset>
```

```
public:  
virtual Nullable<DateTimeOffset> ReadAsDateTimeOffset();
```

### Return Value

A [Nullable<T>](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# ReadAsDecimal Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenReader](#) ►

## ReadAsDecimal()

C# ▾

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>](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

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"/> Inherit

Icon	Member	Description
≡	<a href="#">JTokenWriter(JContainer)</a>	Initializes a new instance <b>JTokenWriter</b> class writing to a <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 underlying <a href="#">Stream</a> .

		(Overrides <a href="#">JsonWriter.CloseOutput</a> )
	<a href="#">CloseOutput</a>	Gets or sets a value indicating whether the underlying stream or streams should be closed when the <a href="#">Close</a> method is called.
		(Inherited from <a href="#">JsonWriter</a> )
	<a href="#">Equals(Object)</a>	Determines whether the specified object 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 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 closes the underlying stream.
		(Overrides <a href="#">JsonWriter.Flush</a> )
	<a href="#">Formatting</a>	Indicates how the output is formatted.
		(Inherited from <a href="#">JsonWriter</a> )
	<a href="#">GetHashCode()</a>	Serves as a hash function for a <a href="#">Type</a> when combined with other hash functions.
		(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="#">Token</a>	Gets the token being written.
	<a href="#">Top</a>	Gets the top.
	<a href="#">ToString()</a>	(Inherited from <a href="#">JsonWriter</a> ) Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .
	<a href="#">WriteComment(String)</a>	(Inherited from <a href="#">Object</a> ). Writes out a comment.
		<b>C#</b> <pre>/* . . . */</pre>
	<a href="#">WriteEnd(JsonToken)</a>	containing the specified token.
		(Overrides <a href="#">JsonWriter.WriteComment</a> )
	<a href="#">WriteEnd()</a>	Writes the end.
		(Overrides <a href="#">JsonWriter.WriteEnd(JsonToken)</a> )
	<a href="#">WriteEndConstructor()</a>	Writes the end of the current array.
		(Inherited from <a href="#">JsonWriter</a> )
	<a href="#">WriteEndObject()</a>	Writes the end of an array.
		(Inherited from <a href="#">JsonWriter</a> )
		Writes the end constructor.
		(Inherited from <a href="#">JsonWriter</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.WriteLine()</a> )
💡	<a href="#">WritePropertyName(String)</a>	Writes the property name/value pair on a Json line.  (Overrides <a href="#">JsonWriter.WriteProperty()</a> )
💡	<a href="#">WriteRaw(String)</a>	Writes raw JSON.  (Overrides <a href="#">JsonWriter.WriteLine()</a> )
💡	<a href="#">WriteRawValue(String)</a>	Writes raw JSON where expected and updates the current line.  (Inherited from <a href="#">JsonWriter</a> )
💡	<a href="#">WriteStartArray()</a>	Writes the beginning of an array.  (Overrides <a href="#">JsonWriter.WriteLine()</a> )
💡	<a href="#">WriteStartConstructor(String)</a>	Writes the start of a constructor with the given name.  (Overrides <a href="#">JsonWriter.WriteStartConstructor()</a> )
💡	<a href="#">WriteStartObject()</a>	Writes the beginning of an object.  (Overrides <a href="#">JsonWriter.WriteLine()</a> )
📝	<a href="#">WriteState</a>	Gets the state of the writer.

		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteToken(JsonReader)</a>	Writes the current <a href="#">JsonReader</a> token.
		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteUndefined()</a>	Writes an undefined value.
		(Overrides <a href="#">JsonWriter.WriteLine()</a> )
≡	<a href="#">WriteValue(String)</a>	Writes a <a href="#">String</a> value.
		(Overrides <a href="#">JsonWriter.WriteLine(String)</a> )
≡	<a href="#">WriteValue(Int32)</a>	Writes a <a href="#">Int32</a> value.
		(Overrides <a href="#">JsonWriter.WriteLine(Int32)</a> )
≡	<a href="#">WriteValue(UInt32)</a>	Writes a <a href="#">UInt32</a> value.
		(Overrides <a href="#">JsonWriter.WriteLine(UInt32)</a> )
≡	<a href="#">WriteValue(Int64)</a>	Writes a <a href="#">Int64</a> value.
		(Overrides <a href="#">JsonWriter.WriteLine(Int64)</a> )
≡	<a href="#">WriteValue(UInt64)</a>	Writes a <a href="#">UInt64</a> value.
		(Overrides <a href="#">JsonWriter.WriteLine(UInt64)</a> )
≡	<a href="#">WriteValue(Single)</a>	Writes a <a href="#">Single</a> value.
		(Overrides <a href="#">JsonWriter.WriteLine(Single)</a> )
≡	<a href="#">WriteValue(Double)</a>	Writes a <a href="#">Double</a> value.
		(Overrides <a href="#">JsonWriter.WriteLine(Double)</a> )
≡	<a href="#">WriteValue(Boolean)</a>	Writes a <a href="#">Boolean</a> value.

		(Overrides <a href="#">JsonWriter.WriteLine(B)</a> )
≡	<a href="#">WriteValue(Int16)</a>	Writes a <a href="#">Int16</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(I)</a> )
≡	<a href="#">WriteValue(UInt16)</a>	Writes a <a href="#">UInt16</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(U)</a> )
≡	<a href="#">WriteValue(Char)</a>	Writes a <a href="#">Char</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(C)</a> )
≡	<a href="#">WriteValue(Byte)</a>	Writes a <a href="#">Byte</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(B)</a> )
≡	<a href="#">WriteValue(SByte)</a>	Writes a <a href="#">SByte</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(S)</a> )
≡	<a href="#">WriteValue(Decimal)</a>	Writes a <a href="#">Decimal</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(D)</a> )
≡	<a href="#">WriteValue(DateTime)</a>	Writes a <a href="#">DateTime</a> value  (Overrides <a href="#">JsonWriter.WriteLine(D)</a> )
≡	<a href="#">WriteValue(DateTimeOffset)</a>	Writes a <a href="#">DateTimeOffset</a>  (Overrides <a href="#">JsonWriter.WriteLine(D)</a> )
≡	<a href="#">WriteValue(Byte[])</a>	Writes a <a href="#">Byte[]</a> value.  (Overrides <a href="#">JsonWriter.WriteLine(B)</a> )
≡		

	<a href="#">WriteValue(Nullable&lt;Int32&gt;)</a>	Writes a <code>Nullable&lt;T&gt;</code> value.
		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;UInt32&gt;)</a>	Writes a <code>Nullable&lt;T&gt;</code> value.
		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;Int64&gt;)</a>	Writes a <code>Nullable&lt;T&gt;</code> value.
		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;UInt64&gt;)</a>	Writes a <code>Nullable&lt;T&gt;</code> value.
		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;Single&gt;)</a>	Writes a <code>Nullable&lt;T&gt;</code> value.
		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;Double&gt;)</a>	Writes a <code>Nullable&lt;T&gt;</code> value.
		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;Boolean&gt;)</a>	Writes a <code>Nullable&lt;T&gt;</code> value.
		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;Int16&gt;)</a>	Writes a <code>Nullable&lt;T&gt;</code> value.
		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;UInt16&gt;)</a>	Writes a <code>Nullable&lt;T&gt;</code> value.
		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;Char&gt;)</a>	Writes a <code>Nullable&lt;T&gt;</code> value.
		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;Byte&gt;)</a>	Writes a <code>Nullable&lt;T&gt;</code> value.
		(Inherited from <a href="#">JsonWriter</a> )
≡	<a href="#">WriteValue(Nullable&lt;SByte&gt;)</a>	Writes a <code>Nullable&lt;T&gt;</code> 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(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
    └ JTokenWriter
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



## 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: 4.0.2.0  
(4.0.2.13623)

# JTokenWriter

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► [JTokenWriter\(\)](#)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

## Constructor (container)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► [JTokenWriter\(JContainer\)](#)

C# ▾

Initializes a new instance of the [JTokenWriter](#) class writing to the given [JContainer](#).

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public JTokenWriter(  
    JContainer container  
)
```

```
Public Sub New ( _  
    container As JContainer _  
)
```

```
public:  
JTokenWriter(  
    JContainer^ container  
)
```

### Parameters

#### *container* ([JContainer](#))

The container being written to.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

# Flush Method

[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: 4.0.2.C  
(4.0.2.13623)

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# 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.

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: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

# WritePropertyName

## Method (name)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

### WritePropertyName(String)

C#

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: 4.0.2.0 (4.0.2.13623)

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteStartArray Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **WriteStartArray()**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# WriteStartConstructor

## Method (name)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

**WriteStartConstructor(String)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteStartObject Method

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ►

## WriteStartObject()

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

## 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.

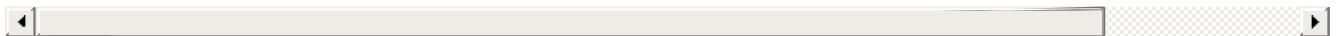
		(Overrides <a href="#">JsonWriter.WriteLine(B</a> )
≡	<a href="#">WriteValue(Int16)</a>	Writes a <a href="#">Int16</a> value.
≡	<a href="#">WriteValue(UInt16)</a>	(Overrides <a href="#">JsonWriter.WriteLine(In</a> )
≡	<a href="#">WriteValue(Char)</a>	Writes a <a href="#">UInt16</a> value.
≡	<a href="#">WriteValue(Byte)</a>	(Overrides <a href="#">JsonWriter.WriteLine(U</a> )
≡	<a href="#">WriteValue(SByte)</a>	Writes a <a href="#">Char</a> value.
≡	<a href="#">WriteValue(Decimal)</a>	(Overrides <a href="#">JsonWriter.WriteLine(C</a> )
≡	<a href="#">WriteValue(DateTime)</a>	Writes a <a href="#">Byte</a> value.
≡	<a href="#">WriteValue(DateTimeOffset)</a>	(Overrides <a href="#">JsonWriter.WriteLine(W</a> )
≡	<a href="#">WriteValue(Byte[])</a>	Writes a <a href="#">SByte</a> value.
≡		(Overrides <a href="#">JsonWriter.WriteLine(S</a> )
≡		Writes a <a href="#">Decimal</a> value.
≡		(Overrides <a href="#">JsonWriter.WriteLine(D</a> )
≡		Writes a <a href="#">DateTime</a> value.
≡		(Overrides <a href="#">JsonWriter.WriteLine(D</a> )
≡		Writes a <a href="#">DateTimeOffset</a> value.
≡		(Overrides <a href="#">JsonWriter.WriteLine(D</a> )
≡		Writes a <a href="#">Byte[]</a> value.
		(Overrides <a href="#">JsonWriter.WriteLine(B</a> )

