
Test Documentation

Release test

Test

March 23, 2015

1	XML Output	1
1.1	BsonObjectId	1
1.2	BsonReader	1
1.3	JsonReader	1
1.4	State	3
1.5	BsonWriter	5
1.6	JsonWriter	5
1.7	ConstructorHandling	14
1.8	BsonObjectIdConverter	14
1.9	JsonConverter	14
1.10	CustomCreationConverter ¹	15
1.11	DateTimeConverterBase	16
1.12	DiscriminatedUnionConverter	16
1.13	ExpandoObjectConverter	17
1.14	IsoDateTimeConverter	17
1.15	JavaScriptDateTimeConverter	18
1.16	KeyValuePairConverter	18
1.17	RegexConverter	19
1.18	StringEnumConverter	19
1.19	VersionConverter	20
1.20	XmlNodeConverter	21
1.21	DateFormatHandling	21
1.22	DateParseHandling	22
1.23	DateTimeZoneHandling	22
1.24	DefaultValueHandling	22
1.25	FloatFormatHandling	23
1.26	FloatParseHandling	23
1.27	FormatterAssemblyStyle	23
1.28	Formatting	24
1.29	IJsonLineInfo	24
1.30	JsonArrayAttribute	24
1.31	JsonContainerAttribute	24
1.32	JsonConstructorAttribute	25
1.33	JsonConvert	26
1.34	JsonConverterAttribute	32
1.35	JsonConverterCollection	33
1.36	JsonDictionaryAttribute	33
1.37	JsonException	33

1.38	JsonExtensionDataAttribute	34
1.39	JsonIgnoreAttribute	34
1.40	JsonObjectAttribute	34
1.41	JsonPropertyAttribute	35
1.42	JsonReaderException	36
1.43	JsonSerializationException	36
1.44	JsonSerializer	37
1.45	JsonSerializerSettings	40
1.46	JsonTextReader	42
1.47	JsonTextWriter	43
1.48	JsonToken	46
1.49	JsonValidatingReader	47
1.50	JsonWriterException	48
1.51	Extensions	49
1.52	IEnumerable<T>	51
1.53	JArray	51
1.54	JContainer	51
1.55	JToken	51
1.56	JConstructor	65
1.57	JEnumerable<T>	66
1.58	JObject	67
1.59	JProperty	69
1.60	JRaw	70
1.61	JValue	70
1.62	JsonMergeSettings	73
1.63	JTokenEqualityComparer	73
1.64	JTokenReader	73
1.65	JTokenType	74
1.66	JTokenWriter	75
1.67	MergeArrayHandling	78
1.68	MemberSerialization	78
1.69	MetadataPropertyHandling	79
1.70	MissingMemberHandling	79
1.71	NullValueHandling	79
1.72	ObjectCreationHandling	79
1.73	PreserveReferencesHandling	80
1.74	ReferenceLoopHandling	80
1.75	Required	80
1.76	Extensions	81
1.77	JsonSchema	81
1.78	JsonSchemaException	84
1.79	JsonSchemaGenerator	85
1.80	JsonSchemaResolver	85
1.81	JsonSchemaType	86
1.82	UndefinedSchemaIdHandling	86
1.83	ValidationEventArgs	87
1.84	ValidationEventHandler	87
1.85	SerializationBinder	87
1.86	CamelCasePropertyNamesContractResolver	87
1.87	DefaultContractResolver	88
1.88	IContractResolver	88
1.89	ExpressionValueProvider	90
1.90	IValueProvider	90
1.91	IReferenceResolver	91

1.92	DefaultSerializationBinder	91
1.93	ErrorContext	92
1.94	ErrorEventArgs	92
1.95	IAttributeProvider	92
1.96	ITraceWriter	93
1.97	JsonArrayContract	93
1.98	JsonContainerContract	93
1.99	JsonContract	93
1.100	SerializationCallback	95
1.101	SerializationErrorCallback	95
1.102	ExtensionDataSetter	95
1.103	ExtensionDataGetter	95
1.104	JsonDictionaryContract	95
1.105	JsonDynamicContract	96
1.106	JsonLinqContract	96
1.107	JsonObjectContract	96
1.108	JsonPrimitiveContract	97
1.109	JsonProperty	97
1.110	JsonPropertyCollection	99
1.111	JsonStringContract	100
1.112	MemoryTraceWriter	100
1.113	ObjectConstructor'1	101
1.114	OnErrorAttribute	101
1.115	ReflectionAttributeProvider	101
1.116	ReflectionValueProvider	101
1.117	StringEscapeHandling	102
1.118	TraceLevel	102
1.119	TypeNameHandling	102
1.120	StringBuffer	104
1.121	WriteState	105

XML Output

Warning: This is a proof of concept, and is vastly unfinished. Only a basic tech spike.

1.1 BsonObjectId

class `Newtonsoft.Json.Bson.BsonObjectId`

Represents a BSON ObjectId (object id).

Newtonsoft.Json.Bson.BsonObjectId.#ctor(System.Byte[])

Initializes a new instance of the `Newtonsoft.Json.Bson.BsonObjectId`:param value: The ObjectId value.

Value

Gets or sets the value of the ObjectId.

1.2 BsonReader

class `Newtonsoft.Json.Bson.BsonReader`

Represents a reader that provides fast, non-cached, forward-only access to serialized JSON data.

1.3 JsonReader

class `Newtonsoft.Json.JsonReader`

Represents a reader that provides fast, non-cached, forward-only access to serialized JSON data.

Newtonsoft.Json.JsonReader.#ctor

Initializes a new instance of the `Newtonsoft.Json.JsonReader` `System.IO.TextReader`

Read()

Reads the next JSON token from the stream. :return: true if the next token was read successfully; false if there are no more tokens to read.

ReadAsInt32()

Reads the next JSON token from the stream as a `System.Nullable`1`:return: A `System.Nullable`1`

ReadAsString()

Reads the next JSON token from the stream as a `System.String` :return: A `System.String`

ReadAsBytes()

Reads the next JSON token from the stream as a `System.Byte` :return: A `System.Byte`

ReadAsDecimal()

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.Nullable`1`

ReadAsDateTime()

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.String`

ReadAsDateTimeOffset()

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.Nullable`1`

Skip()

Skips the children of the current token.

SetToken(Newtonsoft.Json.JsonToken)

Sets the current token. :param newToken: The new token.

SetToken(Newtonsoft.Json.JsonToken, System.Object)

Sets the current token and value. :param newToken: The new token. :param value: The value.

SetStateBasedOnCurrent()

Sets the state based on current token type.

Newtonsoft.Json.JsonReader.System#IDisposable#Dispose

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Dispose(System.Boolean)

Releases unmanaged and - optionally - managed resources :param disposing: None

Close()

Changes the `Newtonsoft.Json.JsonReader.State`

CurrentState

Gets the current reader state.

CloseInput

Gets or sets a value indicating whether the underlying stream or `System.IO.TextReader` `System.IO.TextReader`

SupportMultipleContent

Gets or sets a value indicating whether multiple pieces of JSON content can be read from a continuous stream without erroring.

QuoteChar

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

DateTimeZoneHandling

Get or set how `System.DateTime`

DateParseHandling

Get or set how date formatted strings, e.g. “/Date(1198908717056)/” and “2012-03-21T05:40Z”, are parsed when reading JSON.

FloatParseHandling

Get or set how floating point numbers, e.g. 1.0 and 9.9, are parsed when reading JSON text.

DateFormatString

Get or set how custom date formatted strings are parsed when reading JSON.

MaxDepth

Gets or sets the maximum depth allowed when reading JSON. Reading past this depth will throw a `Newtonsoft.Json.JsonReaderException`

TokenType

Gets the type of the current JSON token.

Value

Gets the text value of the current JSON token.

ValueType

Gets The Common Language Runtime (CLR) type for the current JSON token.

Depth

Gets the depth of the current token in the JSON document.

Path

Gets the path of the current JSON token.

Culture

Gets or sets the culture used when reading JSON. Defaults to `System.Globalization.CultureInfo.InvariantCulture`

1.4 State

class `Newtonsoft.Json.JsonReader.State`

Specifies the state of the reader.

Start

The Read method has not been called.

Complete

The end of the file has been reached successfully.

Property

Reader is at a property.

ObjectStart

Reader is at the start of an object.

Object

Reader is in an object.

ArrayStart

Reader is at the start of an array.

Array

Reader is in an array.

Closed

The Close method has been called.

PostValue

Reader has just read a value.

ConstructorStart

Reader is at the start of a constructor.

Constructor

Reader in a constructor.

Error

An error occurred that prevents the read operation from continuing.

Finished

The end of the file has been reached successfully.

Newtonsoft.Json.Bson.BsonReader.#ctor(System.IO.Stream)

Initializes a new instance of the `Newtonsoft.Json.Bson.BsonReader` :param stream: The stream.

Newtonsoft.Json.Bson.BsonReader.#ctor(System.IO.BinaryReader)

Initializes a new instance of the `Newtonsoft.Json.Bson.BsonReader` :param reader: The reader.

Newtonsoft.Json.Bson.BsonReader.#ctor(System.IO.Stream, System.Boolean, System.DateTimeKind)

Initializes a new instance of the `Newtonsoft.Json.Bson.BsonReader` :param stream: The stream. :param readRootValueAsArray: if set to :param dateTimeKindHandling: The

`System.DateTimeKind` `System.DateTime`

Newtonsoft.Json.Bson.BsonReader.#ctor(System.IO.BinaryReader, System.Boolean, System.DateTimeKind)

Initializes a new instance of the `Newtonsoft.Json.Bson.BsonReader` :param reader: The reader. :param readRootValueAsArray: if set to :param dateTimeKindHandling: The

`System.DateTimeKind` `System.DateTime`

Newtonsoft.Json.Bson.BsonReader.ReadAsBytes()

Reads the next JSON token from the stream as a `System.Byte` :return: A `System.Byte`

Newtonsoft.Json.Bson.BsonReader.ReadAsDecimal()

Reads the next JSON token from the stream as a `System.Nullable<decimal>` :return: A `System.Nullable<decimal>`

Newtonsoft.Json.Bson.BsonReader.ReadAsInt32()

Reads the next JSON token from the stream as a `System.Nullable<int>` :return: A `System.Nullable<int>`

`Newtonsoft.Json.Bson.BsonReader.ReadAsString()`

Reads the next JSON token from the stream as a `System.String` :return: A `System.String`

`Newtonsoft.Json.Bson.BsonReader.ReadAsDateTime()`

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.String`

`Newtonsoft.Json.Bson.BsonReader.ReadAsDateTimeOffset()`

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.Nullable`1`

`Newtonsoft.Json.Bson.BsonReader.Read()`

Reads the next JSON token from the stream. :return: true if the next token was read successfully; false if there are no more tokens to read.

`Newtonsoft.Json.Bson.BsonReader.Close()`

Changes the `Newtonsoft.Json.JsonReader.State`

`Newtonsoft.Json.Bson.BsonReader.JsonNet35BinaryCompatibility`

Gets or sets a value indicating whether binary data reading should compatible with incorrect `Json.NET 3.5` written binary.

`Newtonsoft.Json.Bson.BsonReader.ReadRootValueAsArray`

Gets or sets a value indicating whether the root object will be read as a JSON array.

`Newtonsoft.Json.Bson.BsonReader.DateTimeKindHandling`

Gets or sets the `System.DateTimeKind` `System.DateTime` `System.DateTimeKind` `System.DateTime`

1.5 BsonWriter

class `Newtonsoft.Json.Bson.BsonWriter`

Represents a writer that provides a fast, non-cached, forward-only way of generating JSON data.

1.6 JsonWriter

class `Newtonsoft.Json.JsonWriter`

Represents a writer that provides a fast, non-cached, forward-only way of generating JSON data.

Newtonsoft.Json.JsonWriter.#ctor

Creates an instance of the

Flush()

Flushes whatever is in the buffer to the underlying streams and also flushes the underlying stream.

Close()

Closes this stream and the underlying stream.

WriteStartObject()

Writes the beginning of a `Json` object.

WriteEndObject ()

Writes the end of a Json object.

WriteStartArray ()

Writes the beginning of a Json array.

WriteEndArray ()

Writes the end of an array.

WriteStartConstructor (*System.String*)

Writes the start of a constructor with the given name. :param name: The name of the constructor.

WriteEndConstructor ()

Writes the end constructor.

WritePropertyName (*System.String*)

Writes the property name of a name/value pair on a JSON object. :param name: The name of the property.

WritePropertyName (*System.String, System.Boolean*)

Writes the property name of a name/value pair on a JSON object. :param name: The name of the property. :param escape: A flag to indicate whether the text should be escaped when it is written as a JSON property name.

WriteEnd ()

Writes the end of the current Json object or array.

WriteToken (*Newtonsoft.Json.JsonReader*)

Writes the current `Newtonsoft.Json.JsonReader` :param reader: The

`Newtonsoft.Json.JsonReader`

WriteToken (*Newtonsoft.Json.JsonReader, System.Boolean*)

Writes the current `Newtonsoft.Json.JsonReader` :param reader: The

`Newtonsoft.Json.JsonReader` :param writeChildren: A flag indicating whether the current token's children should be written.

WriteToken (*Newtonsoft.Json.JsonToken, System.Object*)

Writes the `Newtonsoft.Json.JsonToken` :param token: The

`Newtonsoft.Json.JsonToken` :param value: The value to write. A value is only required for tokens that have an associated value, e.g. the

`System.String` `Newtonsoft.Json.JsonToken.PropertyName`
`Newtonsoft.Json.JsonToken.StartObject`

WriteToken (*Newtonsoft.Json.JsonToken*)

Writes the `Newtonsoft.Json.JsonToken` :param token: The

`Newtonsoft.Json.JsonToken`

WriteEnd (*Newtonsoft.Json.JsonToken*)

Writes the specified end token. :param token: The end token to write.

WriteIndent ()

Writes indent characters.

WriteValueDelimiter ()

Writes the JSON value delimiter.

WriteIndentSpace ()

Writes an indent space.

WriteNull ()

Writes a null value.

WriteUndefined ()

Writes an undefined value.

WriteRaw (System.String)

Writes raw JSON without changing the writer's state. :param json: The raw JSON to write.

WriteRawValue (System.String)

Writes raw JSON where a value is expected and updates the writer's state. :param json: The raw JSON to write.

WriteValue (System.String)

Writes a System.String :param value: The

System.String

WriteValue (System.Int32)

Writes a System.Int32 :param value: The

System.Int32

WriteValue (System.UInt32)

Writes a System.UInt32 :param value: The

System.UInt32

WriteValue (System.Int64)

Writes a System.Int64 :param value: The

System.Int64

WriteValue (System.UInt64)

Writes a System.UInt64 :param value: The

System.UInt64

WriteValue (System.Single)

Writes a System.Single :param value: The

System.Single

WriteValue (System.Double)

Writes a System.Double :param value: The

System.Double

WriteValue (*System.Boolean*)

Writes a System.Boolean :param value: The
System.Boolean

WriteValue (*System.Int16*)

Writes a System.Int16 :param value: The
System.Int16

WriteValue (*System.UInt16*)

Writes a System.UInt16 :param value: The
System.UInt16

WriteValue (*System.Char*)

Writes a System.Char :param value: The
System.Char

WriteValue (*System.Byte*)

Writes a System.Byte :param value: The
System.Byte

WriteValue (*System.SByte*)

Writes a System.SByte :param value: The
System.SByte

WriteValue (*System.Decimal*)

Writes a System.Decimal :param value: The
System.Decimal

WriteValue (*System.DateTime*)

Writes a System.DateTime :param value: The
System.DateTime

WriteValue (*System.DateTimeOffset*)

Writes a System.DateTimeOffset :param value: The
System.DateTimeOffset

WriteValue (*System.Guid*)

Writes a System.Guid :param value: The
System.Guid

WriteValue (*System.TimeSpan*)

Writes a System.TimeSpan :param value: The
System.TimeSpan

WriteValue (*System.Nullable<System.Int32>*)

Writes a System.Nullable`1 :param value: The
System.Nullable`1

WriteValue (*System.Nullable{System.UInt32}*)

Writes a System.Nullable '1 :param value: The
System.Nullable '1

WriteValue (*System.Nullable{System.Int64}*)

Writes a System.Nullable '1 :param value: The
System.Nullable '1

WriteValue (*System.Nullable{System.UInt64}*)

Writes a System.Nullable '1 :param value: The
System.Nullable '1

WriteValue (*System.Nullable{System.Single}*)

Writes a System.Nullable '1 :param value: The
System.Nullable '1

WriteValue (*System.Nullable{System.Double}*)

Writes a System.Nullable '1 :param value: The
System.Nullable '1

WriteValue (*System.Nullable{System.Boolean}*)

Writes a System.Nullable '1 :param value: The
System.Nullable '1

WriteValue (*System.Nullable{System.Int16}*)

Writes a System.Nullable '1 :param value: The
System.Nullable '1

WriteValue (*System.Nullable{System.UInt16}*)

Writes a System.Nullable '1 :param value: The
System.Nullable '1

WriteValue (*System.Nullable{System.Char}*)

Writes a System.Nullable '1 :param value: The
System.Nullable '1

WriteValue (*System.Nullable{System.Byte}*)

Writes a System.Nullable '1 :param value: The
System.Nullable '1

WriteValue (*System.Nullable{System.SByte}*)

Writes a System.Nullable '1 :param value: The
System.Nullable '1

WriteValue (*System.Nullable{System.Decimal}*)

Writes a System.Nullable '1 :param value: The
System.Nullable '1

WriteValue (*System.Nullable{System.DateTime}*)

Writes a System.Nullable `1 :param value: The
System.Nullable `1

WriteValue (*System.Nullable{System.DateTimeOffset}*)

Writes a System.Nullable `1 :param value: The
System.Nullable `1

WriteValue (*System.Nullable{System.Guid}*)

Writes a System.Nullable `1 :param value: The
System.Nullable `1

WriteValue (*System.Nullable{System.TimeSpan}*)

Writes a System.Nullable `1 :param value: The
System.Nullable `1

WriteValue (*System.Byte[]*)

Writes a System.Byte :param value: The
System.Byte

WriteValue (*System.Uri*)

Writes a System.Uri :param value: The
System.Uri

WriteValue (*System.Object*)

Writes a System.Object :param value: The
System.Object

WriteComment (*System.String*)

Writes out a comment :param text: Text to place inside the comment.

