



[James Newton-King](#)

No pressure, no diamonds

- [Home](#)
- [Archive](#)
- [About Me](#)

**DynamicPDF Core Suite for .NET**
Real-time PDF solutions (Generator, Merger & ReportWriter)
www.dynamicpdf.com
[TRY THEM FREE TODAY ►](#)



[Rollover for more information](#)
[Replay](#)

Json.NET

Json.NET is a popular high-performance JSON framework for .NET

[Json.NET CodePlex Project](#)

[Json.NET Download](#)

Features

- Flexible JSON serializer for converting between .NET objects and JSON
- LINQ to JSON for manually reading and writing JSON
- High performance, faster than .NET's built-in JSON serializers
- Write indented, easy to read JSON
- Convert JSON to and from XML
- Supports .NET 2, .NET 3.5, .NET 4, Silverlight and Windows Phone

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

LINQ to JSON is good for situations where you are only interested in getting values from JSON, you 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.

Documentation

[Json.NET - Documentation](#)

Serialization Example

```
Product product = new Product();  
product.Name = "Apple";
```

```

product.Expiry = new DateTime(2008, 12, 28);
product.Price = 3.99M;
product.Sizes = new string[] { "Small", "Medium", "Large" };

string json = JsonConvert.SerializeObject(product);
//{
//  "Name": "Apple",
//  "Expiry": new Date(1230422400000),
//  "Price": 3.99,
//  "Sizes": [
//    "Small",
//    "Medium",
//    "Large"
//  ]
//}

Product deserializedProduct = JsonConvert.DeserializeObject<Product>(json);

```

LINQ to JSON Example

```

string json = @"{
  ""Name"": ""Apple"",
  ""Expiry"": new Date(1230422400000),
  ""Price"": 3.99,
  ""Sizes"": [
    ""Small"",
    ""Medium"",
    ""Large""
  ]
}";

JObject o = JObject.Parse(json);

string name = (string)o["Name"];
// Apple

JArray sizes = (JArray)o["Sizes"];

string smallest = (string)sizes[0];
// Small

```

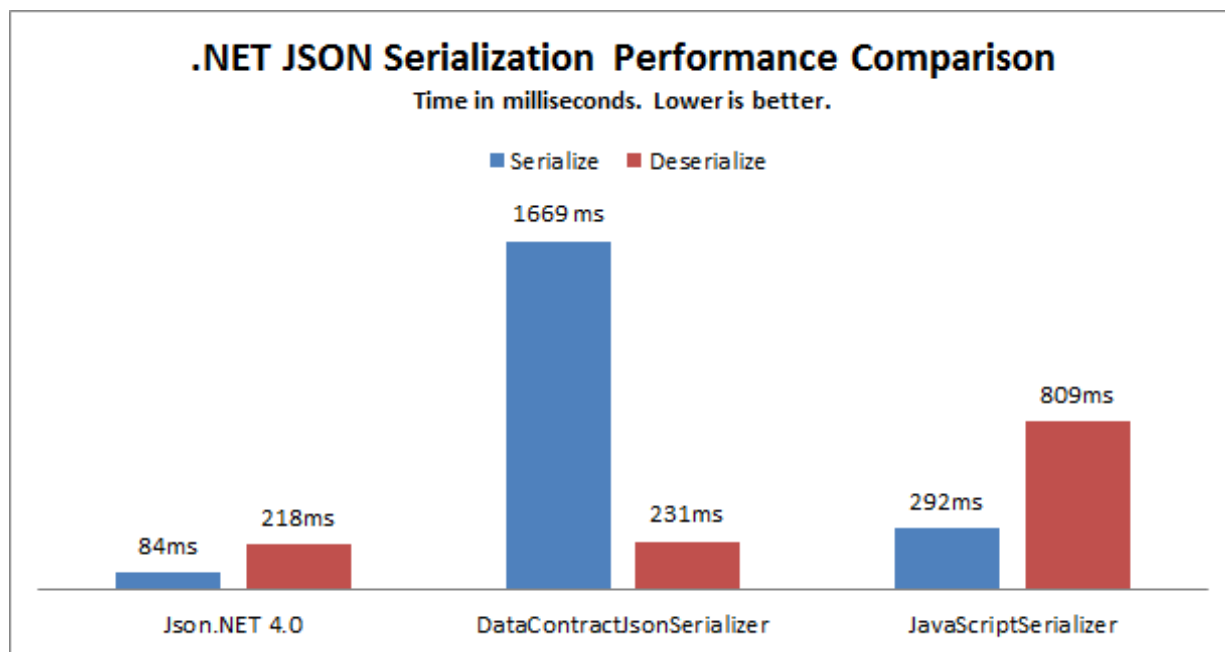
Feature Comparison

	Json.NET	DataContractJsonSerializer	JavaScriptSerializer
Supports JSON	✓	✓	✓
Supports BSON	✓	✗	✗
Supports JSON Schema	✓	✗	✗
Supports .NET 2.0	✓	✗	✗
Supports .NET 3.5	✓	✓	✓
Supports .NET 4.0	✓	✓	✓
Supports Silverlight	✓	✓	✗
Supports Windows Phone	✓	✓	✗
Open Source	✓	✗	✗
MIT License	✓	✗	✗
LINQ to JSON	✓	✗	✗