	<a href="#">JsonWriter.WriteLine(B)</a>	
≡	<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="#"><u>WriteValue(Nullable&lt;SByte&gt;)</u></a>	Writes a <a href="#"><u>Nullable&lt;T&gt;</u></a> value.
≡	<a href="#"><u>WriteValue(Nullable&lt;Decimal&gt;)</u></a>	Writes a <a href="#"><u>Nullable&lt;T&gt;</u></a> value.
≡	<a href="#"><u>WriteValue(Nullable&lt;DateTime&gt;)</u></a>	Writes a <a href="#"><u>Nullable&lt;T&gt;</u></a> value.
≡	<a href="#"><u>WriteValue(Nullable&lt;DateTimeOffset&gt;)</u></a>	Writes a <a href="#"><u>Nullable&lt;T&gt;</u></a> value.
≡	<a href="#"><u>WriteValue(Object)</u></a>	Writes a <a href="#"><u>Object</u></a> value. An exception is raised if the value cannot be converted to a single JSON token.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
 (4.0.2.13623)



# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **WriteValue(Boolean)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **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(  
    unsigned char value  
) override
```

## Parameters

### **value (Byte)**

The [Byte](#) value to write.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **WriteValue(Char)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **WriteValue(DateTime)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **WriteValue(Int64)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **WriteValue(SByte)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **WriteValue(String)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **WriteValue(UInt32)**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteValue Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JTokenWriter](#) ► **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: 4.0.2.0 (4.0.2.13623)

Represents a value in JSON (string, integer, date, etc).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JValue : JToken, IEquatable<JValue>,
    IFormattable, IComparable, IComparable<JValue>
```

```
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

 Public Protected Instance Static

Declared

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(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.
≡ S	<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> .)
≡ S	<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 childen 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="#">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="#">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.  (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](#)

[◀]

[▶]

**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(Object)</a>	Initializes a new instance of the

[JValue](#) class with the given value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# JValue Constructor (other)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(JValue\)](#)

C#

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: 4.0.2.C  
(4.0.2.13623)

# JValue Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(Boolean\)](#)

C#

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: 4.0.2.C  
(4.0.2.13623)

# JValue Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(DateTime\)](#)

C#

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: 4.0.2.C  
(4.0.2.13623)

# JValue Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(Double\)](#)

C#

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: 4.0.2.C  
(4.0.2.13623)

# JValue Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(Int64\)](#)

C#

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: 4.0.2.C  
(4.0.2.13623)

# JValue Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(Object\)](#)

C#

Initializes a new instance of the [JValue](#) class with the given value.

## 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: 4.0.2.C  
(4.0.2.13623)

# JValue Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(String\)](#)

C#

Initializes a new instance of the [JValue](#) class with the given value.

## 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: 4.0.2.C  
(4.0.2.13623)

# JValue Constructor (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► [JValue\(UInt64\)](#)

C#

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: 4.0.2.C  
(4.0.2.13623)

# CompareTo Method (obj)

[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 <i>obj</i> .
Zero	This instance is equal to <i>obj</i> .
Greater than zero	This instance is greater than <i>obj</i> .

### 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: 4.0.2.0  
(4.0.2.13623)

# CreateComment Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► **CreateComment(String)**

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# CreateString Method (value)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► **CreateString(String)**

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# Equals Method

[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: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.0 (4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Equals Method (obj)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► Equals(Object)

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.

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: 4.0.2.C  
(4.0.2.13623)

# GetMetaObject Method (parameter)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ►

**GetMetaObject(Expression)**

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

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 Boolean  
    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: 4.0.2.0 (4.0.2.13623)

**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: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)

# 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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)

# ToString Method (format, formatProvider)

[Namespaces](#) ► [Newtonsoft.Json.Linq](#) ► [JValue](#) ► **ToString(String, IFormatProvider)**

C# ▾

Returns a [String](#) that represents this instance.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public string ToString(  
    string format,  
    IFormatProvider formatProvider  
)
```

```
Public Function ToString ( _  
    format As String, _  
    formatProvider As IFormatProvider _  
) As String
```

```
public:  
virtual String^ ToString(  
    String^ format,  
    IFormatProvider^ formatProvider  
) sealed
```

## 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: 4.0.2.C  
(4.0.2.13623)

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: 4.0.2.C  
(4.0.2.13623)

Gets or sets the underlying token value.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public Object Value { get; set; }
```

```
Public Property Value As Object
    Get
    Set
```

```
public:
property Object^ Value {
    Object^ get ();
    void set (Object^ value);
}
```

## Value

The underlying token value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# 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.

# Documentation **Newtonsoft.Json.Linq.ComponentModel**

## Namespace

[Namespaces](#) ► **Newtonsoft.Json.Linq.ComponentModel**

C# ▾

The **Newtonsoft.Json.Linq.ComponentModel** namespace provides classes for LINQ to JSON databinding.

### [-] Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
namespace Newtonsoft.Json.Linq.ComponentModel
```

```
Namespace Newtonsoft.Json.Linq.ComponentModel
```

```
namespace Newtonsoft.Json.Linq.ComponentModel
```

### [-] Types

All Types

Classes

Icon	Type	Description
	<a href="#">JPropertyDescriptor</a>	Represents a view of a <a href="#">JProperty</a> .

# JPropertyDescriptor Class

[Namespaces](#) ► [Newtonsoft.Json.Linq.ComponentModel](#) ► [JPropertyDescriptor](#)

C# ▾

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

Inherit

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 attrib

		(Inherited from <a href="#">MemberDescriptor</a> )
	<a href="#">Attributes</a>	Gets the collection of attributes associated with this member.
		(Inherited from <a href="#">MemberDescriptor</a> )
	<a href="#">CanResetValue(Object)</a>	When overridden in a derived class, indicates whether resetting an object classifies as a change.
		(Overrides <a href="#">PropertyDescriptor.CanResetValue</a> )
	<a href="#">Category</a>	Gets the name of the category that the member belongs, as specified by the <a href="#">CategoryAttribute</a> .
		(Inherited from <a href="#">MemberDescriptor</a> )
	<a href="#">ComponentType</a>	When overridden in a derived class, gets the component type of the component this property belongs to.
		(Overrides <a href="#">PropertyDescriptor.ComponentType</a> )
	<a href="#">Converter</a>	Gets the type converter for this member.
		(Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">CreateAttributeCollection()</a>	Creates a collection of attributes based on the attributes passed to the constructor.
		(Inherited from <a href="#">MemberDescriptor</a> )
	<a href="#">CreateInstance(Type)</a>	Creates an instance of the specified type.
		(Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">Description</a>	Gets the description of the member, as specified by the <a href="#">DescriptionAttribute</a> .
		(Inherited from <a href="#">MemberDescriptor</a> )
	<a href="#">DesignTimeOnly</a>	Gets whether this member should be shown at design time, as specified in the <a href="#">DesignOnlyAttribute</a> .

		(Inherited from <a href="#">MemberDescriptor</a> )
	<a href="#">DisplayName</a>	Gets the name that can be displayed such as a Properties window.
		(Inherited from <a href="#">MemberDescriptor</a> )
	<a href="#">Equals(Object)</a>	Compares this to another object for equivalence.
		(Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">FillAttributes(IList)</a>	Adds the attributes of the <a href="#">Property</a> to the specified list of attributes.
		(Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to attempt to perform other cleanup operations before the <a href="#">Object</a> is reclaimed by garbage collection.
		(Inherited from <a href="#">Object</a> )
	<a href="#">GetChildProperties()</a>	Returns the default <a href="#">PropertyDescriptor</a> .
		(Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">GetChildProperties(Attribute[])</a>	Returns a <a href="#">PropertyDescriptor</a> for the specified array of attributes.
		(Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">GetChildProperties(Object)</a>	Returns a <a href="#">PropertyDescriptor</a> for the given object.
		(Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">GetChildProperties(Object, Attribute[])</a>	Returns a <a href="#">PropertyDescriptor</a> for the given object using a specified array of attributes as a filter.
		(Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">GetEditor(Type)</a>	Gets an editor of the specified type.

		(Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">GetHashCode()</a>	Returns the hash code for this <a href="#">Object</a> .
		(Inherited from <a href="#">PropertyDescriptor</a> )
	<a href="#">GetInvocationTarget(Type, Object)</a>	This method returns the object used during invocation of methods.
		(Inherited from <a href="#">PropertyDescriptor</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="#">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, gets the current value of the property.
		(Overrides <a href="#">PropertyDescriptor</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 this <a href="#">Object</a> 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 this <a href="#">Object</a> 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, indicates whether this property is read-only.
		(Overrides <a href="#">PropertyDescriptor</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. This method is virtual and can be overridden. This method is implemented by the base class.
		Raises the ValueChanged event. This method is virtual and can be overridden. (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 value for this property of the component to its 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 the <a href="#">DesignerSerializationVisibility</a> attribute.  (Inherited from <a href="#">PropertyDescriptor</a> .)
	<a href="#">SetValue(Object, Object)</a>	When overridden in a derived class, sets the value of the component to a value.  (Overrides <a href="#">PropertyDescriptor.SetValue(Object, Object)</a> .)