WriteWhitespace (*System.String*)

Writes out the given white space. :param ws: The string of white space characters.

SetWriteState (*Newtonsoft.Json.JsonToken, System.Object*)

Sets the state of the JsonWriter, :param token: The JsonToken being written. :param value: The value being written.

CloseOutput

Gets or sets a value indicating whether the underlying stream or System.IO.TextReader
System.IO.TextReader

Top

Gets the top.

WriteState

Gets the state of the writer.

Path

Gets the path of the writer.

Formatting

Indicates how JSON text output is formatted.

DateFormatHandling

Get or set how dates are written to JSON text.

DateTimeZoneHandling

Get or set how `System.DateTime`

StringEscapeHandling

Get or set how strings are escaped when writing JSON text.

FloatFormatHandling

Get or set how special floating point numbers, e.g. `System.Double.NaN`
`System.Double.PositiveInfinity` `System.Double.NegativeInfinity`

DateFormatString

Get or set how `System.DateTime` `System.DateTimeOffset`

Culture

Gets or sets the culture used when writing JSON. Defaults to
`System.Globalization.CultureInfo.InvariantCulture`

Newtonsoft.Json.Bson.BsonWriter.#ctor(System.IO.Stream)

Initializes a new instance of the `Newtonsoft.Json.Bson.BsonWriter` :param stream: The stream.

Newtonsoft.Json.Bson.BsonWriter.#ctor(System.IO.BinaryWriter)

Initializes a new instance of the `Newtonsoft.Json.Bson.BsonWriter` :param writer: The writer.

`Newtonsoft.Json.Bson.BsonWriter.Flush()`

Flushes whatever is in the buffer to the underlying streams and also flushes the underlying stream.

`Newtonsoft.Json.Bson.BsonWriter.WriteEnd(Newtonsoft.Json.JsonToken)`

Writes the end. :param token: The token.

`Newtonsoft.Json.Bson.BsonWriter.WriteComment(System.String)`

Writes out a comment :param text: Text to place inside the comment.

`Newtonsoft.Json.Bson.BsonWriter.WriteStartConstructor(System.String)`

Writes the start of a constructor with the given name. :param name: The name of the constructor.

`Newtonsoft.Json.Bson.BsonWriter.WriteRaw(System.String)`

Writes raw JSON. :param json: The raw JSON to write.

`Newtonsoft.Json.Bson.BsonWriter.WriteRawValue(System.String)`

Writes raw JSON where a value is expected and updates the writer's state. :param json: The raw JSON to write.

`Newtonsoft.Json.Bson.BsonWriter.WriteStartArray()`

Writes the beginning of a Json array.

`Newtonsoft.Json.Bson.BsonWriter.WriteStartObject()`

Writes the beginning of a Json object.

`Newtonsoft.Json.Bson.BsonWriter.WritePropertyName (System.String)`

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

`Newtonsoft.Json.Bson.BsonWriter.Close ()`

Closes this stream and the underlying stream.

`Newtonsoft.Json.Bson.BsonWriter.WriteValue (System.Object)`

Writes a `System.Object` :param value: The

`System.Object`

`Newtonsoft.Json.Bson.BsonWriter.WriteNull ()`

Writes a null value.

`Newtonsoft.Json.Bson.BsonWriter.WriteUndefined ()`

Writes an undefined value.

`Newtonsoft.Json.Bson.BsonWriter.WriteValue (System.String)`

Writes a `System.String` :param value: The

`System.String`

`Newtonsoft.Json.Bson.BsonWriter.WriteValue (System.Int32)`

Writes a `System.Int32` :param value: The

`System.Int32`

`Newtonsoft.Json.Bson.BsonWriter.WriteValue (System.UInt32)`

Writes a `System.UInt32` :param value: The

`System.UInt32`

`Newtonsoft.Json.Bson.BsonWriter.WriteValue (System.Int64)`

Writes a `System.Int64` :param value: The

`System.Int64`

`Newtonsoft.Json.Bson.BsonWriter.WriteValue (System.UInt64)`

Writes a `System.UInt64` :param value: The

`System.UInt64`

`Newtonsoft.Json.Bson.BsonWriter.WriteValue (System.Single)`

Writes a `System.Single` :param value: The

`System.Single`

`Newtonsoft.Json.Bson.BsonWriter.WriteValue (System.Double)`

Writes a `System.Double` :param value: The

`System.Double`

`Newtonsoft.Json.Bson.BsonWriter.WriteValue (System.Boolean)`

Writes a `System.Boolean` :param value: The

System.Boolean

Newtonsoft.Json.Bson.BsonWriter.**WriteValue** (*System.Int16*)

Writes a System.Int16 :param value: The

System.Int16

Newtonsoft.Json.Bson.BsonWriter.**WriteValue** (*System.UInt16*)

Writes a System.UInt16 :param value: The

System.UInt16

Newtonsoft.Json.Bson.BsonWriter.**WriteValue** (*System.Char*)

Writes a System.Char :param value: The

System.Char

Newtonsoft.Json.Bson.BsonWriter.**WriteValue** (*System.Byte*)

Writes a System.Byte :param value: The

System.Byte

Newtonsoft.Json.Bson.BsonWriter.**WriteValue** (*System.SByte*)

Writes a System.SByte :param value: The

System.SByte

Newtonsoft.Json.Bson.BsonWriter.**WriteValue** (*System.Decimal*)

Writes a System.Decimal :param value: The

System.Decimal

Newtonsoft.Json.Bson.BsonWriter.**WriteValue** (*System.DateTime*)

Writes a System.DateTime :param value: The

System.DateTime

Newtonsoft.Json.Bson.BsonWriter.**WriteValue** (*System.DateTimeOffset*)

Writes a System.DateTimeOffset :param value: The

System.DateTimeOffset

Newtonsoft.Json.Bson.BsonWriter.**WriteValue** (*System.Byte[]*)

Writes a System.Byte :param value: The

System.Byte

Newtonsoft.Json.Bson.BsonWriter.**WriteValue** (*System.Guid*)

Writes a System.Guid :param value: The

System.Guid

Newtonsoft.Json.Bson.BsonWriter.**WriteValue** (*System.TimeSpan*)

Writes a System.TimeSpan :param value: The

System.TimeSpan

Newtonsoft.Json.Bson.BsonWriter.**WriteValue** (*System.Uri*)

Writes a System.Uri :param value: The

`System.Uri`

`Newtonsoft.Json.Bson.BsonWriter.WriteObjectId(System.Byte[])`

Writes a `System.Byte` :param value: The Object ID value to write.

`Newtonsoft.Json.Bson.BsonWriter.WriteRegex(System.String, System.String)`

Writes a BSON regex. :param pattern: The regex pattern. :param options: The regex options.

`Newtonsoft.Json.Bson.BsonWriter.DateTimeKindHandling`

Gets or sets the `System.DateTimeKind` `System.DateTime`
`System.DateTimeKind.Unspecified` `System.DateTimeKind` `System.DateTime`

1.7 ConstructorHandling

class `Newtonsoft.Json.ConstructorHandling`

Specifies how constructors are used when initializing objects during deserialization by the `Newtonsoft.Json.JsonSerializer`

Default

First attempt to use the public default constructor, then fall back to single paramatized constructor, then the non-public default constructor.

AllowNonPublicDefaultConstructor

Json.NET will use a non-public default constructor before falling back to a paramatized constructor.

1.8 BsonObjectIdConverter

class `Newtonsoft.Json.Converters.BsonObjectIdConverter`

Converts a `Newtonsoft.Json.Bson.BsonObjectId`

1.9 JsonConvert

class `Newtonsoft.Json.JsonConverter`

Converts an object to and from JSON.

WriteJson (`Newtonsoft.Json.JsonWriter`, `System.Object`, `Newtonsoft.Json.JsonSerializer`)

Writes the JSON representation of the object. :param writer: The

`Newtonsoft.Json.JsonWriter` :param value: The value.

param serializer The calling serializer.

ReadJson (`Newtonsoft.Json.JsonReader`, `System.Type`, `System.Object`, `Newtonsoft.Json.JsonSerializer`)

Reads the JSON representation of the object. :param reader: The

`Newtonsoft.Json.JsonReader` :param objectType: Type of the object.

param existingValue The existing value of object being read.

param serializer The calling serializer.

return The object value.

CanConvert (*System.Type*)

Determines whether this instance can convert the specified object type. :param objectType: Type of the object. :return:

GetSchema ()

Gets the `Newtonsoft.Json.Schema.JsonSchema` :return: The `Newtonsoft.Json.Schema.JsonSchema`

CanRead

Gets a value indicating whether this `Newtonsoft.Json.JsonConverter` `Newtonsoft.Json.JsonConverter`

CanWrite

Gets a value indicating whether this `Newtonsoft.Json.JsonConverter` `Newtonsoft.Json.JsonConverter`

`Newtonsoft.Json.Converters.BsonObjectIdConverter`. **WriteJson** (*Newtonsoft.Json.JsonWriter*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Writes the JSON representation of the object. :param writer: The

`Newtonsoft.Json.JsonWriter` :param value: The value.

param serializer The calling serializer.

`Newtonsoft.Json.Converters.BsonObjectIdConverter`. **ReadJson** (*Newtonsoft.Json.JsonReader*, *System.Type*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Reads the JSON representation of the object. :param reader: The

`Newtonsoft.Json.JsonReader` :param objectType: Type of the object.

param existingValue The existing value of object being read.

param serializer The calling serializer.

return The object value.

`Newtonsoft.Json.Converters.BsonObjectIdConverter`. **CanConvert** (*System.Type*)

Determines whether this instance can convert the specified object type. :param objectType: Type of the object. :return:

1.10 CustomCreationConverter`1

`Newtonsoft.Json.Converters.CustomCreationConverter`1`

Create a custom object

`Newtonsoft.Json.Converters.CustomCreationConverter`1`. **WriteJson** (*Newtonsoft.Json.Json*

Writes the JSON representation of the object. :param writer: The

Newtonsoft.Json.JsonWriter :param value: The value.

param serializer The calling serializer.

Newtonsoft.Json.Converters.CustomCreationConverter`1.ReadJson(Newtonsoft.Json.JsonF

Reads the JSON representation of the object. :param reader: The

Newtonsoft.Json.JsonReader :param objectType: Type of the object.

param existingValue The existing value of object being read.

param serializer The calling serializer.

return The object value.

Newtonsoft.Json.Converters.CustomCreationConverter`1.Create(System.Type)

Creates an object which will then be populated by the serializer. :param objectType: Type of the object.
:return: The created object.

Newtonsoft.Json.Converters.CustomCreationConverter`1.CanConvert(System.Type)

Determines whether this instance can convert the specified object type. :param objectType: Type of the object. :return:

Newtonsoft.Json.Converters.CustomCreationConverter`1.CanWrite

Gets a value indicating whether this [Newtonsoft.Json.JsonConverter](#)
[Newtonsoft.Json.JsonConverter](#)

1.11 DateTimeConverterBase

class [Newtonsoft.Json.Converters.DateTimeConverterBase](#)

Provides a base class for converting a [System.DateTime](#)

CanConvert (*System.Type*)

Determines whether this instance can convert the specified object type. :param objectType: Type of the object.
:return:

1.12 DiscriminatedUnionConverter

class [Newtonsoft.Json.Converters.DiscriminatedUnionConverter](#)

Converts a F# discriminated union type to and from JSON.

WriteJson (*Newtonsoft.Json.JsonWriter, System.Object, Newtonsoft.Json.JsonSerializer*)

Writes the JSON representation of the object. :param writer: The

Newtonsoft.Json.JsonWriter :param value: The value.

param serializer The calling serializer.

ReadJson (*Newtonsoft.Json.JsonReader, System.Type, System.Object, Newton-*
soft.Json.JsonSerializer)

Reads the JSON representation of the object. :param reader: The

Newtonsoft.Json.JsonReader :param objectType: Type of the object.

param existingValue The existing value of object being read.

param serializer The calling serializer.

return The object value.

CanConvert (*System.Type*)

Determines whether this instance can convert the specified object type. :param objectType: Type of the object. :return:

1.13 ExpandoObjectConverter

class `Newtonsoft.Json.Converters.ExpandoObjectConverter`

Converts an ExpandoObject to and from JSON.

WriteJson (*Newtonsoft.Json.JsonWriter*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Writes the JSON representation of the object. :param writer: The

Newtonsoft.Json.JsonWriter :param value: The value.

param serializer The calling serializer.

ReadJson (*Newtonsoft.Json.JsonReader*, *System.Type*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Reads the JSON representation of the object. :param reader: The

Newtonsoft.Json.JsonReader :param objectType: Type of the object.

param existingValue The existing value of object being read.

param serializer The calling serializer.

return The object value.

CanConvert (*System.Type*)

Determines whether this instance can convert the specified object type. :param objectType: Type of the object. :return:

CanWrite

Gets a value indicating whether this `Newtonsoft.Json.JsonConverter` `Newtonsoft.Json.JsonConverter`

1.14 IsoDateTimeConverter

class `Newtonsoft.Json.Converters.IsoDateTimeConverter`

Converts a `System.DateTime`

WriteJson (*Newtonsoft.Json.JsonWriter*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Writes the JSON representation of the object. :param writer: The

Newtonsoft.Json.JsonWriter :param value: The value.

param serializer The calling serializer.

ReadJson (*Newtonsoft.Json.JsonReader*, *System.Type*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Reads the JSON representation of the object. :param reader: The

Newtonsoft.Json.JsonReader :param objectType: Type of the object.

param existingValue The existing value of object being read.

param serializer The calling serializer.

return The object value.

DateTimeStyles

Gets or sets the date time styles used when converting a date to and from JSON.

DateTimeFormat

Gets or sets the date time format used when converting a date to and from JSON.

Culture

Gets or sets the culture used when converting a date to and from JSON.

1.15 JavaScriptDateTimeConverter

class `Newtonsoft.Json.Converters.JavaScriptDateTimeConverter`

Converts a `System.DateTime`

WriteJson (*Newtonsoft.Json.JsonWriter*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Writes the JSON representation of the object. :param writer: The

Newtonsoft.Json.JsonWriter :param value: The value.

param serializer The calling serializer.

ReadJson (*Newtonsoft.Json.JsonReader*, *System.Type*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Reads the JSON representation of the object. :param reader: The

Newtonsoft.Json.JsonReader :param objectType: Type of the object.

param existingValue The existing property value of the JSON that is being converted.

param serializer The calling serializer.

return The object value.

1.16 KeyValuePairConverter

class `Newtonsoft.Json.Converters.KeyValuePairConverter`

Converts a `System.Collections.Generic.KeyValuePair`2`

WriteJson (*Newtonsoft.Json.JsonWriter*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Writes the JSON representation of the object. :param writer: The

Newtonsoft.Json.JsonWriter :param value: The value.

param serializer The calling serializer.

ReadJson (*Newtonsoft.Json.JsonReader*, *System.Type*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Reads the JSON representation of the object. :param reader: The

Newtonsoft.Json.JsonReader :param objectType: Type of the object.

param existingValue The existing value of object being read.

param serializer The calling serializer.

return The object value.

CanConvert (*System.Type*)

Determines whether this instance can convert the specified object type. :param objectType: Type of the object. :return:

1.17 RegexConverter

class *Newtonsoft.Json.Converters.RegexConverter*

Converts a *System.Text.RegularExpressions.Regex*

WriteJson (*Newtonsoft.Json.JsonWriter*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Writes the JSON representation of the object. :param writer: The

Newtonsoft.Json.JsonWriter :param value: The value.

param serializer The calling serializer.

ReadJson (*Newtonsoft.Json.JsonReader*, *System.Type*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Reads the JSON representation of the object. :param reader: The

Newtonsoft.Json.JsonReader :param objectType: Type of the object.

param existingValue The existing value of object being read.

param serializer The calling serializer.

return The object value.