Thread Safe	✓	✓	✓
XPath-like JSON query syntax	✓	✗	✗
Indented JSON support	✓	✗	✗
Efficient dictionary serialization	✓	✗	✓
Nonsensical dictionary serialization	✗	✓	✗
Deserializes IList, IEnumerable, ICollection, IDictionary properties	✓	✗	✗
Serializes circular references	✓	✗	✗
Supports serializing objects by reference	✓	✗	✗
Deserializes polymorphic properties and collections	✓	✓	✓
Supports including type names with JSON	✓	✓	✓
Globally customize serialization process	✓	✓	✗
Supports excluding null values when serializing	✓	✗	✗
Supports SerializationBinder	✓	✗	✗
Conditional property serialization	✓	✗	✗
Includes line number information in errors	✓	✓	✗
Converts XML to JSON and JSON to XML	✓	✗	✗
JSON Schema validation	✓	✗	✗
JSON Schema generation from .NET types	✓	✗	✗
Camel case JSON property names	✓	✗	✗
Non-default constructors support	✓	✗	✗
Serialization error handling	✓	✗	✗
Supports populating an existing object	✓	✗	✗
Efficiently serializes byte arrays as base64 text	✓	✗	✗
Handles NaN, Infinity, -Infinity and undefined	✓	✗	✗
Handles JavaScript constructors	✓	✗	✗
Serializes .NET 4.0 dynamic objects	✓	✗	✗
Serializes ISerializable objects	✓	✗	✗
Supports serializing enums to their text name	✓	✗	✗
JSON recursion limit support	✓	✓	✓
Attribute property name customization	✓	✓	✗
Attribute property order customization	✓	✓	✗
Attribute property required customization	✓	✓	✗
Supports ISO8601 dates	✓	✗	✗
Supports JavaScript constructor dates	✓	✗	✗
Supports Microsoft AJAX dates	✓	✓	✓
Unquoted property names support	✓	✗	✗
Raw JSON support	✓	✗	✗

Supports reading and writing comments	✓	✗	✗
Deserializes anonymous types	✓	✗	✗
Opt-in property serialization	✓	✓	✗
Opt-out property serialization	✓	✗	✓
Efficiently stream reading and writing JSON	✓	✓	✗
Single or double quote JSON content	✓	✗	✗
Supports overriding a type's serialization	✓	✗	✓
Supports OnDeserialized, OnSerializing, OnSerialized and OnDeserializing attributes	✓	✓	✗
Supports serializing private properties	✓	✓	✗
DataMember attribute support	✓	✓	✗
MetadataType attribute support	✓	✗	✗
DefaultValue attribute support	✓	✗	✗
Serializes DataSets and DataTables	✓	✗	✗
Serializes Entity Framework	✓	✗	✗
Serializes nHibernate	✓	✗	✗
Case-insensitive property deserialization	✓	✗	✗

Performance Comparison



The code for this benchmark is in the Json.NET unit tests.

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 such as [MonoRail](#), Castle Project's MVC web framework, and [Mono](#), an open source implementation of the .NET framework.

Donate

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

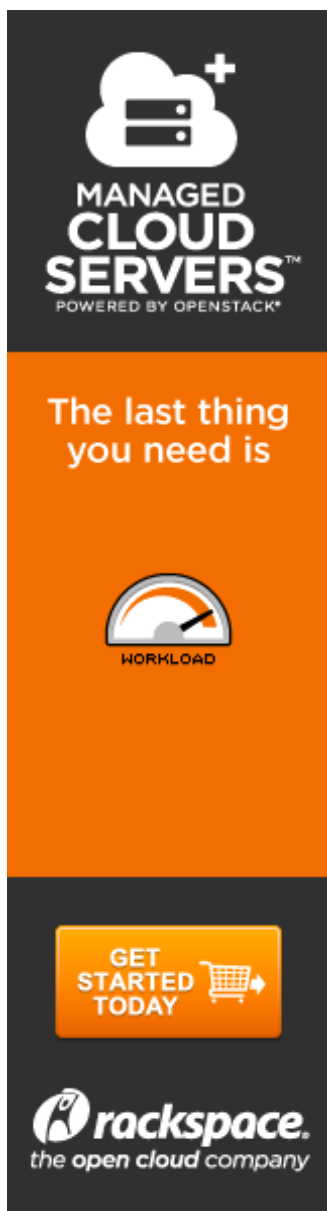
I really appreciate your feedback and support for Json.NET and its future development.



September 09, 2007




- [Json.NET](#)
- [Sitemaps.NET](#)




MANAGED
CLOUD
SERVERS™
POWERED BY OPENSTACK®

The last thing
you need is



WORKLOAD

GET
STARTED
TODAY



rackspace.
the open cloud company

Recent Posts

- [Json.NET vs Windows.Data.Json](#)
- [Json.NET 4.5 Release 8 – Multidimensional Array Support, Unicode Improvements](#)
- [Json.NET 4.5 Release 5 – JsonProperty enhancements](#)
- [Json.NET 4.5 Release 4 – Portable Class Library Build](#)
- [Json.NET 4.5 Release 2 – Serializable support and bug fixes](#)



© 2012 James Newton-King

- [Home](#)
- [RSS for Posts](#)