	<a href="#">Object</a> ).)
≡	<p><a href="#">ShouldSerializeValue(Object)</a></p> <p>When overridden in a derived class, returns a value indicating whether the current <code>Object</code> needs to be persisted.</p> <p>(Overrides <a href="#">PropertyDescriptor.ShouldSerializeValue(Object)</a>.)</p>
⊕	<p><a href="#">SupportsChangeEvents</a></p> <p>Gets a value indicating whether notifications for this property are handled outside the property descriptor.</p> <p>(Inherited from <a href="#">PropertyDescriptor</a>.)</p>
≡	<p><a href="#">ToString()</a></p> <p>Returns a <a href="#">String</a> that represents this <code>Object</code>.</p> <p>(Inherited from <a href="#">Object</a>.)</p>

## [-] Inheritance Hierarchy

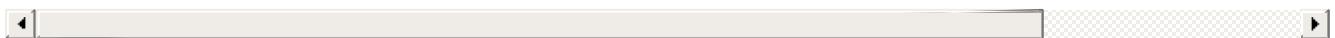
```

Object
└ MemberDescriptor
  └ PropertyDescriptor
    └ JPropertyDescriptor

```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# JPropertyDescriptor

## Constructor (name, propertyType)

[Namespaces](#) ► [Newtonsoft.Json.Linq.ComponentModel](#) ► [JPropertyDescriptor](#) ► [JPropertyDescriptor\(String, Type\)](#)

C# ▾

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

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JPropertyDescriptor(  
    string name,  
    Type propertyType  
)
```

```
Public Sub New ( _  
    name As String, _  
    propertyType As Type _  
)
```

```
public:  
JPropertyDescriptor(  
    String^ name,  
    Type^ propertyType  
)
```

### Parameters

#### *name* ([String](#))

The name.

#### *propertyType* ([Type](#))

Type of the property.

# CanResetValue Method (component)

[Namespaces](#) ► [Newtonsoft.Json.Linq.ComponentModel](#) ►  
[JPropertyDescriptor](#) ► **CanResetValue(Object)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# ComponentType Property

[Namespaces](#) ► [Newtonsoft.Json.Linq.ComponentModel](#) ►

[JPropertyDescriptor](#) ► **ComponentType**

C# ▾

When overridden in a derived class, gets the type of the component this property is bound to.

## 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: 4.0.2.0  
(4.0.2.13623)



# GetValue Method (component)

[Namespaces](#) ► [Newtonsoft.Json.Linq.ComponentModel](#) ►  
[JPropertyDescriptor](#) ► **GetValue(Object)**

C# ▾

When overridden in a derived class, gets the current value of the property on a component.

## 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.

# IsReadOnly Property

[Namespaces](#) ► [Newtonsoft.Json.Linq.ComponentModel](#) ►

[JPropertyDescriptor](#) ► **IsReadOnly**

C#

When overridden in a derived class, gets a value indicating whether this property is read-only.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public override bool IsReadOnly { get; }
```

```
Public Overrides ReadOnly Property IsReadOnly As Boo  
    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: 4.0.2.0  
(4.0.2.13623)





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 As Integer  
    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: 4.0.2.C  
(4.0.2.13623)



C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# ResetValue Method (component)

[Namespaces](#) ► [Newtonsoft.Json.Linq.ComponentModel](#) ►  
[JPropertyDescriptor](#) ► **ResetValue(Object)**

C# ▾

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++ ▾

```
public override void ResetValue(  
    Object component  
)
```

```
Public Overrides Sub ResetValue ( _  
    component As Object _  
)
```

```
public:  
virtual void ResetValue(  
    Object^ component  
) override
```

## Parameters

### *component* ([Object](#))

The component with the property value that is to be reset to the default value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# SetValue Method (component, value)

[Namespaces](#) ► [Newtonsoft.Json.Linq.ComponentModel](#) ►  
[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.

# ShouldSerializeValue

## Method (component)

[Namespaces](#) ► [Newtonsoft.Json.Linq.ComponentModel](#) ►

[JPropertyDescriptor](#) ► **ShouldSerializeValue(Object)**

C#

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.

# Newtonsoft.Json.Schema 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++ ▾

```
namespace Newtonsoft.Json.Schema
```

```
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="#"><b>JsonSchemaResolver</b></a>	Resolves <a href="#">JsonSchema</a> from an id.
	<a href="#"><b>JsonSchemaType</b></a>	The value types allowed by the <a href="#">JsonSchema</a> .
	<a href="#"><b>UndefinedSchemaIdHandling</b></a>	Specifies undefined schema Id handling options for the <a href="#">JsonSchemaGenerator</a> .
	<a href="#"><b>ValidationEventArgs</b></a>	Returns detailed information related to the <a href="#">ValidationEventHandler</a> .
	<a href="#"><b>ValidationEventHandler</b></a>	Represents the callback method that will handle JSON schema validation events and the <a href="#">ValidationEventArgs</a> .

C# ▾

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 Instance Declared Protected Static Inherited

Icon	Member	Description
	<a href="#">IsValid(JToken, JsonSchema)</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, JsonSchema, ValidationEventHandler)</a>	Validates the specified <a href="#">JToken</a> .

## Inheritance Hierarchy

[Object](#)

## └ **Extensions**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# 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++

```
public static bool IsValid(
    this JToken source,
    JsonSchema schema
)
```

```
<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: 4.0.2.C  
(4.0.2.13623)

## 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: 4.0.2.C  
(4.0.2.13623)

# Validate Method (source, schema)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [Extensions](#) ► **Validate(JToken, JsonSchema)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# 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 validationEventHandler
)
```

```
<ExtensionAttribute> _
Public Shared Sub Validate ( _
    source As JToken, _
    schema As JsonSchema, _
    validationEventHandler As ValidationEventHandler
)
```

```
[ExtensionAttribute]
public:
static void Validate(
    JToken^ source,
    JsonSchema^ schema,
    ValidationEventHandler^ validationEventHandler
)
```

## Parameters

### source ([JToken](#))

The source [JToken](#) to test.

### schema ([JsonSchema](#))

The schema to test with.

### ***validationEventHandler* ([ValidationEventHandler](#))**

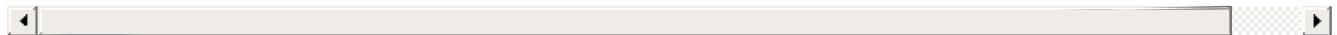
The validation event handler.

#### **[-] 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: 4.0.2.C  
(4.0.2.13623)



# JsonSchema Class

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► **JsonSchema**

C# ▾

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

 Public Protected Instance Static

Declared

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="#"><u>Default</u></a>	Gets or sets the default value.
	<a href="#"><u>Description</u></a>	Gets or sets the description of the object.
	<a href="#"><u>Disallow</u></a>	Gets or sets disallowed types.
	<a href="#"><u>DivisibleBy</u></a>	Gets or sets a number that the value should be divisible by.
	<a href="#"><u>Enum</u></a>	Gets or sets the a collection of valid enum values allowed.
	<a href="#"><u>Equals(Object)</u></a>	Determines whether the specified <a href="#"><u>Object</u></a> is equal to the current <a href="#"><u>Object</u></a> (Inherited from <a href="#"><u>Object</u></a> .)
	<a href="#"><u>ExclusiveMaximum</u></a>	Gets or sets a flag indicating whether the value can not equal the number defined by the "maximum" attribute.
	<a href="#"><u>ExclusiveMinimum</u></a>	Gets or sets a flag indicating whether the value can not equal the number defined by the "minimum" attribute.
	<a href="#"><u>Extends</u></a>	Gets or sets the extend <a href="#"><u>JsonSchema</u></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="#">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="#"><u>Properties</u></a>	Gets or sets the <b>JsonSchema</b> of properties.
	<a href="#"><u>Read(JsonReader)</u></a>	Reads a <b>JsonSchema</b> from the specified <a href="#">JsonReader</a> .
	<a href="#"><u>Read(JsonReader, JsonSchemaResolver)</u></a>	Reads a <b>JsonSchema</b> from the specified <a href="#">JsonReader</a> .
	<a href="#"><u>ReadOnly</u></a>	Gets or sets whether the object is read only.
	<a href="#"><u>Required</u></a>	Gets or sets whether the object is required.
	<a href="#"><u>Requires</u></a>	Gets or sets the required property if this property is present.
	<a href="#"><u>Title</u></a>	Gets or sets the title.
	<a href="#"><u>ToString()</u></a>	Returns a <a href="#">String</a> that represents the current <a href="#">Object</a> .  (Overrides <a href="#">Object.ToString()</a> .)
	<a href="#"><u>Transient</u></a>	Gets or sets whether the object is transient.
	<a href="#"><u>Type</u></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: 4.0.2.C  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# JsonSchema Constructor

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ► [JsonSchema\(\)](#)

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# 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: 4.0.2.0 (4.0.2.13623)

# Documentation **AllowAdditionalProperties Property**

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

## 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: 4.0.2.0  
(4.0.2.13623)

C# ▾

Gets or sets the default value.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JToken Default { get; set; }
```

```
Public Property Default As JToken
    Get
    Set
```

```
public:
property JToken^ Default {
    JToken^ get ();
    void set (JToken^ value);
}
```

## Value

The default value.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# Description Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ► **Description**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

C# ▾

Gets or sets a number that the value should be divisible by.

## 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: 4.0.2.0  
(4.0.2.13623)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# ExclusiveMaximum

## Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

### ExclusiveMaximum

C# ▾

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 Boolean)
    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: 4.0.2.0  
(4.0.2.13623)

# ExclusiveMinimum

## Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

### ExclusiveMinimum

C# ▾

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\(of Boolean\)
    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: 4.0.2.0  
(4.0.2.13623)

# Extends Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ► **Extends**

C# ▾

Gets or sets the extend [JsonSchema](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchema Extends { get; set; }
```

```
Public Property Extends As JsonSchema
    Get
    Set
```

```
public:
property JsonSchema^ Extends {
    JsonSchema^ get ();
    void set (JsonSchema^ value);
}
```

## Value

The extended [JsonSchema](#).

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



Gets or sets the format.

## Declaration Syntax

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

```
Public Property Format As String
    Get
    Set
```

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

## Value

The format.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

Gets or sets whether the object is visible to users.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<bool> Hidden { get; set; }
```

```
Public Property Hidden As Nullable\(Of Boolean\)
    Get
    Set
```

```
public:
property Nullable<bool> Hidden {
    Nullable<bool> get ();
    void set (Nullable<bool> value);
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

C#  
▼

Gets or sets the identity.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public IList<string> Identity { get; set; }
```

```
Public Property Identity As IList\(of String\)
    Get
    Set
```

```
public:
property IList<String^>^ Identity {
    IList<String^>^ get ();
    void set (IList<String^>^ value);
}
```

## Value

The identity.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# Items Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ► **Items**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# Maximum Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ► **Maximum**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# MaximumLength

## Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ►

### MaximumLength

C# ▾

Gets or sets the maximum length.

#### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<int> MaximumLength { get; set; }
```

```
Public Property MaximumLength As Nullable(Of Integer)
    Get
    Set
```

```
public:
property Nullable<int> MaximumLength {
    Nullable<int> get ();
    void set (Nullable<int> value);
}
```

#### Value

The maximum length.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

▾

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: 4.0.2.0  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# MinimumLength Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ► [MinimumLength](#)

Gets or sets the minimum length.