CanConvert (*System.Type*)

Determines whether this instance can convert the specified object type. :param objectType: Type of the object. :return:

1.18 StringEnumConverter

class *Newtonsoft.Json.Converters.StringEnumConverter*

Converts an *System.Enum*

Newtonsoft.Json.Converters.StringEnumConverter.#ctor

Initializes a new instance of the `Newtonsoft.Json.Converters.StringEnumConverter`

WriteJson (*Newtonsoft.Json.JsonWriter*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Writes the JSON representation of the object. :param writer: The

Newtonsoft.Json.JsonWriter :param value: The value.

param serializer The calling serializer.

ReadJson (*Newtonsoft.Json.JsonReader*, *System.Type*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Reads the JSON representation of the object. :param reader: The

Newtonsoft.Json.JsonReader :param objectType: Type of the object.

param existingValue The existing value of object being read.

param serializer The calling serializer.

return The object value.

CanConvert (*System.Type*)

Determines whether this instance can convert the specified object type. :param objectType: Type of the object. :return:

CamelCaseText

Gets or sets a value indicating whether the written enum text should be camel case.

AllowIntegerValues

Gets or sets a value indicating whether integer values are allowed.

1.19 VersionConverter

class `Newtonsoft.Json.Converters.VersionConverter`

Converts a `System.Version`

WriteJson (*Newtonsoft.Json.JsonWriter*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Writes the JSON representation of the object. :param writer: The

Newtonsoft.Json.JsonWriter :param value: The value.

param serializer The calling serializer.

ReadJson (*Newtonsoft.Json.JsonReader*, *System.Type*, *System.Object*, *Newtonsoft.Json.JsonSerializer*)

Reads the JSON representation of the object. :param reader: The

Newtonsoft.Json.JsonReader :param objectType: Type of the object.

param existingValue The existing property value of the JSON that is being converted.

param serializer The calling serializer.

return The object value.

CanConvert (*System.Type*)

Determines whether this instance can convert the specified object type. :param objectType: Type of the object. :return:

1.20 XmlNodeConverter

class `Newtonsoft.Json.Converters.XmlNodeConverter`

Converts XML to and from JSON.

WriteJson (*Newtonsoft.Json.JsonWriter, System.Object, Newtonsoft.Json.JsonSerializer*)

Writes the JSON representation of the object. :param writer: The

Newtonsoft.Json.JsonWriter :param serializer: The calling serializer.

param value The value.

ReadJson (*Newtonsoft.Json.JsonReader, System.Type, System.Object, Newtonsoft.Json.JsonSerializer*)

Reads the JSON representation of the object. :param reader: The

Newtonsoft.Json.JsonReader :param objectType: Type of the object.

param existingValue The existing value of object being read.

param serializer The calling serializer.

return The object value.

IsNamespaceAttribute (*System.String, System.String@*)

Checks if the attributeName is a namespace attribute. :param attributeName: Attribute name to test. :param prefix: The attribute name prefix if it has one, otherwise an empty string. :return: True if attribute name is for a namespace attribute, otherwise false.

CanConvert (*System.Type*)

Determines whether this instance can convert the specified value type. :param valueType: Type of the value. :return:

DeserializeRootElementName

Gets or sets the name of the root element to insert when deserializing to XML if the JSON structure has produces multiple root elements.

WriteArrayAttribute

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

OmitRootObject

Gets or sets a value indicating whether to write the root JSON object.

1.21 DateFormatHandling

class `Newtonsoft.Json.DateFormatHandling`

Specifies how dates are formatted when writing JSON text.

ISODateFormat

Dates are written in the ISO 8601 format, e.g. “2012-03-21T05:40Z”.

MicrosoftDateFormat

Dates are written in the Microsoft JSON format, e.g. “/Date(1198908717056)/”.

1.22 DateParseHandling

class `Newtonsoft.Json.DateParseHandling`

Specifies how date formatted strings, e.g. “/Date(1198908717056)/” and “2012-03-21T05:40Z”, are parsed when reading JSON text.

None

Date formatted strings are not parsed to a date type and are read as strings.

DateTime

Date formatted strings, e.g. “/Date(1198908717056)/” and “2012-03-21T05:40Z”, are parsed to `Newtonsoft.Json.DateParseHandling.DateTime`

DateTimeOffset

Date formatted strings, e.g. “/Date(1198908717056)/” and “2012-03-21T05:40Z”, are parsed to `Newtonsoft.Json.DateParseHandling.DateTimeOffset`

1.23 DateTimeZoneHandling

class `Newtonsoft.Json.DateTimeZoneHandling`

Specifies how to treat the time value when converting between string and `System.DateTime`

Local

Treat as local time. If the `System.DateTime`

Utc

Treat as a UTC. If the `System.DateTime`

Unspecified

Treat as a local time if a `System.DateTime` `System.DateTime`

RoundtripKind

Time zone information should be preserved when converting.

1.24 DefaultValueHandling

class `Newtonsoft.Json.DefaultValueHandling`

Specifies default value handling options for the `Newtonsoft.Json.JsonSerializer`

Include

Include members where the member value is the same as the member’s default value when serializing objects.
Included members are written to JSON. Has no effect when deserializing.

Ignore

Ignore members where the member value is the same as the member's default value when serializing objects so that is is not written to JSON. This option will ignore all default values (e.g. `System.ComponentModel.DefaultValueAttribute`)

Populate

Members with a default value but no JSON will be set to their default value when deserializing.

IgnoreAndPopulate

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

1.25 FloatFormatHandling

class `Newtonsoft.Json.FloatFormatHandling`

Specifies float format handling options when writing special floating point numbers, e.g. `System.Double.NaN` `System.Double.PositiveInfinity` `System.Double.NegativeInfinity` `Newtonsoft.Json.JsonWriter`

String

Write special floating point values as strings in JSON, e.g. "NaN", "Infinity", "-Infinity".

Symbol

Write special floating point values as symbols in JSON, e.g. NaN, Infinity, -Infinity. Note that this will produce non-valid JSON.

DefaultValue

Write special floating point values as the property's default value in JSON, e.g. 0.0 for a `System.Double` `System.Nullable`1`

1.26 FloatParseHandling

class `Newtonsoft.Json.FloatParseHandling`

Specifies how floating point numbers, e.g. 1.0 and 9.9, are parsed when reading JSON text.

Double

Floating point numbers are parsed to `Newtonsoft.Json.FloatParseHandling.Double`

Decimal

Floating point numbers are parsed to `Newtonsoft.Json.FloatParseHandling.Decimal`

1.27 FormatterAssemblyStyle

class `System.Runtime.Serialization.Formatters.FormatterAssemblyStyle`

Indicates the method that will be used during deserialization for locating and loading assemblies.

Simple

In simple mode, the assembly used during deserialization need not match exactly the assembly used during serialization. Specifically, the version numbers need not match as the `LoadWithPartialName` method is used to load the assembly.

Full

In full mode, the assembly used during deserialization must match exactly the assembly used during serialization. The `Load` method of the `Assembly` class is used to load the assembly.

1.28 Formatting

class `Newtonsoft.Json.Formatting`

Specifies formatting options for the `Newtonsoft.Json.JsonTextWriter`

None

No special formatting is applied. This is the default.

Indented

Causes child objects to be indented according to the `Newtonsoft.Json.JsonTextWriter.Indentation` `Newtonsoft.Json.JsonTextWriter.IndentChar`

1.29 IJsonLineInfo

class `Newtonsoft.Json.IJsonLineInfo`

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

HasLineInfo()

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

LineNumber

Gets the current line number.

LinePosition

Gets the current line position.

1.30 JsonArrayAttribute

class `Newtonsoft.Json.JsonArrayAttribute`

Instructs the `Newtonsoft.Json.JsonSerializer`

1.31 JsonContainerAttribute

class `Newtonsoft.Json.JsonContainerAttribute`

Instructs the `Newtonsoft.Json.JsonSerializer`

Newtonsoft.Json.JsonContainerAttribute.#ctor

Initializes a new instance of the `Newtonsoft.Json.JsonContainerAttribute`

Newtonsoft.Json.JsonContainerAttribute.#ctor(System.String)

Initializes a new instance of the `Newtonsoft.Json.JsonContainerAttribute` :param id: The container Id.

Id

Gets or sets the id.

Title

Gets or sets the title.

Description

Gets or sets the description.

ItemConverterType

Gets the collection's items converter.

ItemConverterParameters

The parameter list to use when constructing the `JsonConverter` described by `ItemConverterType`. If null, the default constructor is used. When non-null, there must be a constructor defined in the `JsonConverter` that exactly matches the number, order, and type of these parameters.

IsReference

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

ItemIsReference

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

ItemReferenceLoopHandling

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

ItemTypeNameHandling

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

Newtonsoft.Json.JsonArrayAttribute.#ctor

Initializes a new instance of the `Newtonsoft.Json.JsonArrayAttribute`

Newtonsoft.Json.JsonArrayAttribute.#ctor(System.Boolean)

Initializes a new instance of the `Newtonsoft.Json.JsonObjectAttribute` :param allowNullItems: A flag indicating whether the array can contain null items.

Newtonsoft.Json.JsonArrayAttribute.#ctor(System.String)

Initializes a new instance of the `Newtonsoft.Json.JsonArrayAttribute` :param id: The container Id.

Newtonsoft.Json.JsonArrayAttribute.AllowNullItems

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

1.32 JsonConstructorAttribute

class `Newtonsoft.Json.JsonConstructorAttribute`

Instructs the `Newtonsoft.Json.JsonSerializer`

1.33 JsonConvert

class `Newtonsoft.Json.JsonConvert`

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

True

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

False

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

Null

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

Undefined

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

PositiveInfinity

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

NegativeInfinity

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

NaN

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

ToString (*System.DateTime*)

Converts the `System.DateTime` :param value: The value to convert. :return: A JSON string representation of the `System.DateTime`

ToString (*System.DateTime*, *Newtonsoft.Json.DateFormatHandling*, *Newtonsoft.Json.DateTimeZoneHandling*)

Converts the `System.DateTime` `Newtonsoft.Json.DateFormatHandling` :param value: The value to convert. :param format: The format the date will be converted to. :param timeZoneHandling: The time zone handling when the date is converted to a string. :return: A JSON string representation of the `System.DateTime`

ToString (*System.DateTimeOffset*)

Converts the `System.DateTimeOffset` :param value: The value to convert. :return: A JSON string representation of the `System.DateTimeOffset`

ToString (*System.DateTimeOffset*, *Newtonsoft.Json.DateFormatHandling*)

Converts the `System.DateTimeOffset` `Newtonsoft.Json.DateFormatHandling` :param value: The value to convert. :param format: The format the date will be converted to. :return: A JSON string representation of the `System.DateTimeOffset`

ToString (*System.Boolean*)

Converts the `System.Boolean` :param value: The value to convert. :return: A JSON string representation of the `System.Boolean`

ToString (*System.Char*)

Converts the `System.Char` :param value: The value to convert. :return: A JSON string representation of the `System.Char`

ToString (*System.Enum*)

Converts the `System.Enum` :param value: The value to convert. :return: A JSON string representation of the `System.Enum`

ToString (*System.Int32*)

Converts the `System.Int32` :param value: The value to convert. :return: A JSON string representation of the `System.Int32`

ToString (*System.Int16*)

Converts the `System.Int16` :param value: The value to convert. :return: A JSON string representation of the `System.Int16`

ToString (*System.UInt16*)

Converts the `System.UInt16` :param value: The value to convert. :return: A JSON string representation of the `System.UInt16`

ToString (*System.UInt32*)

Converts the `System.UInt32` :param value: The value to convert. :return: A JSON string representation of the `System.UInt32`

ToString (*System.Int64*)

Converts the `System.Int64` :param value: The value to convert. :return: A JSON string representation of the `System.Int64`

ToString (*System.UInt64*)

Converts the `System.UInt64` :param value: The value to convert. :return: A JSON string representation of the `System.UInt64`

ToString (*System.Single*)

Converts the `System.Single` :param value: The value to convert. :return: A JSON string representation of the `System.Single`

ToString (*System.Double*)

Converts the `System.Double` :param value: The value to convert. :return: A JSON string representation of the `System.Double`

ToString (*System.Byte*)

Converts the `System.Byte` :param value: The value to convert. :return: A JSON string representation of the `System.Byte`

ToString (*System.SByte*)

Converts the `System.SByte` :param value: The value to convert. :return: A JSON string representation of the `System.SByte`

ToString (*System.Decimal*)

Converts the `System.Decimal` :param value: The value to convert. :return: A JSON string representation of the `System.SByte`

ToString (*System.Guid*)

Converts the `System.Guid` :param value: The value to convert. :return: A JSON string representation of the `System.Guid`

ToString (*System.TimeSpan*)

Converts the `System.TimeSpan` :param value: The value to convert. :return: A JSON string representation of the `System.TimeSpan`

ToString (*System.Uri*)

Converts the `System.Uri` :param value: The value to convert. :return: A JSON string representation of the `System.Uri`

ToString (*System.String*)

Converts the `System.String` :param value: The value to convert. :return: A JSON string representation of the `System.String`

ToString (*System.String, System.Char*)

Converts the `System.String` :param value: The value to convert. :param delimiter: The string delimiter character. :return: A JSON string representation of the `System.String`

ToString (*System.String, System.Char, Newtonsoft.Json.StringEscapeHandling*)

Converts the `System.String` :param value: The value to convert. :param delimiter: The string delimiter character. :param stringEscapeHandling: The string escape handling. :return: A JSON string representation of the `System.String`

ToString (*System.Object*)

Converts the `System.Object` :param value: The value to convert. :return: A JSON string representation of the `System.Object`

SerializeObject (*System.Object*)

Serializes the specified object to a JSON string. :param value: The object to serialize. :return: A JSON string representation of the object.

SerializeObject (*System.Object, Newtonsoft.Json.Formatting*)

Serializes the specified object to a JSON string using formatting. :param value: The object to serialize. :param formatting: Indicates how the output is formatted. :return: A JSON string representation of the object.

SerializeObject (*System.Object, Newtonsoft.Json.JsonConverter[]*)

Serializes the specified object to a JSON string using a collection of `Newtonsoft.Json.JsonConverter` :param value: The object to serialize. :param converters: A collection converters used while serializing. :return: A JSON string representation of the object.

SerializeObject (*System.Object, Newtonsoft.Json.Formatting, Newtonsoft.Json.JsonConverter[]*)

Serializes the specified object to a JSON string using formatting and a collection of `Newtonsoft.Json.JsonConverter` :param value: The object to serialize. :param formatting: Indicates how the output is formatted. :param converters: A collection converters used while serializing. :return: A JSON string representation of the object.

SerializeObject (*System.Object, Newtonsoft.Json.JsonSerializerSettings*)

Serializes the specified object to a JSON string using `Newtonsoft.Json.JsonSerializerSettings` :param value: The object to serialize. :param settings: The

`Newtonsoft.Json.JsonSerializerSettings` :return: A JSON string representation of the object.

SerializeObject (*System.Object, System.Type, Newtonsoft.Json.JsonSerializerSettings*)

Serializes the specified object to a JSON string using a type, formatting and `Newtonsoft.Json.JsonSerializerSettings` :param value: The object to serialize. :param settings: The

Newtonsoft.Json.JsonSerializerSettings :param type: The type of the value being serialized. This parameter is used when

`Newtonsoft.Json.TypeNameHandling` :return: A JSON string representation of the object.

SerializeObject (*System.Object*, *Newtonsoft.Json.Formatting*, *Newtonsoft.Json.JsonSerializerSettings*)

Serializes the specified object to a JSON string using formatting and `Newtonsoft.Json.JsonSerializerSettings` :param value: The object to serialize. :param formatting: Indicates how the output is formatted. :param settings: The

`Newtonsoft.Json.JsonSerializerSettings` :return: A JSON string representation of the object.

SerializeObject (*System.Object*, *System.Type*, *Newtonsoft.Json.Formatting*, *Newtonsoft.Json.JsonSerializerSettings*)

Serializes the specified object to a JSON string using a type, formatting and `Newtonsoft.Json.JsonSerializerSettings` :param value: The object to serialize. :param formatting: Indicates how the output is formatted. :param settings: The

Newtonsoft.Json.JsonSerializerSettings :param type: The type of the value being serialized. This parameter is used when

`Newtonsoft.Json.TypeNameHandling` :return: A JSON string representation of the object.

SerializeObjectAsync (*System.Object*)

Asynchronously serializes the specified object to a JSON string. Serialization will happen on a new thread. :param value: The object to serialize.

return A task that represents the asynchronous serialize operation. The value of the

SerializeObjectAsync (*System.Object*, *Newtonsoft.Json.Formatting*)

Asynchronously serializes the specified object to a JSON string using formatting. Serialization will happen on a new thread. :param value: The object to serialize.

param formatting Indicates how the output is formatted.

return A task that represents the asynchronous serialize operation. The value of the

SerializeObjectAsync (*System.Object*, *Newtonsoft.Json.Formatting*, *Newtonsoft.Json.JsonSerializerSettings*)

Asynchronously serializes the specified object to a JSON string using formatting and a collection of `Newtonsoft.Json.JsonConverter` :param value: The object to serialize. :param formatting: Indicates how the output is formatted. :param settings: The

`Newtonsoft.Json.JsonSerializerSettings` :return: A task that represents the asynchronous serialize operation. The value of the

DeserializeObject (*System.String*)

Deserializes the JSON to a .NET object. :param value: The JSON to deserialize. :return: The deserialized object from the JSON string.

DeserializeObject (*System.String*, *Newtonsoft.Json.JsonSerializerSettings*)

Deserializes the JSON to a .NET object using *Newtonsoft.Json.JsonSerializerSettings*
:param value: The JSON to deserialize. :param settings:

The

Newtonsoft.Json.JsonSerializerSettings :return: The deserialized object from the JSON string.

DeserializeObject (*System.String*, *System.Type*)

Deserializes the JSON to the specified .NET type. :param value: The JSON to deserialize. :param type: The

System.Type :return: The deserialized object from the JSON string.

Newtonsoft.Json.JsonConvert.DeserializeObject `1(*System.String*)

Deserializes the JSON to the specified .NET type. :param value: The JSON to deserialize. :return: The deserialized object from the JSON string.

Newtonsoft.Json.JsonConvert.DeserializeAnonymousType `1(*System.String*, ``0)

Deserializes the JSON to the given anonymous type. :param value: The JSON to deserialize. :param anonymousTypeObject: The anonymous type object. :return: The deserialized anonymous type from the JSON string.

Newtonsoft.Json.JsonConvert.DeserializeAnonymousType `1(*System.String*, ``0, *Newtonsoft.Json.JsonSerializerSettings*)

Deserializes the JSON to the given anonymous type using *Newtonsoft.Json.JsonSerializerSettings* :param value: The JSON to deserialize. :param anonymousTypeObject: The anonymous type object. :param settings:

The

Newtonsoft.Json.JsonSerializerSettings :return: The deserialized anonymous type from the JSON string.

Newtonsoft.Json.JsonConvert.DeserializeObject `1(*System.String*, *Newtonsoft.Json.JsonConverter[]*)

Deserializes the JSON to the specified .NET type using a collection of *Newtonsoft.Json.JsonConverter* :param value: The JSON to deserialize. :param converters: Converters to use while deserializing. :return: The deserialized object from the JSON string.

Newtonsoft.Json.JsonConvert.DeserializeObject `1(*System.String*, *Newtonsoft.Json.JsonConverter[]*, *Newtonsoft.Json.JsonSerializerSettings*)

Deserializes the JSON to the specified .NET type using *Newtonsoft.Json.JsonSerializerSettings* :param value: The object to deserialize. :param settings:

The

Newtonsoft.Json.JsonSerializerSettings :return: The deserialized object from the JSON string.

DeserializeObject (*System.String*, *System.Type*, *Newtonsoft.Json.JsonConverter[]*)

Deserializes the JSON to the specified .NET type using a collection of *Newtonsoft.Json.JsonConverter* :param value: The JSON to deserialize. :param type: The type of the object to deserialize. :param converters: Converters to use while deserializing. :return: The deserialized object from the JSON string.

DeserializeObject (*System.String*, *System.Type*, *Newtonsoft.Json.JsonConverter[]*, *Newtonsoft.Json.JsonSerializerSettings*)

Deserializes the JSON to the specified .NET type using `Newtonsoft.Json.JsonSerializerSettings` :param value: The JSON to deserialize. :param type: The type of the object to deserialize to. :param settings:

The

`Newtonsoft.Json.JsonSerializerSettings` :return: The deserialized object from the JSON string.

Newtonsoft.Json.JsonConvert.DeserializeObjectAsync `1(*System.String*)

Asynchronously deserializes the JSON to the specified .NET type. Deserialization will happen on a new thread. :param value: The JSON to deserialize.

return A task that represents the asynchronous deserialize operation. The value of the

Newtonsoft.Json.JsonConvert.DeserializeObjectAsync `1(*System.String*, *Newtonsoft.Json*

Asynchronously deserializes the JSON to the specified .NET type using `Newtonsoft.Json.JsonSerializerSettings` :param value: The JSON to deserialize. :param settings:

The

`Newtonsoft.Json.JsonSerializerSettings` :return: A task that represents the asynchronous deserialize operation. The value of the

DeserializeObjectAsync (*System.String*)

Asynchronously deserializes the JSON to the specified .NET type. Deserialization will happen on a new thread. :param value: The JSON to deserialize.

return A task that represents the asynchronous deserialize operation. The value of the

DeserializeObjectAsync (*System.String*, *System.Type*, *Newtonsoft.Json.JsonSerializerSettings*)

Asynchronously deserializes the JSON to the specified .NET type using `Newtonsoft.Json.JsonSerializerSettings` :param value: The JSON to deserialize. :param type: The type of the object to deserialize to. :param settings:

The

`Newtonsoft.Json.JsonSerializerSettings` :return: A task that represents the asynchronous deserialize operation. The value of the

PopulateObject (*System.String*, *System.Object*)

Populates the object with values from the JSON string. :param value: The JSON to populate values from. :param target: The target object to populate values onto.

PopulateObject (*System.String*, *System.Object*, *Newtonsoft.Json.JsonSerializerSettings*)

Populates the object with values from the JSON string using `Newtonsoft.Json.JsonSerializerSettings` :param value: The JSON to populate values from. :param target: The target object to populate values onto. :param settings:

The

`Newtonsoft.Json.JsonSerializerSettings`

PopulateObjectAsync (*System.String*, *System.Object*, *Newtonsoft.Json.JsonSerializerSettings*)

Asynchronously populates the object with values from the JSON string using `Newtonsoft.Json.JsonSerializerSettings` :param value: The JSON to populate values from. :param target: The target object to populate values onto. :param settings:

The

`Newtonsoft.Json.JsonSerializerSettings` :return: A task that represents the asynchronous populate operation.

SerializeXmlNode (*System.Xml.Linq.XObject*)

Serializes the `System.Xml.Linq.XNode` :param node: The node to convert to JSON. :return: A JSON string of the XNode.

SerializeXmlNode (*System.Xml.Linq.XObject, Newtonsoft.Json.Formatting*)

Serializes the `System.Xml.Linq.XNode` :param node: The node to convert to JSON. :param formatting: Indicates how the output is formatted. :return: A JSON string of the XNode.

SerializeXmlNode (*System.Xml.Linq.XObject, Newtonsoft.Json.Formatting, System.Boolean*)

Serializes the `System.Xml.Linq.XNode` :param node: The node to serialize. :param formatting: Indicates how the output is formatted. :param omitRootObject: Omits writing the root object. :return: A JSON string of the XNode.

DeserializeXmlNode (*System.String*)

Deserializes the `System.Xml.Linq.XNode` :param value: The JSON string. :return: The deserialized XNode

DeserializeXmlNode (*System.String, System.String*)

Deserializes the `System.Xml.Linq.XNode` :param value: The JSON string. :param deserializeRootElementName: The name of the root element to append when deserializing. :return: The deserialized XNode

DeserializeXmlNode (*System.String, System.String, System.Boolean*)

Deserializes the `System.Xml.Linq.XNode` :param value: The JSON string. :param deserializeRootElementName: The name of the root element to append when deserializing. :param writeArrayAttribute:

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 The deserialized XNode

DefaultSettings

Gets or sets a function that creates default `Newtonsoft.Json.JsonSerializerSettings`
`Newtonsoft.Json.JsonConvert` `Newtonsoft.Json.Linq.JToken.ToObject`1`
`Newtonsoft.Json.Linq.JToken.FromObject(System.Object)`
`Newtonsoft.Json.Linq.JToken` `Newtonsoft.Json.JsonSerializer`
`Newtonsoft.Json.JsonSerializer.Create`

1.34 JsonConvertAttribute

class `Newtonsoft.Json.JsonConverterAttribute`

Instructs the `Newtonsoft.Json.JsonSerializer` `Newtonsoft.Json.JsonConverter`

Newtonsoft.Json.JsonConverterAttribute.#ctor(System.Type)

Initializes a new instance of the `Newtonsoft.Json.JsonConverterAttribute` :param converterType: Type of the converter.

Newtonsoft.Json.JsonConverterAttribute.#ctor(System.Type, System.Object[])

Initializes a new instance of the `Newtonsoft.Json.JsonConverterAttribute` :param converterType: Type of the converter. :param converterParameters: Parameter list to use when constructing the `JsonConverter`. Can be null.

ConverterType

Gets the type of the converter.

ConverterParameters

The parameter list to use when constructing the `JsonConverter` described by `ConverterType`. If null, the default constructor is used.

1.35 JsonConvertCollection

class Newtonsoft.Json.JsonConvertCollection

Represents a collection of `Newtonsoft.Json.JsonConverter`

1.36 JsonDictionaryAttribute

class Newtonsoft.Json.JsonDictionaryAttribute

Instructs the `Newtonsoft.Json.JsonSerializer`

Newtonsoft.Json.JsonDictionaryAttribute.#ctor

Initializes a new instance of the `Newtonsoft.Json.JsonDictionaryAttribute`

Newtonsoft.Json.JsonDictionaryAttribute.#ctor(System.String)

Initializes a new instance of the `Newtonsoft.Json.JsonDictionaryAttribute` :param id: The container Id.

1.37 JsonException

class Newtonsoft.Json.JsonException

The exception thrown when an error occurs during Json serialization or deserialization.

Newtonsoft.Json.JsonException.#ctor

Initializes a new instance of the `Newtonsoft.Json.JsonException`

Newtonsoft.Json.JsonException.#ctor(System.String)

Initializes a new instance of the `Newtonsoft.Json.JsonException` :param message: The error message that explains the reason for the exception.

Newtonsoft.Json.JsonException.#ctor(System.String, System.Exception)

Initializes a new instance of the `Newtonsoft.Json.JsonException` :param message: The error message that explains the reason for the exception. :param innerException: The exception that is the cause of the current exception, or a null reference (Nothing in Visual Basic) if no inner exception is specified.

1.38 JsonExtensionDataAttribute

class `Newtonsoft.Json.JsonExtensionDataAttribute`

Instructs the `Newtonsoft.Json.JsonSerializer`

Newtonsoft.Json.JsonExtensionDataAttribute.#ctor

Initializes a new instance of the `Newtonsoft.Json.JsonExtensionDataAttribute`

WriteData

Gets or sets a value that indicates whether to write extension data when serializing the object.

ReadData

Gets or sets a value that indicates whether to read extension data when deserializing the object.

1.39 JsonIgnoreAttribute

class `Newtonsoft.Json.JsonIgnoreAttribute`

Instructs the `Newtonsoft.Json.JsonSerializer`

1.40 JsonObjectAttribute

class `Newtonsoft.Json.JsonObjectAttribute`

Instructs the `Newtonsoft.Json.JsonSerializer`

Newtonsoft.Json.JsonObjectAttribute.#ctor

Initializes a new instance of the `Newtonsoft.Json.JsonObjectAttribute`

Newtonsoft.Json.JsonObjectAttribute.#ctor(Newtonsoft.Json.MemberSerialization)

Initializes a new instance of the `Newtonsoft.Json.JsonObjectAttribute` :param memberSerialization: The member serialization.

Newtonsoft.Json.JsonObjectAttribute.#ctor(System.String)

Initializes a new instance of the `Newtonsoft.Json.JsonObjectAttribute` :param id: The container Id.

MemberSerialization

Gets or sets the member serialization.

ItemRequired

Gets or sets a value that indicates whether the object's properties are required.

1.41 JsonPropertyAttribute

class `Newtonsoft.Json.JsonPropertyAttribute`

Instructs the `Newtonsoft.Json.JsonSerializer`

Newtonsoft.Json.JsonPropertyAttribute.#ctor

Initializes a new instance of the `Newtonsoft.Json.JsonPropertyAttribute`

Newtonsoft.Json.JsonPropertyAttribute.#ctor(System.String)

Initializes a new instance of the `Newtonsoft.Json.JsonPropertyAttribute` :param propertyName: Name of the property.

ItemConverterType

Gets or sets the converter used when serializing the property's collection items.

ItemConverterParameters

The parameter list to use when constructing the JsonConverter described by ItemConverterType. If null, the default constructor is used. When non-null, there must be a constructor defined in the `JsonConverter` that exactly matches the number, order, and type of these parameters.

NullValueHandling

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

DefaultValueHandling

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

ReferenceLoopHandling

Gets or sets the reference loop handling used when serializing this property.

ObjectCreationHandling

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

TypeNameHandling

Gets or sets the type name handling used when serializing this property.

IsReference

Gets or sets whether this property's value is serialized as a reference.

Order

Gets or sets the order of serialization and deserialization of a member.

Required

Gets or sets a value indicating whether this property is required.

PropertyName

Gets or sets the name of the property.

ItemReferenceLoopHandling

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

ItemTypeNameHandling

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

ItemIsReference

Gets or sets whether this property's collection items are serialized as a reference.

1.42 JsonSerializerException

class `Newtonsoft.Json.JsonReaderException`

The exception thrown when an error occurs while reading Json text.

Newtonsoft.Json.JsonReaderException.#ctor

Initializes a new instance of the `Newtonsoft.Json.JsonReaderException`

Newtonsoft.Json.JsonReaderException.#ctor(System.String)

Initializes a new instance of the `Newtonsoft.Json.JsonReaderException` :param message: The error message that explains the reason for the exception.

Newtonsoft.Json.JsonReaderException.#ctor(System.String, System.Exception)

Initializes a new instance of the `Newtonsoft.Json.JsonReaderException` :param message: The error message that explains the reason for the exception. :param innerException: The exception that is the cause of the current exception, or a null reference (Nothing in Visual Basic) if no inner exception is specified.

LineNumber

Gets the line number indicating where the error occurred.

LinePosition

Gets the line position indicating where the error occurred.

Path

Gets the path to the JSON where the error occurred.

1.43 JsonSerializerException

class `Newtonsoft.Json.JsonSerializationException`

The exception thrown when an error occurs during Json serialization or deserialization.

Newtonsoft.Json.JsonSerializationException.#ctor

Initializes a new instance of the `Newtonsoft.Json.JsonSerializationException`

Newtonsoft.Json.JsonSerializationException.#ctor(System.String)

Initializes a new instance of the `Newtonsoft.Json.JsonSerializationException` :param message: The error message that explains the reason for the exception.

Newtonsoft.Json.JsonSerializationException.#ctor(System.String, System.Exception)

Initializes a new instance of the `Newtonsoft.Json.JsonSerializationException` :param message: The error message that explains the reason for the exception. :param innerException: The exception that is the cause of the current exception, or a null reference (Nothing in Visual Basic) if no inner exception is specified.

1.44 JsonSerializer

class `Newtonsoft.Json.JsonSerializer`

Serializes and deserializes objects into and from the JSON format. The

`Newtonsoft.Json.JsonSerializer`

`Newtonsoft.Json.JsonSerializer.#ctor`

Initializes a new instance of the `Newtonsoft.Json.JsonSerializer`

`Create()`

Creates a new `Newtonsoft.Json.JsonSerializer` `Newtonsoft.Json.JsonSerializer`

:return: A new `Newtonsoft.Json.JsonSerializer` `Newtonsoft.Json.JsonSerializer`

`Create(Newtonsoft.Json.JsonSerializerSettings)`

Creates a new `Newtonsoft.Json.JsonSerializer` `Newtonsoft.Json.JsonSerializerSettings`

`Newtonsoft.Json.JsonSerializer` :param settings: The settings to be applied to the

`Newtonsoft.Json.JsonSerializer` :return: A new `Newtonsoft.Json.JsonSerializer`

`Newtonsoft.Json.JsonSerializerSettings` `Newtonsoft.Json.JsonSerializer`

`CreateDefault()`

Creates a new `Newtonsoft.Json.JsonSerializer` `Newtonsoft.Json.JsonSerializer`

:return: A new `Newtonsoft.Json.JsonSerializer` `Newtonsoft.Json.JsonSerializer`

`CreateDefault(Newtonsoft.Json.JsonSerializerSettings)`

Creates a new `Newtonsoft.Json.JsonSerializer` `Newtonsoft.Json.JsonSerializerSettings`

`Newtonsoft.Json.JsonSerializer` :param settings: The settings to be applied to the

`Newtonsoft.Json.JsonSerializer` :return: A new `Newtonsoft.Json.JsonSerializer`

`Newtonsoft.Json.JsonSerializerSettings` `Newtonsoft.Json.JsonSerializer`

`Populate(System.IO.TextReader, System.Object)`

Populates the JSON values onto the target object. :param reader: The

`System.IO.TextReader` :param target: The target object to populate values onto.

`Populate(Newtonsoft.Json.JsonReader, System.Object)`

Populates the JSON values onto the target object. :param reader: The

`Newtonsoft.Json.JsonReader` :param target: The target object to populate values onto.