C# ▾

## 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: 4.0.2.0  
(4.0.2.13623)

C# ▾

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
```

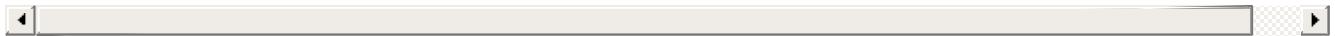
```
public:  
property IDictionary<JToken^, String^>^ Options {  
    IDictionary<JToken^, String^>^ get ();  
    void set (IDictionary<JToken^, String^>^ val  
}
```

## Value

A collection of options.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



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: 4.0.2.0 (4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

# 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: 4.0.2.C  
(4.0.2.13623)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# PatternProperties

## Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ► [PatternProperties](#)

C# ▾

Gets or sets the pattern properties.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public IDictionary<string, JsonSchema> PatternProper
```

```
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: 4.0.2.0 (4.0.2.13623)



C# ▾

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^>^ Properties {
    IDictionary<String^, JsonSchema^>^ get ();
    void set (IDictionary<String^, JsonSchema^>^
}
```

## Value

The [JsonSchema](#) of properties.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# Read Method

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ► **Read()**

C# ▾

## ‑ Members

Icon	Member	Description
= S	<a href="#">Read(JsonReader)</a>	Reads a <a href="#">JsonSchema</a> from the specified <a href="#">JsonReader</a> .
= S	<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: 4.0.2.C  
(4.0.2.13623)

▾

Reads a [JsonSchema](#) from the specified [JsonReader](#).

## Declaration Syntax

**C#****Visual Basic****Visual C++**

```
public static JsonSchema Read(  
    JsonReader reader  
)
```

```
Public Shared Function Read ( _  
    reader As JsonReader _  
) As JsonSchema
```

```
public:  
static JsonSchema^ Read(  
    JsonReader^ reader  
)
```

## 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: 4.0.2.C  
(4.0.2.13623)

# Read Method (reader, resolver)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ► **Read(JsonReader, JsonSchemaResolver)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# ReadOnly Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ► **ReadOnly**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# Required Property

[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: 4.0.2.0  
(4.0.2.13623)

# Requires Property

[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: 4.0.2.0  
(4.0.2.13623)



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: 4.0.2.0  
(4.0.2.13623)

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# Transient Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ► [Transient](#)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

C# ▾

Gets or sets the types of values allowed by the object.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<JsonSchemaType> Type { get; set; }
```

```
Public Property Type As Nullable\(Of JsonSchemaType\)
    Get
    Set
```

```
public:
property Nullable<JsonSchemaType> Type {
    Nullable<JsonSchemaType> get ();
    void set (Nullable<JsonSchemaType> value);
}
```

## Value

The type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

## WriteTo Method

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ► **WriteTo()**

C# ▾

### 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: 4.0.2.0 (4.0.2.13623)

## WriteTo(JsonWriter)

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# WriteTo Method (writer, resolver)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchema](#) ► **WriteTo(JsonWriter, JsonSchemaResolver)**

C# ▾

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.

# JsonSchemaException

## Class

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► **JsonSchemaException**

C# ▾

Returns detailed information about the schema exception.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonSchemaException : Exception
```

```
Public Class JsonSchemaException _
    Inherits Exception
```

```
public ref class JsonSchemaException : public Except
```

### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declared

Inherited

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>JsonSchemaException</b> class with a specified error message.
≡	<a href="#">JsonSchemaException(String, Exception)</a>	Initializes a new instance of the <b>JsonSchemaException</b> class

		with a specified error message and a reference to the inner exception that is the cause of this exception.
	<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="#">LineNumber</a>	Gets the line number indicating where the error occurred.
	<a href="#">LinePosition</a>	Gets the line position indicating where the error occurred.
	<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
  └ JsonSchemaException
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonSchemaException 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

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonSchemaException Constructor

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaException](#) ► **JsonSchemaException()**

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

C# ▾

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchemaException()
```

```
Public Sub New
```

```
public:  
JsonSchemaException()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonSchemaException

## Constructor (message)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaException](#) ► **JsonSchemaException(String)**

Initializes a new instance of the [JsonSchemaException](#) class with a specified error message.

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchemaException(  
    string message  
)
```

```
Public Sub New ( _  
    message As String _  
)
```

```
public:  
JsonSchemaException(  
    String^ message  
)
```

### Parameters

#### **message (String)**

The error message that explains the reason for the exception.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# 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.

C# ▾

### 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: 4.0.2.C  
(4.0.2.13623)

# LineNumber Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaException](#) ► **LineNumber**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# LinePosition Property

[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: 4.0.2.0  
(4.0.2.13623)

# JsonSchemaGenerator Class

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► **JsonSchemaGenerator**

C# ▾

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



Declared



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="#"><u>UndefinedSchemaIdHandling</u></a>	Gets or sets how undefined schemas are handled by the serializer.

## Inheritance Hierarchy

[Object](#)

 [\*\*JsonSchemaGenerator\*\*](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# JsonSchemaGenerator Constructor

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaGenerator](#) ► **JsonSchemaGenerator()**

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

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchemaGenerator()
```

```
Public Sub New
```

```
public:  
JsonSchemaGenerator()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaGenerator](#) ► **ContractResolver**

C# ▾

Gets or sets the contract resolver.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public IContractResolver ContractResolver { get; set }
```

```
Public Property ContractResolver As IContractResolver
    Get
    Set
```

```
public:
property IContractResolver^ ContractResolver {
    IContractResolver^ get ();
    void set (IContractResolver^ value);
}
```

### Value

The contract resolver.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# Generate Method

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaGenerator](#) ► **Generate()**

C# ▾

## 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: 4.0.2.C  
(4.0.2.13623)

C# ▾

Generate a [JsonSchema](#) from the specified type.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchema Generate(  
    Type type  
)
```

```
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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# Generate Method (type, resolver)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaGenerator](#) ► **Generate(Type, JsonSchemaResolver)**

C# ▾

Generate a [JsonSchema](#) from the specified type.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchema Generate(  
    Type type,  
    JsonSchemaResolver resolver  
)
```

```
Public Function Generate ( _  
    type As Type, _  
    resolver As JsonSchemaResolver _  
) As JsonSchema
```

```
public:  
JsonSchema^ Generate(  
    Type^ type,  
    JsonSchemaResolver^ resolver  
)
```

## 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: 4.0.2.C  
(4.0.2.13623)

# Generate Method (type, resolver, rootSchemaNullable)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaGenerator](#) ► [Generate\(Type, JsonSchemaResolver, Boolean\)](#)

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# Generate Method (type, rootSchemaNullable)

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaGenerator](#) ► **Generate(Type, Boolean)**

C# ▾

Generate a [JsonSchema](#) from the specified type.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchema Generate(
    Type type,
    bool rootSchemaNullable
)
```

```
Public Function Generate ( _
    type As Type,
    rootSchemaNullable As Boolean _
) As JsonSchema
```

```
public:
JsonSchema^ Generate(
    Type^ type,
    bool rootSchemaNullable
)
```

## 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: 4.0.2.C  
(4.0.2.13623)

C# ▾

Gets or sets how undefined schemas are handled by the serializer.

## Declaration Syntax

C#

Visual Basic

Visual C++

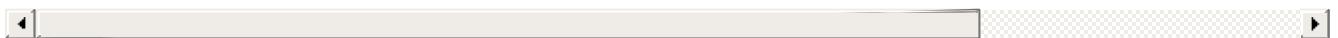
```
public UndefinedSchemaIdHandling UndefinedSchemaIdHa
```

```
Public Property UndefinedSchemaIdHandling As Undefin
    Get
    Set
```

```
public:
property UndefinedSchemaIdHandling UndefinedSchemaId
    UndefinedSchemaIdHandling get ();
    void set (UndefinedSchemaIdHandling value);
}
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# JsonSchemaResolver 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

Declared

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: 4.0.2.0 (4.0.2.13623)

# JsonSchemaResolver Constructor

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaResolver](#) ► **JsonSchemaResolver()**

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonSchemaResolver()
```

```
Public Sub New
```

```
public:  
JsonSchemaResolver()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# LoadedSchemas

## Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [JsonSchemaResolver](#) ► **LoadedSchemas**

C# ▾

Gets or sets the loaded schemas.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public IList<JsonSchema> LoadedSchemas { get; protected set; }
```

```
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: 4.0.2.0 (4.0.2.13623)



# JsonSchemaType Enumeration

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► **JsonSchemaType**

C# ▾

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
<b>None</b>	No type specified.
<b>String</b>	String type.
<b>Float</b>	Float type.
<b>Integer</b>	Integer type.
<b>Boolean</b>	Boolean type.
<b>Object</b>	Object type.
<b>Array</b>	Array type.
<b>Null</b>	Null type.
<b>Any</b>	Any type.

(4.0.2.13623)

# Documentation **UndefinedSchemaIdHandling**

## Enumeration

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

### **UndefinedSchemaIdHandling**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# ValidationEventArgs

## 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

Declared

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 event.
	<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 event
≡	<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**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# Exception Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [ValidationEventArgs](#) ► **Exception**

C# ▾

Gets the [JsonSchemaException](#) associated with the validation event.

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public JsonSchemaException Exception { get; }
```

```
Public ReadOnly Property Exception As JsonSchemaException
    Get
```

```
public:
property JsonSchemaException^ Exception {
    JsonSchemaException^ get ();
}
```

## Value

The [JsonSchemaException](#) associated with the validation event.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# Message Property

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [ValidationEventArgs](#) ► **Message**

C# ▾

Gets the text description corresponding to the validation event.

## 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: 4.0.2.0  
(4.0.2.13623)

## Delegate

[Namespaces](#) ► [Newtonsoft.Json.Schema](#) ► [ValidationEventHandler](#)

C#

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: 4.0.2.0  
(4.0.2.13623)

# Documentation **Newtonsoft.Json.Serialization**

## Namespace

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

C# ▾

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



[CamelCasePropertyNamesContractResolver](#)

Resolves member mappings for a type, camel casing property names.



[DefaultContractResolver](#)

Used by [JsonSerializer](#) to resolves a [JsonContract](#) for a given [Type](#).