`Deserialize(Newtonsoft.Json.JsonReader)`

Deserializes the Json structure contained by the specified `Newtonsoft.Json.JsonReader`

:param reader: The

`Newtonsoft.Json.JsonReader` :return: The `System.Object`

`Deserialize(System.IO.TextReader, System.Type)`

Deserializes the Json structure contained by the specified `System.IO.StringReader` :param reader: The

`System.IO.TextReader` :param objectType: The `System.Type` :return: The instance of

Newtonsoft.Json.JsonSerializer.Deserialize`1(Newtonsoft.Json.JsonReader)

Deserializes the Json structure contained by the specified `Newtonsoft.Json.JsonReader`
:param reader: The

`Newtonsoft.Json.JsonReader` :return: The instance of

Deserialize (*Newtonsoft.Json.JsonReader, System.Type*)

Deserializes the Json structure contained by the specified `Newtonsoft.Json.JsonReader`
:param reader: The

`Newtonsoft.Json.JsonReader` :param objectType: The `System.Type` :return: The instance of

Serialize (*System.IO.TextWriter, System.Object*)

Serializes the specified `System.Object` `System.IO.TextWriter` :param textWriter: The

`System.IO.TextWriter` :param value: The `System.Object`

Serialize (*Newtonsoft.Json.JsonWriter, System.Object, System.Type*)

Serializes the specified `System.Object` `System.IO.TextWriter` :param jsonWriter: The

`Newtonsoft.Json.JsonWriter` :param value: The `System.Object` :param objectType:

The type of the value being serialized. This parameter is used when

`Newtonsoft.Json.JsonSerializer.TypeNameHandling`

Serialize (*System.IO.TextWriter, System.Object, System.Type*)

Serializes the specified `System.Object` `System.IO.TextWriter` :param textWriter: The

`System.IO.TextWriter` :param value: The `System.Object` :param objectType:

The type of the value being serialized. This parameter is used when

`Newtonsoft.Json.JsonSerializer.TypeNameHandling`

Serialize (*Newtonsoft.Json.JsonWriter, System.Object*)

Serializes the specified `System.Object` `Newtonsoft.Json.JsonWriter` :param json-
Writer: The

`Newtonsoft.Json.JsonWriter` :param value: The `System.Object`

Occurs when the `Newtonsoft.Json.JsonSerializer`

ReferenceResolver

Gets or sets the `Newtonsoft.Json.Serialization.IReferenceResolver`

Binder

Gets or sets the `Newtonsoft.Json.SerializationBinder`

TraceWriter

Gets or sets the `Newtonsoft.Json.Serialization.ITraceWriter`

TypeNameHandling

Gets or sets how type name writing and reading is handled by the serializer.

TypeNameAssemblyFormat

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

PreserveReferencesHandling

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

ReferenceLoopHandling

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

MissingMemberHandling

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

NullValueHandling

Get or set how null values are handled during serialization and deserialization.

DefaultValueHandling

Get or set how null default are handled during serialization and deserialization.

ObjectCreationHandling

Gets or sets how objects are created during deserialization.

ConstructorHandling

Gets or sets how constructors are used during deserialization.

MetadataPropertyHandling

Gets or sets how metadata properties are used during deserialization.

Converters

Gets a collection `Newtonsoft.Json.JsonConverter` `Newtonsoft.Json.JsonConverter`

ContractResolver

Gets or sets the contract resolver used by the serializer when serializing .NET objects to JSON and vice versa.

Context

Gets or sets the `System.Runtime.Serialization.StreamingContext`

Formatting

Indicates how JSON text output is formatted.

DateFormatHandling

Get or set how dates are written to JSON text.

DateTimeZoneHandling

Get or set how `System.DateTime`

DateParseHandling

Get or set how date formatted strings, e.g. `"/Date(1198908717056)/"` and `"2012-03-21T05:40Z"`, are parsed when reading JSON.

FloatParseHandling

Get or set how floating point numbers, e.g. 1.0 and 9.9, are parsed when reading JSON text.

FloatFormatHandling

Get or set how special floating point numbers, e.g. `System.Double.NaN`
`System.Double.PositiveInfinity` `System.Double.NegativeInfinity`

StringEscapeHandling

Get or set how strings are escaped when writing JSON text.

DateFormatString

Get or set how `System.DateTime` `System.DateTimeOffset`

Culture

Gets or sets the culture used when reading JSON. Defaults to `System.Globalization.CultureInfo.InvariantCulture`

MaxDepth

Gets or sets the maximum depth allowed when reading JSON. Reading past this depth will throw a `Newtonsoft.Json.JsonReaderException`

CheckAdditionalContent

Gets a value indicating whether there will be a check for additional JSON content after deserializing an object.

1.45 JsonSerializerSettings

class `Newtonsoft.Json.JsonSerializerSettings`

Specifies the settings on a `Newtonsoft.Json.JsonSerializer`

Newtonsoft.Json.JsonSerializerSettings.#ctor

Initializes a new instance of the `Newtonsoft.Json.JsonSerializerSettings`

ReferenceLoopHandling

Gets or sets how reference loops (e.g. a class referencing itself) is handled.

MissingMemberHandling

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

ObjectCreationHandling

Gets or sets how objects are created during deserialization.

NullValueHandling

Gets or sets how null values are handled during serialization and deserialization.

DefaultValueHandling

Gets or sets how null default are handled during serialization and deserialization.

Converters

Gets or sets a collection `Newtonsoft.Json.JsonConverter`

PreserveReferencesHandling

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

TypeNameHandling

Gets or sets how type name writing and reading is handled by the serializer.

MetadataPropertyHandling

Gets or sets how metadata properties are used during deserialization.

TypeNameAssemblyFormat

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

ConstructorHandling

Gets or sets how constructors are used during deserialization.

ContractResolver

Gets or sets the contract resolver used by the serializer when serializing .NET objects to JSON and vice versa.

ReferenceResolver

Gets or sets the `Newtonsoft.Json.Serialization.IReferenceResolver`

TraceWriter

Gets or sets the `Newtonsoft.Json.Serialization.ITraceWriter`

Binder

Gets or sets the `Newtonsoft.Json.SerializationBinder`

Error

Gets or sets the error handler called during serialization and deserialization.

Context

Gets or sets the `System.Runtime.Serialization.StreamingContext`

DateFormatString

Get or set how `System.DateTime` `System.DateTimeOffset`

MaxDepth

Gets or sets the maximum depth allowed when reading JSON. Reading past this depth will throw a `Newtonsoft.Json.JsonReaderException`

Formatting

Indicates how JSON text output is formatted.

DateFormatHandling

Get or set how dates are written to JSON text.

DateTimeZoneHandling

Get or set how `System.DateTime`

DateParseHandling

Get or set how date formatted strings, e.g. `"/Date(1198908717056)/"` and `"2012-03-21T05:40Z"`, are parsed when reading JSON.

FloatFormatHandling

Get or set how special floating point numbers, e.g. `System.Double.NaN` `System.Double.PositiveInfinity` `System.Double.NegativeInfinity`

FloatParseHandling

Get or set how floating point numbers, e.g. 1.0 and 9.9, are parsed when reading JSON text.

StringEscapeHandling

Get or set how strings are escaped when writing JSON text.

Culture

Gets or sets the culture used when reading JSON. Defaults to `System.Globalization.CultureInfo.InvariantCulture`.

CheckAdditionalContent

Gets a value indicating whether there will be a check for additional content after deserializing an object.

1.46 JsonTextReader

class `Newtonsoft.Json.JsonTextReader`

Represents a reader that provides fast, non-cached, forward-only access to JSON text data.

Newtonsoft.Json.JsonTextReader.#ctor(System.IO.TextReader)

Initializes a new instance of the `Newtonsoft.Json.JsonTextReader` `System.IO.TextReader` :param reader: The

Read()

Reads the next JSON token from the stream. :return: true if the next token was read successfully; false if there are no more tokens to read.

ReadAsBytes()

Reads the next JSON token from the stream as a `System.Byte` :return: A `System.Byte`

ReadAsDecimal()

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.Nullable`1`

ReadAsInt32()

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.Nullable`1`

ReadAsString()

Reads the next JSON token from the stream as a `System.String` :return: A `System.String`

ReadAsDateTime()

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.String`

ReadAsDateTimeOffset()

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.DateTimeOffset`

Close()

Changes the state to closed.

HasLineInfo()

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

LineNumber

Gets the current line number.

LinePosition

Gets the current line position.

1.47 JsonTextWriter

class `Newtonsoft.Json.JsonTextWriter`

Represents a writer that provides a fast, non-cached, forward-only way of generating JSON data.

Newtonsoft.Json.JsonTextWriter.#ctor (`System.IO.TextWriter`)

Creates an instance of the `System.IO.TextWriter` :param textWriter: The

Flush ()

Flushes whatever is in the buffer to the underlying streams and also flushes the underlying stream.

Close ()

Closes this stream and the underlying stream.

WriteStartObject ()

Writes the beginning of a Json object.

WriteStartArray ()

Writes the beginning of a Json array.

WriteStartConstructor (`System.String`)

Writes the start of a constructor with the given name. :param name: The name of the constructor.

WriteEnd (`Newtonsoft.Json.JsonToken`)

Writes the specified end token. :param token: The end token to write.

WritePropertyName (`System.String`)

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

WritePropertyName (`System.String`, `System.Boolean`)

Writes the property name of a name/value pair on a JSON object. :param name: The name of the property. :param escape: A flag to indicate whether the text should be escaped when it is written as a JSON property name.

WriteIndent ()

Writes indent characters.

WriteValueDelimiter ()

Writes the JSON value delimiter.

WriteIndentSpace ()

Writes an indent space.

WriteValue (`System.Object`)

Writes a `System.Object` :param value: The

System.Object

WriteNull()

Writes a null value.

WriteUndefined()

Writes an undefined value.

WriteRaw(*System.String*)

Writes raw JSON. :param json: The raw JSON to write.

WriteValue(*System.String*)

Writes a System.String :param value: The

System.String

WriteValue(*System.Int32*)

Writes a System.Int32 :param value: The

System.Int32

WriteValue(*System.UInt32*)

Writes a System.UInt32 :param value: The

System.UInt32

WriteValue(*System.Int64*)

Writes a System.Int64 :param value: The

System.Int64

WriteValue(*System.UInt64*)

Writes a System.UInt64 :param value: The

System.UInt64

WriteValue(*System.Single*)

Writes a System.Single :param value: The

System.Single

WriteValue(*System.Nullable{System.Single}*)

Writes a System.Nullable`1 :param value: The

System.Nullable`1

WriteValue(*System.Double*)

Writes a System.Double :param value: The

System.Double

WriteValue(*System.Nullable{System.Double}*)

Writes a System.Nullable`1 :param value: The

System.Nullable`1

WriteValue(*System.Boolean*)

Writes a System.Boolean :param value: The

System.Boolean

WriteValue (*System.Int16*)

Writes a System.Int16 :param value: The

System.Int16

WriteValue (*System.UInt16*)

Writes a System.UInt16 :param value: The

System.UInt16

WriteValue (*System.Char*)

Writes a System.Char :param value: The

System.Char

WriteValue (*System.Byte*)

Writes a System.Byte :param value: The

System.Byte

WriteValue (*System.SByte*)

Writes a System.SByte :param value: The

System.SByte

WriteValue (*System.Decimal*)

Writes a System.Decimal :param value: The

System.Decimal

WriteValue (*System.DateTime*)

Writes a System.DateTime :param value: The

System.DateTime

WriteValue (*System.Byte[]*)

Writes a System.Byte :param value: The

System.Byte

WriteValue (*System.DateTimeOffset*)

Writes a System.DateTimeOffset :param value: The

System.DateTimeOffset

WriteValue (*System.Guid*)

Writes a System.Guid :param value: The

System.Guid

WriteValue (*System.TimeSpan*)

Writes a System.TimeSpan :param value: The

System.TimeSpan

WriteValue (*System.Uri*)

Writes a System.Uri :param value: The

`System.Uri`

WriteComment (*System.String*)

Writes out a comment :param text: Text to place inside the comment.

WriteWhitespace (*System.String*)

Writes out the given white space. :param ws: The string of white space characters.

Indentation

Gets or sets how many `IndentChars` to write for each level in the hierarchy when `Newtonsoft.Json.Formatting`

QuoteChar

Gets or sets which character to use to quote attribute values.

IndentChar

Gets or sets which character to use for indenting when `Newtonsoft.Json.Formatting`

QuoteName

Gets or sets a value indicating whether object names will be surrounded with quotes.

1.48 JsonToken

class `Newtonsoft.Json.JsonToken`

Specifies the type of Json token.

None

This is returned by the `Newtonsoft.Json.JsonReader` `Newtonsoft.Json.JsonReader.Read`

StartObject

An object start token.

StartArray

An array start token.

StartConstructor

A constructor start token.

PropertyName

An object property name.

Comment

A comment.

Raw

Raw JSON.

Integer

An integer.

Float

A float.

String

A string.

Boolean

A boolean.

Null

A null token.

Undefined

An undefined token.

EndObject

An object end token.

EndArray

An array end token.

EndConstructor

A constructor end token.

Date

A Date.

Bytes

Byte data.

1.49 JsonValidatingReader

class `Newtonsoft.Json.JsonValidatingReader`

Represents a reader that provides `Newtonsoft.Json.Schema.JsonSchema`

`Newtonsoft.Json.JsonValidatingReader.#ctor(Newtonsoft.Json.JsonReader)`

Initializes a new instance of the `Newtonsoft.Json.JsonValidatingReader`
`Newtonsoft.Json.JsonReader` :param reader: The

`Newtonsoft.Json.JsonReader`

`ReadAsInt32()`

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.Nullable`1`

`ReadAsBytes()`

Reads the next JSON token from the stream as a `System.Byte` :return: A `System.Byte`

`ReadAsDecimal()`

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.Nullable`1`

`ReadAsString()`

Reads the next JSON token from the stream as a `System.String` :return: A `System.String`

ReadAsDateTime()

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.String`

ReadAsDateTimeOffset()

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.Nullable`1`

Read()

Reads the next JSON token from the stream. :return: true if the next token was read successfully; false if there are no more tokens to read.

Sets an event handler for receiving schema validation errors.

Value

Gets the text value of the current JSON token.

Depth

Gets the depth of the current token in the JSON document.

Path

Gets the path of the current JSON token.

QuoteChar

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

TokenType

Gets the type of the current JSON token.

ValueType

Gets the Common Language Runtime (CLR) type for the current JSON token.

Schema

Gets or sets the schema.

Reader

Gets the `Newtonsoft.Json.JsonReader` `Newtonsoft.Json.JsonValidatingReader`
`Newtonsoft.Json.JsonReader`

1.50 JsonWriterException

class `Newtonsoft.Json.JsonWriterException`

The exception thrown when an error occurs while reading Json text.

Newtonsoft.Json.JsonWriterException.#ctor

Initializes a new instance of the `Newtonsoft.Json.JsonWriterException`

Newtonsoft.Json.JsonWriterException.#ctor(System.String)

Initializes a new instance of the `Newtonsoft.Json.JsonWriterException` :param message: The error message that explains the reason for the exception.

Newtonsoft.Json.JsonWriterException.#ctor(System.String, System.Exception)

Initializes a new instance of the `Newtonsoft.Json.JsonWriterException` :param message: The error message that explains the reason for the exception. :param innerException: The exception that is the cause of the current exception, or a null reference (Nothing in Visual Basic) if no inner exception is specified.

Path

Gets the path to the JSON where the error occurred.

1.51 Extensions

class `Newtonsoft.Json.Linq.Extensions`

Contains the LINQ to JSON extension methods.

Newtonsoft.Json.Linq.Extensions.Ancestors ``1 (System.Collections.Generic.IEnumerable

Returns a collection of tokens that contains the ancestors of every token in the source collection.

`Newtonsoft.Json.Linq.JToken` :param source: An

`System.Collections.Generic.IEnumerable` ``1 `Newtonsoft.Json.Linq.JToken` :return:

An `System.Collections.Generic.IEnumerable` ``1 `Newtonsoft.Json.Linq.JToken`

Newtonsoft.Json.Linq.Extensions.AncestorsAndSelf ``1 (System.Collections.Generic.IEnumerable

Returns a collection of tokens that contains every token in the source collection, and the ancestors of every token in the source collection. `Newtonsoft.Json.Linq.JToken` :param source: An

`System.Collections.Generic.IEnumerable` ``1 `Newtonsoft.Json.Linq.JToken` :return:

An `System.Collections.Generic.IEnumerable` ``1 `Newtonsoft.Json.Linq.JToken`

Newtonsoft.Json.Linq.Extensions.Descendants ``1 (System.Collections.Generic.IEnumerable

Returns a collection of tokens that contains the descendants of every token in the source collection.

`Newtonsoft.Json.Linq.JContainer` :param source: An

`System.Collections.Generic.IEnumerable` ``1 `Newtonsoft.Json.Linq.JToken` :return:

An `System.Collections.Generic.IEnumerable` ``1 `Newtonsoft.Json.Linq.JToken`

Newtonsoft.Json.Linq.Extensions.DescendantsAndSelf ``1 (System.Collections.Generic.IEnumerable

Returns a collection of tokens that contains every token in the source collection, and the descendants of every token in the source collection. `Newtonsoft.Json.Linq.JContainer` :param source: An

`System.Collections.Generic.IEnumerable` ``1 `Newtonsoft.Json.Linq.JToken` :return:

An `System.Collections.Generic.IEnumerable` ``1 `Newtonsoft.Json.Linq.JToken`

Properties (System.Collections.Generic.IEnumerable{Newtonsoft.Json.Linq.JObject})

Returns a collection of child properties of every object in the source collection. :param source: An

`System.Collections.Generic.IEnumerable` ``1 `Newtonsoft.Json.Linq.JObject` :return:

An `System.Collections.Generic.IEnumerable` ``1 `Newtonsoft.Json.Linq.JProperty`

Values (System.Collections.Generic.IEnumerable{Newtonsoft.Json.Linq.JToken}, System.Object)

Returns a collection of child values of every object in the source collection with the given key. :param source: An

`System.Collections.Generic.IEnumerable` ``1 `Newtonsoft.Json.Linq.JToken` :param key: The token key

```
return An System.Collections.Generic.IEnumerable`1
Newtonsoft.Json.Linq.JToken
```

Values (*System.Collections.Generic.IEnumerable{Newtonsoft.Json.Linq.JToken}*)

Returns a collection of child values of every object in the source collection. :param source: An

```
System.Collections.Generic.IEnumerable`1 Newtonsoft.Json.Linq.JToken :return:
An System.Collections.Generic.IEnumerable`1 Newtonsoft.Json.Linq.JToken
```

Newtonsoft.Json.Linq.Extensions.Values ``1(*System.Collections.Generic.IEnumerable{New*

Returns a collection of converted child values of every object in the source collection with the given key. :param source: An

System.Collections.Generic.IEnumerable`1 *Newtonsoft.Json.Linq.JToken* :param key: The token ke

```
return An System.Collections.Generic.IEnumerable`1
```

Newtonsoft.Json.Linq.Extensions.Values ``1(*System.Collections.Generic.IEnumerable{New*

Returns a collection of converted child values of every object in the source collection. :param source: An

```
System.Collections.Generic.IEnumerable`1 Newtonsoft.Json.Linq.JToken :return:
An System.Collections.Generic.IEnumerable`1
```

Newtonsoft.Json.Linq.Extensions.Value ``1(*System.Collections.Generic.IEnumerable{New*

Converts the value. :param value: A

```
Newtonsoft.Json.Linq.JToken System.Collections.Generic.IEnumerable`1
Newtonsoft.Json.Linq.JToken :return: A converted value.
```

Newtonsoft.Json.Linq.Extensions.Value ``2(*System.Collections.Generic.IEnumerable{ ``0*

Converts the value. :param value: A

```
Newtonsoft.Json.Linq.JToken System.Collections.Generic.IEnumerable`1
Newtonsoft.Json.Linq.JToken :return: A converted value.
```

Newtonsoft.Json.Linq.Extensions.Children ``1(*System.Collections.Generic.IEnumerable{*

Returns a collection of child tokens of every array in the source collection. :param source: An

```
System.Collections.Generic.IEnumerable`1 Newtonsoft.Json.Linq.JToken :return:
An System.Collections.Generic.IEnumerable`1 Newtonsoft.Json.Linq.JToken
```

Newtonsoft.Json.Linq.Extensions.Children ``2(*System.Collections.Generic.IEnumerable{*

Returns a collection of converted child tokens of every array in the source collection. :param source: An

```
System.Collections.Generic.IEnumerable`1 Newtonsoft.Json.Linq.JToken :return:
An System.Collections.Generic.IEnumerable`1
```

AsJEnumerable (*System.Collections.Generic.IEnumerable{Newtonsoft.Json.Linq.JToken}*)

Returns the input typed as *Newtonsoft.Json.Linq.IJEnumerable`1* :param source: An

```
System.Collections.Generic.IEnumerable`1 Newtonsoft.Json.Linq.JToken :return:
The input typed as Newtonsoft.Json.Linq.IJEnumerable`1
```

Newtonsoft.Json.Linq.Extensions.AsJEnumerable ``1(*System.Collections.Generic.IEnumer*

Returns the input typed as *Newtonsoft.Json.Linq.IJEnumerable`1* :param source: An

`System.Collections.Generic.IEnumerable`1 Newtonsoft.Json.Linq.JToken` :return:
The input typed as `Newtonsoft.Json.Linq.IEnumerable`1`

1.52 IEnumerable`1

Newtonsoft.Json.Linq.IEnumerable`1

Represents a collection of `Newtonsoft.Json.Linq.JToken`

Newtonsoft.Json.Linq.IEnumerable`1.Item(System.Object)

Gets the `Newtonsoft.Json.Linq.IEnumerable`1`

1.53 JArray

class Newtonsoft.Json.Linq.JArray

Represents a JSON array.

1.54 JContainer

class Newtonsoft.Json.Linq.JContainer

Represents a token that can contain other tokens.

1.55 JToken

class Newtonsoft.Json.Linq.JToken

Represents an abstract JSON token.

DeepEquals (*Newtonsoft.Json.Linq.JToken, Newtonsoft.Json.Linq.JToken*)

Compares the values of two tokens, including the values of all descendant tokens. :param t1: The first

`Newtonsoft.Json.Linq.JToken` :param t2: The second `Newtonsoft.Json.Linq.JToken` :return: true if the tokens are equal; otherwise false.

AddAfterSelf (*System.Object*)

Adds the specified content immediately after this token. :param content: A content object that contains simple content or a collection of content objects to be added after this token.

AddBeforeSelf (*System.Object*)

Adds the specified content immediately before this token. :param content: A content object that contains simple content or a collection of content objects to be added before this token.

Ancestors ()

Returns a collection of the ancestor tokens of this token. :return: A collection of the ancestor tokens of this token.

AncestorsAndSelf ()

Returns a collection of tokens that contain this token, and the ancestors of this token. :return: A collection of tokens that contain this token, and the ancestors of this token.

AfterSelf()

Returns a collection of the sibling tokens after this token, in document order. :return: A collection of the sibling tokens after this tokens, in document order.

BeforeSelf()

Returns a collection of the sibling tokens before this token, in document order. :return: A collection of the sibling tokens before this token, in document order.

Newtonsoft.Json.Linq.JToken.Value`1(System.Object)

Gets the `Newtonsoft.Json.Linq.JToken` :param key: The token key. :return: The converted token value.

Children()

Returns a collection of the child tokens of this token, in document order. :return: An `System.Collections.Generic.IEnumerable`1` `Newtonsoft.Json.Linq.JToken` `Newtonsoft.Json.Linq.JToken`

Newtonsoft.Json.Linq.JToken.Children`1

Returns a collection of the child tokens of this token, in document order, filtered by the specified type. :return: A `Newtonsoft.Json.Linq.JEnumerable`1` `Newtonsoft.Json.Linq.JToken`

Newtonsoft.Json.Linq.JToken.Values`1

Returns a collection of the child values of this token, in document order. :return: A `System.Collections.Generic.IEnumerable`1` `Newtonsoft.Json.Linq.JToken`

Remove()

Removes this token from its parent.

Replace(Newtonsoft.Json.Linq.JToken)

Replaces this token with the specified token. :param value: The value.

WriteTo(Newtonsoft.Json.JsonWriter, Newtonsoft.Json.JsonConverter[])

Writes this token to a `Newtonsoft.Json.JsonWriter` :param writer: A

`Newtonsoft.Json.JsonWriter` :param converters: A collection of `Newtonsoft.Json.JsonConverter`

ToString()

Returns the indented JSON for this token. :return: The indented JSON for this token.

ToString(Newtonsoft.Json.Formatting, Newtonsoft.Json.JsonConverter[])

Returns the JSON for this token using the given formatting and converters. :param formatting: Indicates how the output is formatted. :param converters: A collection of

`Newtonsoft.Json.JsonConverter` :return: The JSON for this token using the given formatting and converters.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Boolean

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Boolean` :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.DateTime

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.DateTimeOffset` :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Nullable`1` :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Int64

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Int64` :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Nullable`1` :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Nullable`1` :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Nullable`1` :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Nullable`1` :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Nullable`1` :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Int32

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Int32` :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Int16

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Int16` :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.UInt16

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.UInt16` :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Char

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Char` :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Byte

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Byte` :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.SByte

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.SByte`
:param value: The value. :return: The result of the conversion.

`Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable<SByte>`

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Nullable<SByte>`
:param value: The value. :return: The result of the conversion.

`Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable<Short>`

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Nullable<Short>`
:param value: The value. :return: The result of the conversion.

`Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable<Int>`

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Nullable<Int>`
:param value: The value. :return: The result of the conversion.

`Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable<Int32>`

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Nullable<Int32>`
:param value: The value. :return: The result of the conversion.

`Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable<Int64>`

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Nullable<Int64>`
:param value: The value. :return: The result of the conversion.

`Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.DateTime`

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.DateTime`
:param value: The value. :return: The result of the conversion.

`Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable<DateTime>`

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Nullable<DateTime>`
:param value: The value. :return: The result of the conversion.

`Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable<Decimal>`

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Nullable<Decimal>`
:param value: The value. :return: The result of the conversion.

`Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Decimal`

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Decimal`
:param value: The value. :return: The result of the conversion.

`Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable<Double>`

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Nullable<Double>`
:param value: The value. :return: The result of the conversion.

`Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable<Single>`

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Nullable<Single>`
:param value: The value. :return: The result of the conversion.

`Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Double`

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Double`
:param value: The value. :return: The result of the conversion.

`Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Single`

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Single`
:param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.String

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.String`
 :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.UInt32

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.UInt32`
 :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.UInt64

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.UInt64`
 :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Byte[]

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Byte`
 :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Guid

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Guid`
 :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable<System.Guid>

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Guid`
 :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.TimeSpan

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.TimeSpan`
 :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Nullable<System.TimeSpan>

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.TimeSpan`
 :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Explicit(Newtonsoft.Json.Linq.JToken) ~System.Uri

Performs an explicit conversion from `Newtonsoft.Json.Linq.JToken` `System.Uri`
 :param value: The value. :return: The result of the conversion.

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Boolean) ~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from `System.Boolean` `Newtonsoft.Json.Linq.JToken`
 :param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

Newtonsoft.Json.Linq.JToken.op_Implicit(System.DateTimeOffset) ~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from `System.DateTimeOffset` `Newtonsoft.Json.Linq.JToken`
 :param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Byte) ~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from `System.Byte` `Newtonsoft.Json.Linq.JToken`
 :param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable<System.Byte>) ~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.Nullable<System.Byte> to Newtonsoft.Json.Linq.JToken :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.SByte) ~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.SByte to Newtonsoft.Json.Linq.JToken :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable<System.SByte>) ~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.Nullable<System.SByte> to Newtonsoft.Json.Linq.JToken :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable<System.Boolean>) ~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.Nullable<System.Boolean> to Newtonsoft.Json.Linq.JToken :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Int64) ~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.Int64 to Newtonsoft.Json.Linq.JToken :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable<System.DateTime>) ~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.Nullable<System.DateTime> to Newtonsoft.Json.Linq.JToken :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable<System.DateTimeOffset>) ~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.Nullable<System.DateTimeOffset> to Newtonsoft.Json.Linq.JToken :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable<System.Decimal>) ~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.Nullable<System.Decimal> to Newtonsoft.Json.Linq.JToken :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable<System.Double>) ~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.Nullable<System.Double> to Newtonsoft.Json.Linq.JToken :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Int16) ~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.Int16 to Newtonsoft.Json.Linq.JToken :param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

`Newtonsoft.Json.Linq.JToken.op_Implicit(System.UInt16)~Newtonsoft.Json.Linq.JToken`

Performs an implicit conversion from `System.UInt16` `Newtonsoft.Json.Linq.JToken`
:param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

`Newtonsoft.Json.Linq.JToken.op_Implicit(System.Int32)~Newtonsoft.Json.Linq.JToken`

Performs an implicit conversion from `System.Int32` `Newtonsoft.Json.Linq.JToken`
:param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

`Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable{System.Int32})~Newtonsoft.Json.Linq.JToken`

Performs an implicit conversion from `System.Nullable`1` `Newtonsoft.Json.Linq.JToken` :param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

`Newtonsoft.Json.Linq.JToken.op_Implicit(System.DateTime)~Newtonsoft.Json.Linq.JToken`

Performs an implicit conversion from `System.DateTime` `Newtonsoft.Json.Linq.JToken` :param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

`Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable{System.Int64})~Newtonsoft.Json.Linq.JToken`

Performs an implicit conversion from `System.Nullable`1` `Newtonsoft.Json.Linq.JToken` :param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

`Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable{System.Single})~Newtonsoft.Json.Linq.JToken`

Performs an implicit conversion from `System.Nullable`1` `Newtonsoft.Json.Linq.JToken` :param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

`Newtonsoft.Json.Linq.JToken.op_Implicit(System.Decimal)~Newtonsoft.Json.Linq.JToken`

Performs an implicit conversion from `System.Decimal` `Newtonsoft.Json.Linq.JToken`
:param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

`Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable{System.Int16})~Newtonsoft.Json.Linq.JToken`

Performs an implicit conversion from `System.Nullable`1` `Newtonsoft.Json.Linq.JToken` :param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

`Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable{System.UInt16})~Newtonsoft.Json.Linq.JToken`

Performs an implicit conversion from `System.Nullable`1` `Newtonsoft.Json.Linq.JToken` :param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable{System.UInt32})~Newtonsoft.

Performs an implicit conversion from System.Nullable`1
 Newtonsoft.Json.Linq.JToken :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable{System.UInt64})~Newtonsoft.

Performs an implicit conversion from System.Nullable`1
 Newtonsoft.Json.Linq.JToken :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Double)~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.Double Newtonsoft.Json.Linq.JToken
 :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Single)~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.Single Newtonsoft.Json.Linq.JToken
 :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.String)~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.String Newtonsoft.Json.Linq.JToken
 :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.UInt32)~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.UInt32 Newtonsoft.Json.Linq.JToken
 :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.UInt64)~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.UInt64 Newtonsoft.Json.Linq.JToken
 :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Byte[])~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.Byte Newtonsoft.Json.Linq.JToken
 :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.Uri)~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.Uri Newtonsoft.Json.Linq.JToken
 :param value: The value to create a

Newtonsoft.Json.Linq.JValue :return: The Newtonsoft.Json.Linq.JValue

Newtonsoft.Json.Linq.JToken.op_Implicit(System.TimeSpan)~Newtonsoft.Json.Linq.JToken

Performs an implicit conversion from System.TimeSpan
 Newtonsoft.Json.Linq.JToken :param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

`Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable{System.TimeSpan}) ~Newtonsoft.Json.Linq.JToken`

Performs an implicit conversion from `System.Nullable`1`
`Newtonsoft.Json.Linq.JToken` :param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

`Newtonsoft.Json.Linq.JToken.op_Implicit(System.Guid) ~Newtonsoft.Json.Linq.JToken`

Performs an implicit conversion from `System.Guid` `Newtonsoft.Json.Linq.JToken`
 :param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

`Newtonsoft.Json.Linq.JToken.op_Implicit(System.Nullable{System.Guid}) ~Newtonsoft.Json.Linq.JToken`

Performs an implicit conversion from `System.Nullable`1`
`Newtonsoft.Json.Linq.JToken` :param value: The value to create a

`Newtonsoft.Json.Linq.JValue` :return: The `Newtonsoft.Json.Linq.JValue`

`CreateReader()`

Creates an `Newtonsoft.Json.JsonReader` :return: An
`Newtonsoft.Json.JsonReader`

`FromObject(System.Object)`

Creates a `Newtonsoft.Json.Linq.JToken` :param o: The object that will be used to create

`Newtonsoft.Json.Linq.JToken` :return: A `Newtonsoft.Json.Linq.JToken`

`FromObject(System.Object, Newtonsoft.Json.JsonSerializer)`

Creates a `Newtonsoft.Json.Linq.JToken` `Newtonsoft.Json.JsonSerializer`
 :param o: The object that will be used to create

`Newtonsoft.Json.Linq.JToken` :param jsonSerializer: The `Newtonsoft.Json.JsonSerializer`
 :return: A `Newtonsoft.Json.Linq.JToken`

`Newtonsoft.Json.Linq.JToken.ToObject`1`

Creates the specified .NET type from the `Newtonsoft.Json.Linq.JToken` :return: The new
 object created from the JSON value.

`ToObject(System.Type)`

Creates the specified .NET type from the `Newtonsoft.Json.Linq.JToken` :param object-
 Type: The object type that the token will be deserialized to. :return: The new object created from the
 JSON value.

`Newtonsoft.Json.Linq.JToken.ToObject`1(Newtonsoft.Json.JsonSerializer)`

Creates the specified .NET type from the `Newtonsoft.Json.Linq.JToken`
`Newtonsoft.Json.JsonSerializer` :param jsonSerializer: The

`Newtonsoft.Json.JsonSerializer` :return: The new object created from the JSON value.

`ToObject(System.Type, Newtonsoft.Json.JsonSerializer)`

Creates the specified .NET type from the `Newtonsoft.Json.Linq.JToken`
`Newtonsoft.Json.JsonSerializer` :param objectType: The object type that the to-
 ken will be deserialized to. :param jsonSerializer: The

`Newtonsoft.Json.JsonSerializer` :return: The new object created from the JSON value.