	<a href="#"><b>DefaultSerializationBinder</b></a>	The default serialization binder used when resolving and loading classes from type names.
	<a href="#"><b>DynamicValueProvider</b></a>	Get and set values for a <a href="#"><b>MemberInfo</b></a> using dynamic methods.
	<a href="#"><b>ErrorContext</b></a>	Provides information surrounding an error.
	<a href="#"><b>ErrorEventArgs</b></a>	Provides data for the Error event.
	<a href="#"><b>IContractResolver</b></a>	Used by <a href="#"><b>JsonSerializer</b></a> to resolves a <a href="#"><b>JsonContract</b></a> for a given <a href="#"><b>Type</b></a> .
	<a href="#"><b>IReferenceResolver</b></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>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>JsonISerializableContract</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.
	<a href="#"><b>JsonPropertyCollection</b></a>	A collection of <a href="#">JsonProperty</a> objects.
	<a href="#"><b>JsonStringContract</b></a>	Contract details for a <a href="#">Type</a> used by the <a href="#">JsonSerializer</a> .

	<a href="#"><b>ObjectConstructor&lt;T&gt;</b></a>	Represents a method that constructs an object.
	<a href="#"><b>OnErrorAttribute</b></a>	When applied to a method, specifies that the method is called when an error occurs serializing an object.
	<a href="#"><b>ReflectionValueProvider</b></a>	Get and set values for a <a href="#"><b>MethodInfo</b></a> using reflection.

# Documentation **CamelCasePropertyNamesContractResolver**

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► **CamelCasePropertyNamesContractResolver**

C# ▾

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 CamelCasePropertyNamesContractResolver
```

### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declared

Inherited

Icon	Member	Description
💡	<a href="#">CamelCasePropertyNamesContractResolver()</a>	Initializes a new instance of the <b>CamelCasePropertyNamesContractResolver</b> class.
💡	<a href="#">CreateArrayContract(Type)</a>	Creates a <a href="#">JsonArrayContract</a> for the specified type. (Inherited from <a href="#">DefaultContractResolver</a> )
💡	<a href="#">CreateContract(Type)</a>	Determines which contract to use for the specified type.

		(Inherited from <a href="#">DictionaryContract</a> )
💡	<a href="#">CreateDictionaryContract(Type)</a>	Creates a <a href="#">JsonDictionaryContract</a>
		(Inherited from <a href="#">DictionaryContract</a> )
💡	<a href="#">CreateDynamicContract(Type)</a>	Creates a <a href="#">JsonDynamicContract</a>
		(Inherited from <a href="#">DictionaryContract</a> )
💡	<a href="#">CreateSerializableContract(Type)</a>	Creates a <a href="#">JsonSerializableContract</a>
		(Inherited from <a href="#">DictionaryContract</a> )
💡	<a href="#">CreateLinqContract(Type)</a>	Creates a <a href="#">JsonLinqContract</a>
		(Inherited from <a href="#">DictionaryContract</a> )
💡	<a href="#">CreateMemberValueProvider(MemberInfo)</a>	Creates the <a href="#">IValueProvider</a> and set values from <a href="#">MemberInfo</a>
		(Inherited from <a href="#">DictionaryContract</a> )
💡	<a href="#">CreateObjectContract(Type)</a>	Creates a <a href="#">JsonObjectContract</a>
		(Inherited from <a href="#">DictionaryContract</a> )
💡	<a href="#">CreatePrimitiveContract(Type)</a>	Creates a <a href="#">JsonPrimitiveContract</a>
		(Inherited from <a href="#">DictionaryContract</a> )
💡	<a href="#">CreateProperties(Type, MemberSerialization)</a>	Creates properties
		(Inherited from <a href="#">DictionaryContract</a> )
💡	<a href="#">CreateProperty(MemberInfo, MemberSerialization)</a>	Creates a <a href="#">JsonProperty</a>
		(Inherited from <a href="#">DictionaryContract</a> )
💡	<a href="#">CreateStringContract(Type)</a>	Creates a <a href="#">JsonStringContract</a>
		(Inherited from <a href="#">DictionaryContract</a> )
📋	<a href="#">DefaultMembersSearchFlags</a>	Gets or sets the <a href="#">DefaultMembersSearchFlags</a>
		(Inherited from <a href="#">DictionaryContract</a> )

	<a href="#">DynamicCodeGeneration</a>	Gets a value indicating whether the current object can be serialized and set using dynamic code generation, based on the type information determined by the runtime.
		(Inherited from <a href="#">Object</a> )
	<a href="#">Equals(Object)</a>	Determines whether the current <a href="#">Object</a> is equal to another object of the same type.
		(Inherited from <a href="#">Object</a> )
	<a href="#">Finalize()</a>	Allows an <a href="#">Object</a> to perform other cleaning up before it is reclaimed by garbage collection.
		(Inherited from <a href="#">Object</a> )
	<a href="#">GetHashCode()</a>	Serves as a hash function for this <a href="#">Object</a> .
		(Inherited from <a href="#">Object</a> )
	<a href="#">GetSerializableMembers(Type)</a>	Gets the serializable members of the current <a href="#">Object</a> .
		(Inherited from <a href="#">DynamicObject</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="#">ResolveContract(Type)</a>	Resolves the contract for the specified <a href="#">Type</a> .
		(Inherited from <a href="#">DynamicObject</a> )
	<a href="#">ResolveContractConverter(Type)</a>	Resolves the default converter for the specified <a href="#">Type</a> .
		(Inherited from <a href="#">DynamicObject</a> )
	<a href="#">ResolvePropertyName(String)</a>	Resolves the name of the property.
		(Overrides <a href="#">Object.ResolvePropertyName(String)</a> )

		<a href="#">DefaultContractR</a>
	<a href="#">SerializeCompilerGeneratedMembers</a>	Gets or sets a value indicating whether compiler-generated members should be serialized.
	<a href="#">ToString()</a>	Returns a <a href="#">String</a> representation of this object.

## Inheritance Hierarchy

[Object](#)

  └ [DefaultContractResolver](#)

    └ **CamelCasePropertyNamesContractResolver**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# Documentation **CamelCasePropertyNamesContractResolver**

## **Constructor**

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[CamelCasePropertyNamesContractResolver](#) ►  
**CamelCasePropertyNamesContractResolver()**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# ResolvePropertyName

## Method (propertyName)

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

[CamelCasePropertyNamesContractResolver](#) ►

**ResolvePropertyName(String)**

C# ▾

Resolves the name of the property.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected override string ResolvePropertyName(
    string propertyName
)
```

```
Protected Overrides Function ResolvePropertyName ( _
    propertyName As String _
) As String
```

```
protected:
virtual String^ ResolvePropertyName(
    String^ propertyName
) override
```

### Parameters

**propertyName (String)**

Name of the property.

### Return Value

The property name camel cased.

# DefaultContractResolver

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DefaultContractResolver](#)

Used by [JsonSerializer](#) to resolves a [JsonContract](#) for a given [Type](#).

C#

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class DefaultContractResolver : IContractResolver
```

```
Public Class DefaultContractResolver _
    Implements IContractResolver
```

```
public ref class DefaultContractResolver : IContractResolver
```

### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherit

Icon	Member	Description
	<a href="#">DefaultContractResolver()</a>	Initializes a new instance of the <b>DefaultContractResolver</b> class.
	<a href="#">DefaultContractResolver(Boolean)</a>	Initializes a new instance of the <b>DefaultContractResolver</b> class.

💡	<a href="#">CreateArrayContract(Type)</a>	Creates a <a href="#">JsonArrayContract</a> for given type.
💡	<a href="#">CreateContract(Type)</a>	Determines which contract type is created for the given type.
💡	<a href="#">CreateDictionaryContract(Type)</a>	Creates a <a href="#">JsonDictionaryContract</a> for the given type.
💡	<a href="#">CreateDynamicContract(Type)</a>	Creates a <a href="#">JsonDynamicContract</a> for the given type.
💡	<a href="#">CreateSerializableContract(Type)</a>	Creates a <a href="#">JsonSerializableContract</a> for the given type.
💡	<a href="#">CreateLinqContract(Type)</a>	Creates a <a href="#">JsonLinqContract</a> for given type.
💡	<a href="#">CreateMemberValueProvider(MemberInfo)</a>	Creates the <a href="#">IValueProvider</a> used by the serializer to get a set values from a member.

	 <a href="#">CreateObjectContract(Type)</a>	Creates a <a href="#">JsonObjectContract</a> for the given type.
	 <a href="#">CreatePrimitiveContract(Type)</a>	Creates a <a href="#">JsonPrimitiveContract</a> for the given type.
	 <a href="#">CreateProperties(Type, MemberSerialization)</a>	Creates properties for the given <a href="#">JsonContract</a> .
	 <a href="#">CreateProperty(MemberInfo, MemberSerialization)</a>	Creates a <a href="#">JsonProperty</a> for the given <a href="#">MemberInfo</a> .
	 <a href="#">CreateStringContract(Type)</a>	Creates a <a href="#">JsonStringContract</a> for the given type.
	<a href="#">DefaultMembersSearchFlags</a>	Gets or sets the default members search flag.
	<a href="#">DynamicCodeGeneration</a>	Gets a value indicating whether members are being get and set using dynamic code generation. This value is determined by the runtime permissions available.

≡	<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="#">GetSerializableMembers(Type)</a>	Gets the serializable members for the type.
≡	<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="#">ResolveContract(Type)</a>	Resolves the contract for a given type.
💡	<a href="#">ResolveContractConverter(Type)</a>	Resolves the default <a href="#">JsonConverter</a> for the type.

		contract.
💡	<a href="#">ResolvePropertyName(String)</a>	Resolves the name of a property.
📝	<a href="#">SerializeCompilerGeneratedMembers</a>	Gets or sets a value indicating whether compiler generated members should be serialized.
🌐	<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](#)

└ **DefaultContractResolver**

  └ [CamelCasePropertyNamesContractResolver](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# DefaultContractResolver

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[DefaultContractResolver](#) ► **DefaultContractResolver()**

C# ▾

### 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: 4.0.2.C  
(4.0.2.13623)

# DefaultContractResolver Constructor

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[DefaultContractResolver](#) ► **DefaultContractResolver()**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# DefaultContractResolver

## Constructor (shareCache)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[DefaultContractResolver](#) ► **DefaultContractResolver(Boolean)**

C#

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)**

C# ▾

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](#))

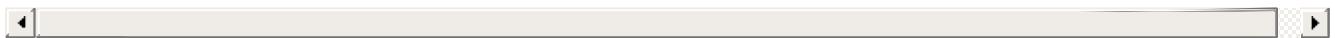
Type of the object.

### Return Value

A [JsonArrayContract](#) for the given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# CreateContract Method (*objectType*)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[DefaultContractResolver](#) ► **CreateContract(Type)**

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# CreateDictionaryContract

## Method (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[DefaultContractResolver](#) ► **CreateDictionaryContract(Type)**

C# ▾

Creates a [JsonDictionaryContract](#) for the given type.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual JsonDictionaryContract CreateDictionaryContract(  
    Type objectType  
)
```

```
Protected Overridable Function CreateDictionaryContract(  
    objectType As Type  
) As JsonDictionaryContract
```

```
protected:  
virtual JsonDictionaryContract^ CreateDictionaryContract(  
    Type^ objectType  
)
```

### Parameters

#### objectType ([Type](#))

Type of the object.

### Return Value

A [JsonDictionaryContract](#) for the given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# CreateDynamicContract

## Method (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[DefaultContractResolver](#) ► **CreateDynamicContract(Type)**

C# ▾

Creates a [JsonDynamicContract](#) for the given type.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual JsonDynamicContract CreateDynamicC
    Type objectType
)
```

```
Protected Overridable Function CreateDynamicContract
    objectType As Type _
) As JsonDynamicContract
```

```
protected:
virtual JsonDynamicContract^ CreateDynamicContract(
    Type^ objectType
)
```

### Parameters

#### *objectType* ([Type](#))

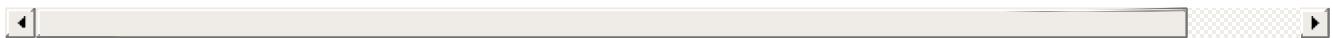
Type of the object.

### Return Value

A [JsonDynamicContract](#) for the given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# Documentation **CreateSerializableContract** Method (**objectType**)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[DefaultContractResolver](#) ► **CreateSerializableContract(Type)**

C# ▾

Creates a [JsonSerializableContract](#) for the given type.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual JsonSerializableContract CreateIS
    Type objectType
)
```

```
Protected Overridable Function CreateSerializableCo
    objectType As Type _
) As JsonSerializableContract
```

```
protected:
virtual JsonSerializableContract^ CreateSerializable
    Type^ objectType
)
```

## Parameters

### **objectType** ([Type](#))

Type of the object.

## Return Value

A [JsonSerializableContract](#) for the given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# CreateLinqContract

## Method (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[DefaultContractResolver](#) ► **CreateLinqContract(Type)**

C# ▾

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

## 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++ ▾

```
protected virtual IValueProvider CreateMemberValuePr
    MemberInfo member
)
```

```
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.

(4.0.2.13623)



# 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: 4.0.2.0 (4.0.2.13623)



# CreatePrimitiveContract

## Method (objectType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[DefaultContractResolver](#) ► **CreatePrimitiveContract(Type)**

C# ▾

Creates a [JsonPrimitiveContract](#) for the given type.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual JsonPrimitiveContract CreatePrimitiveContract(  
    Type objectType  
)
```

```
Protected Overridable Function CreatePrimitiveContract(  
    objectType As Type) As JsonPrimitiveContract
```

```
protected:  
virtual JsonPrimitiveContract^ CreatePrimitiveContract(  
    Type^ objectType  
)
```

### Parameters

#### objectType ([Type](#))

Type of the object.

### Return Value

A [JsonPrimitiveContract](#) for the given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# CreateProperties Method (**type**, **memberSerialization**)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DefaultContractResolver](#) ► **CreateProperties(Type, MemberSerialization)**

C# ▾

Creates properties for the given [JsonContract](#).

## Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual IList<JsonProperty> CreateProperties(
    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: 4.0.2.C  
(4.0.2.13623)

# CreateProperty Method (member, memberSerialization)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[DefaultContractResolver](#) ► **CreateProperty(MemberInfo, MemberSerialization)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# 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: 4.0.2.0 (4.0.2.13623)



# Documentation **DefaultMembersSearchFlags Property**

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

[DefaultContractResolver](#) ► **DefaultMembersSearchFlags**

C# ▾

Gets or sets the default members search flags.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public BindingFlags DefaultMembersSearchFlags { get;
```

```
Public Property DefaultMembersSearchFlags As BindingFlags
    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: 4.0.2.C  
(4.0.2.13623)



## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[DefaultContractResolver](#) ► **DynamicCodeGeneration**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)



## Method (objectType)

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

[DefaultContractResolver](#) ► **GetSerializableMembers(Type)**

C# ▾

Gets the serializable members for the type.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual List<MemberInfo> GetSerializableMe  
    Type objectType  
)
```

```
Protected Overridable Function GetSerializableMember  
    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: 4.0.2.0 (4.0.2.13623)



# ResolveContract Method (*type*)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DefaultContractResolver](#) ► **ResolveContract(*Type*)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# Documentation **ResolveContractConverter** Method (**objectType**)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[DefaultContractResolver](#) ► **ResolveContractConverter(Type)**

C# ▾

Resolves the default [JsonConverter](#) for the contract.

## [-] Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual JsonConverter ResolveContractConve
    Type objectType
)
```

```
Protected Overridable Function ResolveContractConver
    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(**



# ResolvePropertyName

## Method (propertyName)

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

[DefaultContractResolver](#) ► **ResolvePropertyName(String)**

C# ▾

Resolves the name of the property.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
protected virtual string ResolvePropertyName(  
    string propertyName  
)
```

```
Protected Overridable Function ResolvePropertyName (  
    propertyName As String _  
) As String
```

```
protected:  
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: 4.0.2.0  
(4.0.2.13623)

## 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: 4.0.2.C  
(4.0.2.13623)

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

 Public  
 Protected

 Instance  
 Static

 Declared  
 Inherited

Icon	Member	Description
≡	<a href="#">DefaultSerializationBinder()</a>	Initializes a new instance of the <b>DefaultSerializationBinder</b> class
≡	<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> )

		<code>String).</code> )
≡	<a href="#">Equals(Object)</a>	Determines whether the specified <code>Object</code> is equal to the current <code>Object</code> (Inherited from <a href="#">Object</a> .)
💡	<a href="#">Finalize()</a>	Allows an <code>Object</code> to attempt to free resources and perform other cleanup operations before the <code>Object</code> 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 <code>Type</code> of the current instance (Inherited from <a href="#">Object</a> .)
💡	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the current <code>Object</code> . (Inherited from <a href="#">Object</a> .)
≡	<a href="#">ToString()</a>	Returns a <code>String</code> that represents the current <code>Object</code> . (Inherited from <a href="#">Object</a> .)

## [-] Inheritance Hierarchy

[Object](#)

└ [SerializationBinder](#)

  └ [DefaultSerializationBinder](#)

[1]

[1]

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public DefaultSerializationBinder()
```

```
Public Sub New
```

```
public:  
DefaultSerializationBinder()
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# BindToType Method (assemblyName, typeName)

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

[DefaultSerializationBinder](#) ► **BindToType(String, String)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# DynamicValueProvider

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DynamicValueProvider](#)

C# ▾

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 : IValueProvider
```

### 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="#">DynamicValueProvider(MemberInfo)</a>	Initializes a new instance of the <b>DynamicValueProvider</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: 4.0.2.C  
(4.0.2.13623)

# DynamicValueProvider Constructor (memberInfo)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[DynamicValueProvider](#) ► **DynamicValueProvider(MemberInfo)**

C# ▾

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

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public DynamicValueProvider(  
    MemberInfo memberInfo  
)
```

```
Public Sub New ( _  
    memberInfo As MemberInfo _  
)
```

```
public:  
DynamicValueProvider(  
    MemberInfo^ memberInfo  
)
```

## Parameters

### **memberInfo** ([MemberInfo](#))

The member info.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

C# ▾

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.

# SetValue Method (target, value)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [DynamicValueProvider](#) ► **SetValue(Object, Object)**

C# ▾

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.

C# ▾

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 Instance

Declared

 Protected Static

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="#">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](#)

Json.NET - Quick Starts & API Documentation

# Error Property

[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: 4.0.2.0  
(4.0.2.13623)

C#  
▼

Gets or sets a value indicating whether this [ErrorContext](#) is handled.

## 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: 4.0.2.0  
(4.0.2.13623)

# Member Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [ErrorContext](#) ► **Member**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

C# ▾

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

 Public  
 Protected

 Instance  
 Static

 Declared  
 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 <b>Object</b> is equal to the current <b>Object</b> .  (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
```

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# ErrorEventArgs

## Constructor (currentObject, errorContext)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [ErrorEventArgs](#) ►

[ErrorEventArgs\(Object, ErrorContext\)](#)

C# ▾

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.

C# ▾

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: 4.0.2.0  
(4.0.2.13623)



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: 4.0.2.0  
(4.0.2.13623)

# IContractResolver

## Interface

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► **IContractResolver**

C#

Used by [JsonSerializer](#) to resolves a [JsonContract](#) for a given [Type](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public interface IContractResolver
```

```
Public Interface IContractResolver
```

```
public interface class IContractResolver
```

### Members

All Members

Methods

Public

Instance

Declared

Protected

Static

Inherited

Icon	Member	Description
	<a href="#">ResolveContract(Type)</a>	Resolves the contract for a given type.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

# ResolveContract Method (*type*)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [IContractResolver](#) ► **ResolveContract(*Type*)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# IReferenceResolver

## Interface

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► **IReferenceResolver**

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.
≡		

	<a href="#"><u>ResolveReference(Object, String)</u></a>	Resolves a reference to its object.
--	---	-------------------------------------

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# AddReference Method (context, reference, value)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [IReferenceResolver](#)

► [AddReference\(Object, String, Object\)](#)

C#

Adds a reference to the specified object.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
void AddReference(
    Object context,
    string reference,
    Object value
)
```

```
Sub AddReference ( _
    context As Object, _
    reference As String, _
    value As Object _
)
```

```
void AddReference(
    Object^ context,
    String^ reference,
    Object^ value
)
```

## Parameters

### **context (Object)**

The serialization context.

### **reference (String)**

The reference.

### **value (Object)**

The object to reference.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# GetReference Method (context, value)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [IResolver](#)  
► **GetReference(Object, Object)**

C# ▾

Gets the reference for the specified object.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
string GetReference(  
    Object context,  
    Object value  
)
```

```
Function GetReference ( _  
    context As Object, _  
    value As Object _  
) As String
```

```
String^ GetReference(  
    Object^ context,  
    Object^ value  
)
```

## Parameters

### context (Object)

The serialization context.

### value (Object)

The object to get a reference for.

## Return Value

The reference to the object.