ReadFrom (*Newtonsoft.Json.JsonReader*)

Creates a `Newtonsoft.Json.Linq.JToken` `Newtonsoft.Json.JsonReader` :param reader: An

`Newtonsoft.Json.JsonReader` `Newtonsoft.Json.Linq.JToken` :return: An
`Newtonsoft.Json.Linq.JToken`

Parse (*System.String*)

Load a `Newtonsoft.Json.Linq.JToken` :param json: A

`System.String` :return: A `Newtonsoft.Json.Linq.JToken`

Load (*Newtonsoft.Json.JsonReader*)

Creates a `Newtonsoft.Json.Linq.JToken` `Newtonsoft.Json.JsonReader` :param reader: An

`Newtonsoft.Json.JsonReader` `Newtonsoft.Json.Linq.JToken` :return: An
`Newtonsoft.Json.Linq.JToken`

SelectToken (*System.String*)

Selects a `Newtonsoft.Json.Linq.JToken` :param path: A

`System.String` :return: A `Newtonsoft.Json.Linq.JToken`

SelectToken (*System.String, System.Boolean*)

Selects a `Newtonsoft.Json.Linq.JToken` :param path: A

`System.String` :param errorWhenNoMatch: A flag to indicate whether an error should be thrown if no tokens are found

return A `Newtonsoft.Json.Linq.JToken`

SelectTokens (*System.String*)

Selects a collection of elements using a JPath expression. :param path: A

`System.String` :return: An `System.Collections.Generic.IEnumerable`1`

SelectTokens (*System.String, System.Boolean*)

Selects a collection of elements using a JPath expression. :param path: A

`System.String` :param errorWhenNoMatch: A flag to indicate whether an error should be thrown if no tokens are found

return An `System.Collections.Generic.IEnumerable`1`

GetMetaObject (*System.Linq.Expressions.Expression*)

Returns the `System.Dynamic.DynamicMetaObject` :param parameter: The expression tree representation of the runtime value. :return: The `System.Dynamic.DynamicMetaObject`

`Newtonsoft.Json.Linq.JToken` `System#Dynamic#IDynamicMetaObjectProvider#GetMetaObject`

Returns the `System.Dynamic.DynamicMetaObject` :param parameter: The expression tree representation of the runtime value. :return: The `System.Dynamic.DynamicMetaObject`

DeepClone ()

Creates a new instance of the `Newtonsoft.Json.Linq.JToken` :return: A new instance of the `Newtonsoft.Json.Linq.JToken`

AddAnnotation (*System.Object*)

Adds an object to the annotation list of this `Newtonsoft.Json.Linq.JToken` :param annotation: The annotation to add.

Newtonsoft.Json.Linq.JToken.Annotation ``1

Get the first annotation object of the specified type from this `Newtonsoft.Json.Linq.JToken` :return: The first annotation object that matches the specified type, or

Annotation (*System.Type*)

Gets the first annotation object of the specified type from this `Newtonsoft.Json.Linq.JToken` :param type: The

`Newtonsoft.Json.Linq.JToken.Type` :return: The first annotation object that matches the specified type, or

Newtonsoft.Json.Linq.JToken.Annotations ``1

Gets a collection of annotations of the specified type for this `Newtonsoft.Json.Linq.JToken` :return: An `System.Collections.Generic.IEnumerable<Newtonsoft.Json.Linq.JToken>`

Annotations (*System.Type*)

Gets a collection of annotations of the specified type for this `Newtonsoft.Json.Linq.JToken` :param type: The

`Newtonsoft.Json.Linq.JToken.Type` :return: An `System.Collections.Generic.IEnumerable<System.Object>` `Newtonsoft.Json.Linq.JToken`

Newtonsoft.Json.Linq.JToken.RemoveAnnotations ``1

Removes the annotations of the specified type from this `Newtonsoft.Json.Linq.JToken`

RemoveAnnotations (*System.Type*)

Removes the annotations of the specified type from this `Newtonsoft.Json.Linq.JToken` :param type: The

`Newtonsoft.Json.Linq.JToken.Type`

EqualityComparer

Gets a comparer that can compare two tokens for value equality. `Newtonsoft.Json.Linq.JTokenEqualityComparer`

Parent

Gets or sets the parent.

Root

Gets the root `Newtonsoft.Json.Linq.JToken` `Newtonsoft.Json.Linq.JToken` `Newtonsoft.Json.Linq.JToken`

Type

Gets the node type for this `Newtonsoft.Json.Linq.JToken`

HasValues

Gets a value indicating whether this token has child tokens.

Next

Gets the next sibling token of this node. `Newtonsoft.Json.Linq.JToken`

Previous

Gets the previous sibling token of this node. `Newtonsoft.Json.Linq.JToken`

Path

Gets the path of the JSON token.

Item (*System.Object*)

Gets the `Newtonsoft.Json.Linq.JToken` `Newtonsoft.Json.Linq.JToken`

First

Get the first child token of this token. `Newtonsoft.Json.Linq.JToken`
`Newtonsoft.Json.Linq.JToken`

Last

Get the last child token of this token. `Newtonsoft.Json.Linq.JToken`
`Newtonsoft.Json.Linq.JToken`

`Newtonsoft.Json.Linq.JContainer.OnCollectionChanged (System.Collections.Specialized.NotifyCollectionChangedEventArgs)`

Raises the `Newtonsoft.Json.Linq.JContainer.CollectionChanged` :param e: The

`System.Collections.Specialized.NotifyCollectionChangedEventArgs`

`Newtonsoft.Json.Linq.JContainer.Children ()`

Returns a collection of the child tokens of this token, in document order.
:return: An `System.Collections.Generic.IEnumerable<Newtonsoft.Json.Linq.JToken>`

`Newtonsoft.Json.Linq.JContainer.Values <T>`

Returns a collection of the child values of this token, in document order.
:return: A `System.Collections.Generic.IEnumerable<T>`

`Newtonsoft.Json.Linq.JContainer.Descendants ()`

Returns a collection of the descendant tokens for this token in document order.
:return: An `System.Collections.Generic.IEnumerable<Newtonsoft.Json.Linq.JToken>`

`Newtonsoft.Json.Linq.JContainer.DescendantsAndSelf ()`

Returns a collection of the tokens that contain this token, and all descendant tokens of this token, in document order. :return: An `System.Collections.Generic.IEnumerable<Newtonsoft.Json.Linq.JToken>`

`Newtonsoft.Json.Linq.JContainer.Add (System.Object)`

Adds the specified content as children of this `Newtonsoft.Json.Linq.JToken` :param content: The content to be added.

`Newtonsoft.Json.Linq.JContainer.AddFirst (System.Object)`

Adds the specified content as the first children of this `Newtonsoft.Json.Linq.JToken` :param content: The content to be added.

`Newtonsoft.Json.Linq.JContainer.CreateWriter ()`

Creates an `Newtonsoft.Json.JsonWriter` `Newtonsoft.Json.Linq.JToken` :return: An `Newtonsoft.Json.JsonWriter`

`Newtonsoft.Json.Linq.JContainer`.**ReplaceAll** (*System.Object*)

Replaces the children nodes of this token with the specified content. :param content: The content.

`Newtonsoft.Json.Linq.JContainer`.**RemoveAll** ()

Removes the child nodes from this token.

`Newtonsoft.Json.Linq.JContainer`.**Merge** (*System.Object*)

Merge the specified content into this `Newtonsoft.Json.Linq.JToken` :param content: The content to be merged.

`Newtonsoft.Json.Linq.JContainer`.**Merge** (*System.Object*, *Newtonsoft.Json.Linq.JsonMergeSettings*)

Merge the specified content into this `Newtonsoft.Json.Linq.JToken` `Newtonsoft.Json.Linq.JsonMergeSettings` :param content: The content to be merged. :param settings: The

`Newtonsoft.Json.Linq.JsonMergeSettings`

Occurs when the items list of the collection has changed, or the collection is reset.

`Newtonsoft.Json.Linq.JContainer`.**ChildrenTokens**

Gets the container's children tokens.

`Newtonsoft.Json.Linq.JContainer`.**HasValues**

Gets a value indicating whether this token has child tokens.

`Newtonsoft.Json.Linq.JContainer`.**First**

Get the first child token of this token. `Newtonsoft.Json.Linq.JToken` `Newtonsoft.Json.Linq.JToken`

`Newtonsoft.Json.Linq.JContainer`.**Last**

Get the last child token of this token. `Newtonsoft.Json.Linq.JToken` `Newtonsoft.Json.Linq.JToken`

`Newtonsoft.Json.Linq.JContainer`.**Count**

Gets the count of child JSON tokens.

Newtonsoft.Json.Linq.JArray.#ctor

Initializes a new instance of the `Newtonsoft.Json.Linq.JArray`

Newtonsoft.Json.Linq.JArray.#ctor (Newtonsoft.Json.Linq.JArray)

Initializes a new instance of the `Newtonsoft.Json.Linq.JArray` `Newtonsoft.Json.Linq.JArray` :param other: A

`Newtonsoft.Json.Linq.JArray`

Newtonsoft.Json.Linq.JArray.#ctor (System.Object [])

Initializes a new instance of the `Newtonsoft.Json.Linq.JArray` :param content: The contents of the array.

Newtonsoft.Json.Linq.JArray.#ctor (System.Object)

Initializes a new instance of the `Newtonsoft.Json.Linq.JArray` :param content: The contents of the array.

`Newtonsoft.Json.Linq.JArray.Load(Newtonsoft.Json.JsonReader)`

Loads an `Newtonsoft.Json.Linq.JArray` `Newtonsoft.Json.JsonReader` :param reader: A

`Newtonsoft.Json.JsonReader` `Newtonsoft.Json.Linq.JArray` :return: A
`Newtonsoft.Json.Linq.JArray` `Newtonsoft.Json.JsonReader`

`Newtonsoft.Json.Linq.JArray.Parse(System.String)`

Load a `Newtonsoft.Json.Linq.JArray` :param json: A

`System.String` :return: A `Newtonsoft.Json.Linq.JArray`

`Newtonsoft.Json.Linq.JArray.FromObject(System.Object)`

Creates a `Newtonsoft.Json.Linq.JArray` :param o: The object that will be used to create

`Newtonsoft.Json.Linq.JArray` :return: A `Newtonsoft.Json.Linq.JArray`

`Newtonsoft.Json.Linq.JArray.FromObject(System.Object, Newtonsoft.Json.JsonSerializer)`

Creates a `Newtonsoft.Json.Linq.JArray` :param o: The object that will be used to create

`Newtonsoft.Json.Linq.JArray` :param jsonSerializer: The `Newtonsoft.Json.JsonSerializer`
:return: A `Newtonsoft.Json.Linq.JArray`

`Newtonsoft.Json.Linq.JArray.WriteTo(Newtonsoft.Json.JsonWriter, Newtonsoft.Json.JsonConverter[])`

Writes this token to a `Newtonsoft.Json.JsonWriter` :param writer: A

`Newtonsoft.Json.JsonWriter` :param converters: A collection of
`Newtonsoft.Json.JsonConverter`

`Newtonsoft.Json.Linq.JArray.IndexOf(Newtonsoft.Json.Linq.JToken)`

Determines the index of a specific item in the `System.Collections.Generic.IList`1`
:param item: The object to locate in the

`System.Collections.Generic.IList`1` :return: The index of

`Newtonsoft.Json.Linq.JArray.Insert(System.Int32, Newtonsoft.Json.Linq.JToken)`

Inserts an item to the `System.Collections.Generic.IList`1` :param index: The zero-based index at which :param item: The object to insert into the

`System.Collections.Generic.IList`1` `System.Collections.Generic.IList`1`
`System.Collections.Generic.IList`1`

`Newtonsoft.Json.Linq.JArray.RemoveAt(System.Int32)`

Removes the `System.Collections.Generic.IList`1` :param index: The zero-based index of the item to remove.

`System.Collections.Generic.IList`1` `System.Collections.Generic.IList`1`

`Newtonsoft.Json.Linq.JArray.GetEnumerator()`

Returns an enumerator that iterates through the collection. :return: A
`System.Collections.Generic.IEnumerator`1`

`Newtonsoft.Json.Linq.JArray.Add(Newtonsoft.Json.Linq.JToken)`

Adds an item to the `System.Collections.Generic.ICollection`1` :param item: The object to add to the

`System.Collections.Generic.ICollection`1` `System.Collections.Generic.ICollection`1`

`Newtonsoft.Json.Linq.JArray.Clear()`

Removes all items from the `System.Collections.Generic.ICollection`1`
`System.Collections.Generic.ICollection`1`

`Newtonsoft.Json.Linq.JArray.Contains(Newtonsoft.Json.Linq.JToken)`

Determines whether the `System.Collections.Generic.ICollection`1` :param item:
 The object to locate in the

`System.Collections.Generic.ICollection`1` :return: true if
`System.Collections.Generic.ICollection`1`

`Newtonsoft.Json.Linq.JArray.CopyTo(Newtonsoft.Json.Linq.JToken[], System.Int32)`

Copies to. :param array: The array. :param arrayIndex: Index of the array.

`Newtonsoft.Json.Linq.JArray.Remove(Newtonsoft.Json.Linq.JToken)`

Removes the first occurrence of a specific object from the
`System.Collections.Generic.ICollection`1` :param item: The object to remove
 from the

`System.Collections.Generic.ICollection`1` :return: true if
`System.Collections.Generic.ICollection`1` `System.Collections.Generic.ICollection`1`
`System.Collections.Generic.ICollection`1`

`Newtonsoft.Json.Linq.JArray.ChildrenTokens`

Gets the container's children tokens.

`Newtonsoft.Json.Linq.JArray.Type`

Gets the node type for this `Newtonsoft.Json.Linq.JToken`

`Newtonsoft.Json.Linq.JArray.Item(System.Object)`

Gets the `Newtonsoft.Json.Linq.JToken` `Newtonsoft.Json.Linq.JToken`

`Newtonsoft.Json.Linq.JArray.Item(System.Int32)`

Gets or sets the `Newtonsoft.Json.Linq.JToken`

`Newtonsoft.Json.Linq.JArray.IsReadOnly`

Gets a value indicating whether the `System.Collections.Generic.ICollection`1` :re-
 turn: true if the `System.Collections.Generic.ICollection`1`

1.56 JConstructor

class `Newtonsoft.Json.Linq.JConstructor`

Represents a JSON constructor.

Newtonsoft.Json.Linq.JConstructor.#ctor

Initializes a new instance of the `Newtonsoft.Json.Linq.JConstructor`

Newtonsoft.Json.Linq.JConstructor.#ctor(Newtonsoft.Json.Linq.JConstructor)

Initializes a new instance of the `Newtonsoft.Json.Linq.JConstructor`
`Newtonsoft.Json.Linq.JConstructor` :param other: A

`Newtonsoft.Json.Linq.JConstructor`

`Newtonsoft.Json.Linq.JConstructor.#ctor(System.String, System.Object[])`

Initializes a new instance of the `Newtonsoft.Json.Linq.JConstructor` :param name: The constructor name. :param content: The contents of the constructor.

`Newtonsoft.Json.Linq.JConstructor.#ctor(System.String, System.Object)`

Initializes a new instance of the `Newtonsoft.Json.Linq.JConstructor` :param name: The constructor name. :param content: The contents of the constructor.

`Newtonsoft.Json.Linq.JConstructor.#ctor(System.String)`

Initializes a new instance of the `Newtonsoft.Json.Linq.JConstructor` :param name: The constructor name.

`WriteTo(Newtonsoft.Json.JsonWriter, Newtonsoft.Json.JsonConverter[])`

Writes this token to a `Newtonsoft.Json.JsonWriter` :param writer: A

`Newtonsoft.Json.JsonWriter` :param converters: A collection of
`Newtonsoft.Json.JsonConverter`

`Load(Newtonsoft.Json.JsonReader)`

Loads an `Newtonsoft.Json.Linq.JConstructor` `Newtonsoft.Json.JsonReader`
:param reader: A

`Newtonsoft.Json.JsonReader` `Newtonsoft.Json.Linq.JConstructor` :return: A
`Newtonsoft.Json.Linq.JConstructor` `Newtonsoft.Json.JsonReader`

`ChildrenTokens`

Gets the container's children tokens.

`Name`

Gets or sets the name of this constructor.

`Type`

Gets the node type for this `Newtonsoft.Json.Linq.JToken`

`Item(System.Object)`

Gets the `Newtonsoft.Json.Linq.JToken` `Newtonsoft.Json.Linq.JToken`

1.57 JEnumerable`1

`Newtonsoft.Json.Linq.JEnumerable`1`

Represents a collection of `Newtonsoft.Json.Linq.JToken`

`Newtonsoft.Json.Linq.JEnumerable`1.Empty`

An empty collection of `Newtonsoft.Json.Linq.JToken`

`Newtonsoft.Json.Linq.JEnumerable`1.#ctor(System.Collections.Generic.IEnumerable<T>)`

Initializes a new instance of the `Newtonsoft.Json.Linq.JEnumerable`1` :param enumerable: The enumerable.

`Newtonsoft.Json.Linq.JEnumerable`1.GetEnumerator`

Returns an enumerator that iterates through the collection. :return: A
 System.Collections.Generic.IEnumerator`1

Newtonsoft.Json.Linq.JEnumerable`1.System#Collections#IEnumerable#GetEnumerator

Returns an enumerator that iterates through a collection. :return: An
 System.Collections.IEnumerator

Newtonsoft.Json.Linq.JEnumerable`1.Equals(Newtonsoft.Json.Linq.JEnumerable`0)

Determines whether the specified Newtonsoft.Json.Linq.JEnumerable`1 :param other:
 The

Newtonsoft.Json.Linq.JEnumerable`1 :return: Newtonsoft.Json.Linq.JEnumerable`1

Newtonsoft.Json.Linq.JEnumerable`1.Equals(System.Object)

Determines whether the specified System.Object :param obj: The

System.Object :return: System.Object

Newtonsoft.Json.Linq.JEnumerable`1.GetHashCode

Returns a hash code for this instance. :return: A hash code for this instance, suitable for use in
 hashing algorithms and data structures like a hash table.