(4.0.2.13623)

# IsReferenced Method (context, value)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [IReferenceResolver](#)  
 ► **IsReferenced(Object, Object)**

C# ▾

Determines whether the specified object is referenced.

## 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: 4.0.2.C  
(4.0.2.13623)

# ResolveReference

## Method (context, reference)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [IResolverResolver](#)

► **ResolveReference(Object, String)**

C#

Resolves a reference to its object.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
Object ResolveReference(
    Object context,
    string reference
)
```

```
Function ResolveReference ( _
    context As Object, _
    reference As String _
) As Object
```

```
Object^ ResolveReference(
    Object^ context,
    String^ reference
)
```

### Parameters

#### **context (Object)**

The serialization context.

#### **reference (String)**

The reference to resolve.

### Return Value

The object that

(4.0.2.13623)

C# ▾

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

 Public Instance Declared Protected Static Inherited

Icon	Member	Description
≡	<a href="#">GetValue(Object)</a>	Gets the value.
≡	<a href="#">SetValue(Object, Object)</a>	Sets the value.



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.

# SetValue Method (target, value)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [IValueProvider](#) ► **SetValue(Object, Object)**

C# ▾

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.

Json.NET - Quick Starts & API Documentation

# 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 : JsonContract
```

```
Public Class JsonArrayContract _  
    Inherits JsonContract
```

```
public ref class JsonArrayContract : public JsonCont
```

## 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="#">JsonArrayContract(Type)</a>	Initializes a new instance of the <b>JsonArrayContract</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 [default creator 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](#)

  └ [JsonContract](#)

    └ **JsonArrayContract**



# JsonArrayContract

## Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonArrayContract](#) ► [JsonArrayContract\(Type\)](#)

C#

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

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonArrayContract(  
    Type underlyingType  
)
```

```
Public Sub New (  
    underlyingType As Type  
)
```

```
public:  
JsonArrayContract(  
    Type^ underlyingType  
)
```

### Parameters

#### *underlyingType* ([Type](#))

The underlying type for the contract.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonContract Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#)

C# ▾

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"/> 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="#">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 [default creator 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](#)



---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

## 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: 4.0.2.0  
(4.0.2.13623)

C#

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: 4.0.2.0  
(4.0.2.13623)

# DefaultCreatorNonPublic Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#) ► [DefaultCreatorNonPublic](#)

C# ▾

Gets or sets a value indicating whether [default creator 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: 4.0.2.C  
(4.0.2.13623)

# IsReference Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#) ► **IsReference**

C#

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: 4.0.2.0  
(4.0.2.13623)

## OnDeserialized

C#  
▼

Gets or sets the method called immediately after deserialization of the object.

### 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: 4.0.2.0  
(4.0.2.13623)

## OnDeserializing

C#

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: 4.0.2.0  
(4.0.2.13623)

# OnError Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#) ►

## OnError

C# ▾

Gets or sets the method called when an error is thrown during the serialization of the object.

### 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: 4.0.2.C  
(4.0.2.13623)

# OnSerialized Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonContract](#) ►

## OnSerialized

C#

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: 4.0.2.0  
(4.0.2.13623)

C#

Gets or sets the method called before serialization of the object.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public MethodInfo OnSerializing { get; set; }
```

```
Public Property OnSerializing As MethodInfo
    Get
    Set
```

```
public:
property MethodInfo^ OnSerializing {
    MethodInfo^ get ();
    void set (MethodInfo^ value);
}
```

## Value

The method called before serialization of the object.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# JsonDictionaryContract

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonDictionaryContract](#)

C# ▾

Contract details for a [Type](#) used by the [JsonSerializer](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonDictionaryContract : JsonContract
```

```
Public Class JsonDictionaryContract _
    Inherits JsonContract
```

```
public ref class JsonDictionaryContract : public JsonContract
```

### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherit

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 [default creator 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="#">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 [JsonContract](#).)

## **[-] Inheritance Hierarchy**

[Object](#)

└ [JsonContract](#)

└ **JsonDictionaryContract**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# JsonDictionaryContract

## Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonDictionaryContract](#) ► **JsonDictionaryContract(Type)**

C# ▾

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

### 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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonDictionaryContract](#) ► **PropertyNameResolver**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# JsonDynamicContract

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonDynamicContract](#)

C# ▾

Contract details for a [Type](#) used by the [JsonSerializer](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonDynamicContract : JsonContract
```

```
Public Class JsonDynamicContract _
    Inherits JsonContract
```

```
public ref class JsonDynamicContract : public JsonCo
```

### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherit

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 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 [default creator 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="#">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="#"><u>UnderlyingType</u></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: 4.0.2.0  
(4.0.2.13623)



# JsonDynamicContract

## Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonDynamicContract](#)  
► **JsonDynamicContract(Type)**

C#

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

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonDynamicContract(  
    Type underlyingType  
)
```

```
Public Sub New (  
    underlyingType As Type  
)
```

```
public:  
JsonDynamicContract(  
    Type^ underlyingType  
)
```

### Parameters

#### *underlyingType* ([Type](#))

The underlying type for the contract.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# Properties Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonDynamicContract](#)  
► **Properties**

Gets the object's properties.

C# ▾

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonPropertyCollection Properties { get; private set; }
```

```
Public Property Properties As JsonPropertyCollection
    Get
    Private Set
```

```
public:
property JsonPropertyCollection^ Properties {
    JsonPropertyCollection^ get ();
    private: void set (JsonPropertyCollection^ value);
}
```

## Value

The object's properties.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# PropertyNameResolver

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonDynamicContract](#)  
► **PropertyNameResolver**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# JsonSerializableContract

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► **JsonSerializableContract**

Contract details for a [Type](#) used by the [JsonSerializer](#).

C# ▾

### 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

Public

Protected

Instance

Static

Declare

Inherit

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>	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 [default creator 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="#">ISerializableCreator</a>	Gets or sets the ISerializable object constructor.

	<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> .

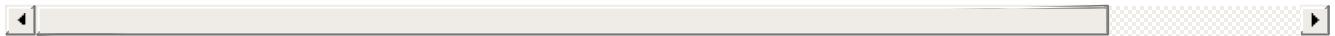
		(Inherited from <a href="#">Object</a> .)
	<a href="#">UnderlyingType</a>	Gets the underlying type for the contract.  (Inherited from <a href="#">JsonContract</a> .)

## **[-] Inheritance Hierarchy**

[Object](#)  
└ [JsonContract](#)  
  └ **JsonISerializableContract**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# JsonSerializableContract

## Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonSerializableContract](#) ► **JsonSerializableContract(Type)**

C#

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.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonSerializableContract](#) ► **ISerializableCreator**

C# ▾

Gets or sets the ISerializable object constructor.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public ObjectConstructor<Object> ISerializableCreate
```

```
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: 4.0.2.C  
(4.0.2.13623)



C# ▾

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

 Public Protected Instance Static

Declared



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>	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 [default creator 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](#)

  └ [JsonContract](#)

    └ **JsonLinqContract**

# JsonLinqContract

## Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonLinqContract](#) ► [JsonLinqContract\(Type\)](#)

C# ▾

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: 4.0.2.0 (4.0.2.13623)

# JsonObjectContract

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► **JsonObjectContract**

C# ▾

Contract details for a [Type](#) used by the [JsonSerializer](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonObjectContract : JsonContract
```

```
Public Class JsonObjectContract _
    Inherits JsonContract
```

```
public ref class JsonObjectContract : public JsonCon
```

### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherit

Icon	Member	Description
	<a href="#">JsonObjectContract(Type)</a>	Initializes a new instance of the <b>JsonObjectContract</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 [default creator 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="#">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](#)

  └ **JsonObjectContract**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# JsonObjectContract

## Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonObjectContract](#)  
► **JsonObjectContract(Type)**

C# ▾

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

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonObjectContract(  
    Type underlyingType  
)
```

```
Public Sub New (  
    underlyingType As Type  
)
```

```
public:  
JsonObjectContract(  
    Type^ underlyingType  
)
```

### Parameters

#### *underlyingType* ([Type](#))

The underlying type for the contract.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# MemberSerialization

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonObjectContract](#)

► **MemberSerialization**

C# ▾

Gets or sets the object member serialization.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public MemberSerialization MemberSerialization { get
```

```
Public Property MemberSerialization As MemberSerialization
    Get
    Set
```

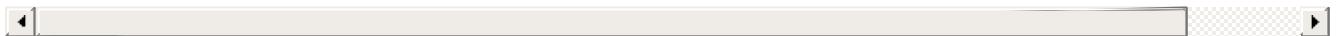
```
public:
property MemberSerialization MemberSerialization {
    MemberSerialization get ();
    void set (MemberSerialization value);
}
```

### Value

The member object serialization.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# OverrideConstructor

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonObjectContract](#)

► **OverrideConstructor**

C# ▾

Gets or sets the override constructor used to create the object. This is set when a constructor is marked up using the JsonConstructor attribute.

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public ConstructorInfo OverrideConstructor { get; se
```

```
Public Property OverrideConstructor As ConstructorIn
    Get
    Set
```

```
public:
property ConstructorInfo^ OverrideConstructor {
    ConstructorInfo^ get ();
    void set (ConstructorInfo^ value);
}
```

### Value

The override constructor.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



# ParametrizedConstructor

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonObjectContract](#)

► **ParametrizedConstructor**

C#

Gets or sets the parametrized constructor used to create the object.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public ConstructorInfo ParametrizedConstructor { get
```

```
Public Property ParametrizedConstructor As Construct
    Get
    Set
```

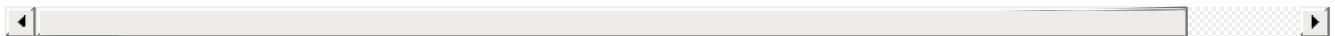
```
public:
property ConstructorInfo^ ParametrizedConstructor {
    ConstructorInfo^ get ();
    void set (ConstructorInfo^ value);
}
```

### Value

The parametrized constructor.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



Json.NET - Quick Starts & API Documentation

# Properties Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonObjectContract](#)

► **Properties**

Gets the object's properties.

C# ▾

## 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: 4.0.2.0  
(4.0.2.13623)



# JsonPrimitiveContract

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonPrimitiveContract](#)

C# ▾

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

Public

Protected

Instance

Static

Declare

Inherit

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 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 [default creator 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](#)

└ [JsonContract](#)

  └ [JsonPrimitiveContract](#)

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# JsonPrimitiveContract

## Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonPrimitiveContract](#)  
► **JsonPrimitiveContract(Type)**

C# ▾

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

### Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public JsonPrimitiveContract(  
    Type underlyingType  
)
```

```
Public Sub New (  
    underlyingType As Type  
)
```

```
public:  
JsonPrimitiveContract(  
    Type^ underlyingType  
)
```

### Parameters

#### *underlyingType* ([Type](#))

The underlying type for the contract.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

# JsonProperty Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#)

C# ▾

Maps a JSON property to a .NET member.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonProperty
```

```
Public Class JsonProperty
```

```
public ref class JsonProperty
```

## 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="#">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="#">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="#">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="#">PropertyName</a>	Gets 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 this instance.  (Overrides <a href="#">Object.ToString()</a> .)
 <a href="#">TypeNameHandling</a>	Gets or sets the type name handling
 <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: 4.0.2.0 (4.0.2.13623)

# JsonProperty

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ► [JsonProperty\(\)](#)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

## 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: 4.0.2.0  
(4.0.2.13623)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

**DefaultValueHandling**

C# ▾

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: 4.0.2.0 (4.0.2.13623)



## GetIsSpecified

C# ▾

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\(of Object\)
    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: 4.0.2.0  
(4.0.2.13623)

# Ignored Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ► [Ignored](#)

C# ▾

Gets a value indicating whether this [JsonProperty](#) is ignored.

## 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: 4.0.2.0 (4.0.2.13623)

# IsReference Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ► **IsReference**

C# ▾

Gets a value indicating whether this property preserves object references.

## 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: 4.0.2.0  
(4.0.2.13623)

# MemberConverter

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

**MemberConverter**

C# ▾

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: 4.0.2.0 (4.0.2.13623)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

### NullValueHandling

C# ▾

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> NullValueHandling
    Nullable<NullValueHandling> get ();
    void set (Nullable<NullValueHandling> value)
}
```

#### Value

The null value handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

[ObjectCreationHandling](#)

C# ▾

Gets the property object creation handling.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Nullable<ObjectCreationHandling> ObjectCreati
```

```
Public Property ObjectCreationHandling As Nullable<ObjectCreationHandling>
    Get
    Set
```

```
public:
property Nullable<ObjectCreationHandling> ObjectCreationHandling {
    Nullable<ObjectCreationHandling> get ();
    void set (Nullable<ObjectCreationHandling> v
}
```

### Value

The object creation handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



Json.NET - Quick Starts & API Documentation

# PropertyName Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ► [PropertyName](#)

Gets the name of the property.

C# ▾

## 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: 4.0.2.0  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# .PropertyType Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ► [.PropertyType](#)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

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: 4.0.2.0  
(4.0.2.13623)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ► [ReferenceLoopHandling](#)

C# ▾

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: 4.0.2.0 (4.0.2.13623)



# Required Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ► **Required**

C# ▾

Gets a value indicating whether this [JsonProperty](#) 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 [JsonProperty](#) is required.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

## SetIsSpecified

C# ▾

Gets or sets an action used to set whether the property has been deserialized.

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public Action SetIsSpecified { get; set; }
```

```
Public Property SetIsSpecified As Action
    Get
    Set
```

```
public:
property Action^ SetIsSpecified {
    Action^ get ();
    void set (Action^ value);
}
```

### Value

An action used to set whether the property has been deserialized.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)

Json.NET - Quick Starts & API Documentation

# ShouldSerialize Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ► [ShouldSerialize](#)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# ToString Method

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ► [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: 4.0.2.C  
(4.0.2.13623)

## Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ►

### TypeNameHandling

C# ▾

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
```

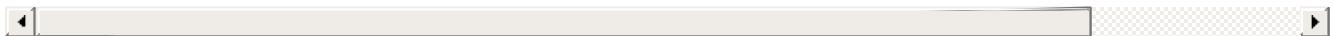
```
public:
property Nullable<TypeNameHandling> TypeNameHandling
    Nullable<TypeNameHandling> get ();
    void set (Nullable<TypeNameHandling> value);
}
```

#### Value

The type name handling.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)



C# ▾

Gets the [IValueProvider](#) that will get and set the [JsonProperty](#) during serialization.

## Declaration Syntax

C#

Visual Basic

Visual C++ ▾

```
public IValueProvider ValueProvider { get; set; }
```

```
Public Property ValueProvider As IValueProvider
    Get
    Set
```

```
public:
property IValueProvider^ ValueProvider {
    IValueProvider^ get ();
    void set (IValueProvider^ value);
}
```

## Value

The [IValueProvider](#) that will get and set the [JsonProperty](#) during serialization.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)

# Writable Property

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonProperty](#) ► **Writable**

C# ▾

Gets a value indicating whether this [JsonProperty](#) is writable.

## 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: 4.0.2.0 (4.0.2.13623)

# JsonPropertyCollection Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonPropertyCollection](#)

C# ▾

A collection of [JsonProperty](#) objects.

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonPropertyCollection : KeyedCollection
```

```
Public Class JsonPropertyCollection
    Inherits KeyedCollection(Of String, JsonProperty)
```

```
public ref class JsonPropertyCollection : public KeyedCollection
```

## Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declared

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 <b>Collection&lt;T&gt;</b> .  (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 is 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> compatible one-dimensional <a href="#">Array</a> starting at the specified index to 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="#">KeyedCollection&lt;TKey, 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 attempts to 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; TKey, TValue &gt;.GetKeyForItem</a> .)
≡	<a href="#">GetProperty(String, StringComparison)</a>	Gets a property by property name.  (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;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>	Inserts an element into the <a href="#">KeyedCollection&lt; TKey, TValue &gt;</a> at the specified index.

		(Inherited from <a href="#">KeyedCollection&lt;String, JsonProperty&gt;</a> .)
	<a href="#">Item[TKey]</a>	Gets the element with the specified key.  (Inherited from <a href="#">KeyedCollection&lt;String, JsonProperty&gt;</a> .)
	<a href="#">Item[Int32]</a>	Gets or sets the element at the specified index.  (Inherited from <a href="#">Collection&lt;JsonProperty&gt;</a> .)
	<a href="#">Items</a>	Gets a <a href="#">IList&lt;T&gt;</a> wrapper around <a href="#">Collection&lt;T&gt;</a> .  (Inherited from <a href="#">Collection&lt;JsonProperty&gt;</a> .)
	<a href="#">MemberwiseClone()</a>	Creates a shallow copy of the <a href="#">Object</a> .  (Inherited from <a href="#">Object</a> .)
	<a href="#">Remove(TKey)</a>	Removes the element with the specified key from the <a href="#">KeyedCollection&lt;TKey, TItem&gt;</a> .  (Inherited from <a href="#">KeyedCollection&lt;String, 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: 4.0.2.0 (4.0.2.13623)



# JsonPropertyCollection

## Constructor (type)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[JsonPropertyCollection](#) ► **JsonPropertyCollection(Type)**

C# ▾

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

### 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: 4.0.2.0 (4.0.2.13623)

# AddProperty Method (property)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[JsonPropertyCollection](#) ► **AddProperty(JsonProperty)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# 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.

# GetKeyForItem Method (item)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[JsonPropertyCollection](#) ► **GetKeyForItem(JsonProperty)**

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# GetProperty Method (*propertyName*, *comparisonType*)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[JsonPropertyCollection](#) ► **GetProperty(String, StringComparison)**

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# JsonStringContract

## Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► **JsonStringContract**

C# ▾

Contract details for a [Type](#) used by the [JsonSerializer](#).

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public class JsonStringContract : JsonContract
```

```
Public Class JsonStringContract _
    Inherits JsonContract
```

```
public ref class JsonStringContract : public JsonCon
```

### Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherit

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 [default creator 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](#)

└ [JsonContract](#)

  └ [JsonStringContract](#)

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0  
(4.0.2.13623)



# JsonStringContract

## Constructor (underlyingType)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [JsonStringContract](#) ► **JsonStringContract(Type)**

C# ▾

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

### Declaration Syntax

C#

Visual Basic

Visual C++

```
public JsonStringContract(  
    Type underlyingType  
)
```

```
Public Sub New (  
    underlyingType As Type  
)
```

```
public:  
JsonStringContract(  
    Type^ underlyingType  
)
```

### Parameters

#### *underlyingType* ([Type](#))

The underlying type for the contract.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

## Delegate

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

**ObjectConstructor<T>**

C# ▾

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
<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="#">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 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 <a href="#">Attribute</a> .  (Inherited from <a href="#">Attribute</a> .)

## Inheritance Hierarchy

[Object](#)

└ [Attribute](#)

  └ **OnErrorAttribute**

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.C  
(4.0.2.13623)



# OnErrorAttribute

## Constructor

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [OnErrorAttribute](#) ► [OnErrorAttribute\(\)](#)

C# ▾

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: 4.0.2.0  
(4.0.2.13623)

# ReflectionValueProvider Class

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [ReflectionValueProvider](#)

C# ▾

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

Public

Protected

Instance

Static

Declares

Implements

Inherits

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: 4.0.2.C  
(4.0.2.13623)



# ReflectionValueProvider Constructor (memberInfo)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ►  
[ReflectionValueProvider](#) ► **ReflectionValueProvider(MemberInfo)**

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

C#

## Declaration Syntax

C#

Visual Basic

Visual C++

```
public ReflectionValueProvider(  
    MemberInfo memberInfo  
)
```

```
Public Sub New ( _  
    memberInfo As MemberInfo _  
)
```

```
public:  
ReflectionValueProvider(  
    MemberInfo^ memberInfo  
)
```

## Parameters

### **memberInfo** ([MemberInfo](#))

The member info.

---

Assembly: Newtonsoft.Json (Module: Newtonsoft.Json) Version: 4.0.2.0 (4.0.2.13623)

C# ▾

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: 4.0.2.C  
(4.0.2.13623)

# SetValue Method (target, value)

[Namespaces](#) ► [Newtonsoft.Json.Serialization](#) ► [ReflectionValueProvider](#) ► **SetValue(Object, Object)**

C# ▾

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

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 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.Linq.ComponentModel</a>	The <b>Newtonsoft.Json.Linq.ComponentModel</b> namespace provides classes for JSON databinding.
<a href="#">Newtonsoft.Json.Schema</a>	The <b>Newtonsoft.Json.Schema</b> namespace provides classes to implement JSON schema validation.

## [\*\*Newtonsoft.Json.Serialization\*\*](#)

The **Newtonsoft.Json.Linq** namespace provides classes that are used for serializing and deserializing JSON objects.