Newtonsoft.Json.Linq.JEnumerable`1.Item(System.Object)

Gets the Newtonsoft.Json.Linq.IJEnumerable`1

1.58 JObject

class Newtonsoft.Json.Linq.JObject

Represents a JSON object.

Newtonsoft.Json.Linq.JObject.#ctor

Initializes a new instance of the `Newtonsoft.Json.Linq.JObject`

Newtonsoft.Json.Linq.JObject.#ctor(Newtonsoft.Json.Linq.JObject)

Initializes a new instance of the `Newtonsoft.Json.Linq.JObject`
`Newtonsoft.Json.Linq.JObject` :param other: A

`Newtonsoft.Json.Linq.JObject`

Newtonsoft.Json.Linq.JObject.#ctor(System.Object[])

Initializes a new instance of the `Newtonsoft.Json.Linq.JObject` :param content: The con-
 tents of the object.

Newtonsoft.Json.Linq.JObject.#ctor(System.Object)

Initializes a new instance of the `Newtonsoft.Json.Linq.JObject` :param content: The con-
 tents of the object.

Properties()

Gets an `System.Collections.Generic.IEnumerable`1` :return: An
`System.Collections.Generic.IEnumerable`1`

Property(System.String)

Gets a `Newtonsoft.Json.Linq.JProperty` :param name: The property name. :return: A `Newtonsoft.Json.Linq.JProperty`

PropertyValues ()

Gets an `Newtonsoft.Json.Linq.JEnumerable`1` :return: An `Newtonsoft.Json.Linq.JEnumerable`1`

Load (*Newtonsoft.Json.JsonReader*)

Loads an `Newtonsoft.Json.Linq.JObject` `Newtonsoft.Json.JsonReader` :param reader: A

`Newtonsoft.Json.JsonReader` `Newtonsoft.Json.Linq.JObject` :return: A `Newtonsoft.Json.Linq.JObject` `Newtonsoft.Json.JsonReader`

Parse (*System.String*)

Load a `Newtonsoft.Json.Linq.JObject` :param json: A `System.String` :return: A `Newtonsoft.Json.Linq.JObject`

FromObject (*System.Object*)

Creates a `Newtonsoft.Json.Linq.JObject` :param o: The object that will be used to create `Newtonsoft.Json.Linq.JObject` :return: A `Newtonsoft.Json.Linq.JObject`

FromObject (*System.Object, Newtonsoft.Json.JsonSerializer*)

Creates a `Newtonsoft.Json.Linq.JObject` :param o: The object that will be used to create `Newtonsoft.Json.Linq.JObject` :param jsonSerializer: The `Newtonsoft.Json.JsonSerializer` :return: A `Newtonsoft.Json.Linq.JObject`

WriteTo (*Newtonsoft.Json.JsonWriter, Newtonsoft.Json.JsonConverter[]*)

Writes this token to a `Newtonsoft.Json.JsonWriter` :param writer: A `Newtonsoft.Json.JsonWriter` :param converters: A collection of `Newtonsoft.Json.JsonConverter`

GetValue (*System.String*)

Gets the `Newtonsoft.Json.Linq.JToken` :param propertyName: Name of the property. :return: The `Newtonsoft.Json.Linq.JToken`

GetValue (*System.String, System.StringComparison*)

Gets the `Newtonsoft.Json.Linq.JToken` `System.StringComparison` :param propertyName: Name of the property. :param comparison: One of the enumeration values that specifies how the strings will be compared. :return: The `Newtonsoft.Json.Linq.JToken`

TryGetValue (*System.String, System.StringComparison, Newtonsoft.Json.Linq.JToken@*)

Tries to get the `Newtonsoft.Json.Linq.JToken` `System.StringComparison` :param propertyName: Name of the property. :param value: The value. :param comparison: One of the enumeration values that specifies how the strings will be compared. :return: true if a value was successfully retrieved; otherwise, false.

Add (*System.String, Newtonsoft.Json.Linq.JToken*)

Adds the specified property name. :param propertyName: Name of the property. :param value: The value.

Remove (*System.String*)

Removes the property with the specified name. :param propertyName: Name of the property. :return: true if item was successfully removed; otherwise, false.

TryGetValue (*System.String, Newtonsoft.Json.Linq.JToken@*)

Tries the get value. :param propertyName: Name of the property. :param value: The value. :return: true if a value was successfully retrieved; otherwise, false.

GetEnumerator ()

Returns an enumerator that iterates through the collection. :return: A `System.Collections.Generic.IEnumerator`1`

OnPropertyChanged (*System.String*)

Raises the `Newtonsoft.Json.Linq.JObject.PropertyChanged` :param propertyName: Name of the property.

GetMetaObject (*System.Linq.Expressions.Expression*)

Returns the `System.Dynamic.DynamicMetaObject` :param parameter: The expression tree representation of the runtime value. :return: The `System.Dynamic.DynamicMetaObject`

ChildrenTokens

Gets the container's children tokens.

Occurs when a property value changes.

Type

Gets the node type for this `Newtonsoft.Json.Linq.JToken`

Item (*System.Object*)

Gets the `Newtonsoft.Json.Linq.JToken` `Newtonsoft.Json.Linq.JToken`

Item (*System.String*)

Gets or sets the `Newtonsoft.Json.Linq.JToken`

1.59 JProperty

class `Newtonsoft.Json.Linq.JProperty`

Represents a JSON property.

Newtonsoft.Json.Linq.JProperty.#ctor (`Newtonsoft.Json.Linq.JProperty`)

Initializes a new instance of the `Newtonsoft.Json.Linq.JProperty` `Newtonsoft.Json.Linq.JProperty` :param other: A

`Newtonsoft.Json.Linq.JProperty`

Newtonsoft.Json.Linq.JProperty.#ctor (`System.String, System.Object[]`)

Initializes a new instance of the `Newtonsoft.Json.Linq.JProperty` :param name: The property name. :param content: The property content.

Newtonsoft.Json.Linq.JProperty.#ctor (`System.String, System.Object`)

Initializes a new instance of the `Newtonsoft.Json.Linq.JProperty` :param name: The property name. :param content: The property content.

WriteTo (`Newtonsoft.Json.JsonWriter, Newtonsoft.Json.JsonConverter[]`)

Writes this token to a `Newtonsoft.Json.JsonWriter` :param writer: A
`Newtonsoft.Json.JsonWriter` :param converters: A collection of
`Newtonsoft.Json.JsonConverter`

Load (*Newtonsoft.Json.JsonReader*)

Loads an `Newtonsoft.Json.Linq.JProperty` `Newtonsoft.Json.JsonReader`
:param reader: A
`Newtonsoft.Json.JsonReader` `Newtonsoft.Json.Linq.JProperty` :return: A
`Newtonsoft.Json.Linq.JProperty` `Newtonsoft.Json.JsonReader`

ChildrenTokens

Gets the container's children tokens.

Name

Gets the property name.

Value

Gets or sets the property value.

Type

Gets the node type for this `Newtonsoft.Json.Linq.JToken`

1.60 JRaw

class `Newtonsoft.Json.Linq.JRaw`
Represents a raw JSON string.

1.61 JValue

class `Newtonsoft.Json.Linq.JValue`

Represents a value in JSON (string, integer, date, etc).

Newtonsoft.Json.Linq.JValue.#ctor(Newtonsoft.Json.Linq.JValue)

Initializes a new instance of the `Newtonsoft.Json.Linq.JValue`
`Newtonsoft.Json.Linq.JValue` :param other: A
`Newtonsoft.Json.Linq.JValue`

Newtonsoft.Json.Linq.JValue.#ctor(System.Int64)

Initializes a new instance of the `Newtonsoft.Json.Linq.JValue` :param value: The value.

Newtonsoft.Json.Linq.JValue.#ctor(System.Decimal)

Initializes a new instance of the `Newtonsoft.Json.Linq.JValue` :param value: The value.

Newtonsoft.Json.Linq.JValue.#ctor(System.Char)

Initializes a new instance of the `Newtonsoft.Json.Linq.JValue` :param value: The value.

Newtonsoft.Json.Linq.JValue.#ctor(System.UInt64)

Initializes a new instance of the `Newtonsoft.Json.Linq.JValue` :param value: The value.

Newtonsoft.Json.Linq.JValue.#ctor(System.Double)

Initializes a new instance of the `Newtonsoft.Json.Linq.JValue` :param value: The value.

Newtonsoft.Json.Linq.JValue.#ctor(System.Single)

Initializes a new instance of the `Newtonsoft.Json.Linq.JValue` :param value: The value.

Newtonsoft.Json.Linq.JValue.#ctor(System.DateTime)

Initializes a new instance of the `Newtonsoft.Json.Linq.JValue` :param value: The value.

Newtonsoft.Json.Linq.JValue.#ctor(System.DateTimeOffset)

Initializes a new instance of the `Newtonsoft.Json.Linq.JValue` :param value: The value.

Newtonsoft.Json.Linq.JValue.#ctor(System.Boolean)

Initializes a new instance of the `Newtonsoft.Json.Linq.JValue` :param value: The value.

Newtonsoft.Json.Linq.JValue.#ctor(System.String)

Initializes a new instance of the `Newtonsoft.Json.Linq.JValue` :param value: The value.

Newtonsoft.Json.Linq.JValue.#ctor(System.Guid)

Initializes a new instance of the `Newtonsoft.Json.Linq.JValue` :param value: The value.

Newtonsoft.Json.Linq.JValue.#ctor(System.Uri)

Initializes a new instance of the `Newtonsoft.Json.Linq.JValue` :param value: The value.

Newtonsoft.Json.Linq.JValue.#ctor(System.TimeSpan)

Initializes a new instance of the `Newtonsoft.Json.Linq.JValue` :param value: The value.

Newtonsoft.Json.Linq.JValue.#ctor(System.Object)

Initializes a new instance of the `Newtonsoft.Json.Linq.JValue` :param value: The value.

CreateComment (*System.String*)

Creates a `Newtonsoft.Json.Linq.JValue` :param value: The value. :return: A `Newtonsoft.Json.Linq.JValue`

CreateString (*System.String*)

Creates a `Newtonsoft.Json.Linq.JValue` :param value: The value. :return: A `Newtonsoft.Json.Linq.JValue`

CreateNull ()

Creates a `Newtonsoft.Json.Linq.JValue` :return: A `Newtonsoft.Json.Linq.JValue`

CreateUndefined ()

Creates a `Newtonsoft.Json.Linq.JValue` :return: A `Newtonsoft.Json.Linq.JValue`

WriteTo (*Newtonsoft.Json.JsonWriter*, *Newtonsoft.Json.JsonConverter[]*)

Writes this token to a `Newtonsoft.Json.JsonWriter` :param writer: A

`Newtonsoft.Json.JsonWriter` :param converters: A collection of `Newtonsoft.Json.JsonConverter`

Equals (*Newtonsoft.Json.Linq.JValue*)

Indicates whether the current object is equal to another object of the same type. :return: true if the current object is equal to the :param other: An object to compare with this object.

Equals (*System.Object*)

Determines whether the specified *System.Object* *System.Object* :param obj: The *System.Object* *System.Object* :return: true if the specified *System.Object* *System.Object*

GetHashCode ()

Serves as a hash function for a particular type. :return: A hash code for the current *System.Object*

ToString ()

Returns a *System.String* :return: A *System.String*

ToString (*System.String*)

Returns a *System.String* :param format: The format. :return: A *System.String*

ToString (*System.IFormatProvider*)

Returns a *System.String* :param formatProvider: The format provider. :return: A *System.String*

ToString (*System.String*, *System.IFormatProvider*)

Returns a *System.String* :param format: The format. :param formatProvider: The format provider. :return: A *System.String*

GetMetaObject (*System.Linq.Expressions.Expression*)

Returns the *System.Dynamic.DynamicMetaObject* :param parameter: The expression tree representation of the runtime value. :return: The *System.Dynamic.DynamicMetaObject*

CompareTo (*Newtonsoft.Json.Linq.JValue*)

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. :param obj: An object to compare with this instance. :return: 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

HasValues

Gets a value indicating whether this token has child tokens.

Type

Gets the node type for this *Newtonsoft.Json.Linq.JToken*

Value

Gets or sets the underlying token value.

Newtonsoft.Json.Linq.JRaw.#ctor (*Newtonsoft.Json.Linq.JRaw*)

Initializes a new instance of the *Newtonsoft.Json.Linq.JRaw* *Newtonsoft.Json.Linq.JRaw* :param other: A

Newtonsoft.Json.Linq.JRaw

Newtonsoft.Json.Linq.JRaw.#ctor(System.Object)

Initializes a new instance of the `Newtonsoft.Json.Linq.JRaw` :param rawJson: The raw json.

Newtonsoft.Json.Linq.JRaw.Create(Newtonsoft.Json.JsonReader)

Creates an instance of `Newtonsoft.Json.Linq.JRaw` :param reader: The reader. :return: An instance of `Newtonsoft.Json.Linq.JRaw`

1.62 JsonMergeSettings

class Newtonsoft.Json.Linq.JsonMergeSettings

Specifies the settings used when merging JSON.

MergeArrayHandling

Gets or sets the method used when merging JSON arrays.

1.63 JTokenEqualityComparer

class Newtonsoft.Json.Linq.JTokenEqualityComparer

Compares tokens to determine whether they are equal.

Equals (*Newtonsoft.Json.Linq.JToken, Newtonsoft.Json.Linq.JToken*)

Determines whether the specified objects are equal. :param x: The first object of type

`Newtonsoft.Json.Linq.JToken` :param y: The second object of type `Newtonsoft.Json.Linq.JToken` :return: true if the specified objects are equal; otherwise, false.

GetHashCode (*Newtonsoft.Json.Linq.JToken*)

Returns a hash code for the specified object. :param obj: The

`System.Object` :return: A hash code for the specified object.

1.64 JTokenReader

class Newtonsoft.Json.Linq.JTokenReader

Represents a reader that provides fast, non-cached, forward-only access to serialized JSON data.

Newtonsoft.Json.Linq.JTokenReader.#ctor(Newtonsoft.Json.Linq.JToken)

Initializes a new instance of the `Newtonsoft.Json.Linq.JTokenReader` :param token: The token to read from.

ReadAsBytes ()

Reads the next JSON token from the stream as a `System.Byte` :return: A `System.Byte`

ReadAsDecimal ()

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.Nullable`1`

ReadAsInt32 ()

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.Nullable`1`

ReadAsString()

Reads the next JSON token from the stream as a `System.String` :return: A `System.String`

ReadAsDateTime()

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.String`

ReadAsDateTimeOffset()

Reads the next JSON token from the stream as a `System.Nullable`1` :return: A `System.Nullable`1`

Read()

Reads the next JSON token from the stream. :return: true if the next token was read successfully; false if there are no more tokens to read.

CurrentToken

Gets the `Newtonsoft.Json.Linq.JToken`

Path

Gets the path of the current JSON token.

1.65 JTokenType

class `Newtonsoft.Json.Linq.JTokenType`

Specifies the type of token.

None

No token type has been set.

Object

A JSON object.

Array

A JSON array.

Constructor

A JSON constructor.

Property

A JSON object property.

Comment

A comment.

Integer

An integer value.

Float

A float value.

String

A string value.

Boolean

A boolean value.

Null

A null value.

Undefined

An undefined value.

Date

A date value.

Raw

A raw JSON value.

Bytes

A collection of bytes value.

Guid

A Guid value.

Uri

A Uri value.

TimeSpan

A TimeSpan value.

1.66 JTokenWriter

class `Newtonsoft.Json.Linq.JTokenWriter`

Represents a writer that provides a fast, non-cached, forward-only way of generating JSON data.

Newtonsoft.Json.Linq.JTokenWriter.#ctor(Newtonsoft.Json.Linq.JContainer)

Initializes a new instance of the `Newtonsoft.Json.Linq.JTokenWriter` `Newtonsoft.Json.Linq.JContainer` :param container: The container being written to.

Newtonsoft.Json.Linq.JTokenWriter.#ctor

Initializes a new instance of the `Newtonsoft.Json.Linq.JTokenWriter`

Flush()

Flushes whatever is in the buffer to the underlying streams and also flushes the underlying stream.

Close()

Closes this stream and the underlying stream.

WriteStartObject()

Writes the beginning of a Json object.

WriteStartArray()

Writes the beginning of a Json array.

WriteStartConstructor (*System.String*)

Writes the start of a constructor with the given name. :param name: The name of the constructor.

WriteEnd (*Newtonsoft.Json.JsonToken*)

Writes the end. :param token: The token.

WritePropertyName (*System.String*)

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

WriteValue (*System.Object*)

Writes a *System.Object* :param value: The

System.Object

WriteNull ()

Writes a null value.

WriteUndefined ()

Writes an undefined value.

WriteRaw (*System.String*)

Writes raw JSON. :param json: The raw JSON to write.

WriteComment (*System.String*)

Writes out a comment :param text: Text to place inside the comment.

WriteValue (*System.String*)

Writes a *System.String* :param value: The

System.String

WriteValue (*System.Int32*)

Writes a *System.Int32* :param value: The

System.Int32

WriteValue (*System.UInt32*)

Writes a *System.UInt32* :param value: The

System.UInt32

WriteValue (*System.Int64*)

Writes a *System.Int64* :param value: The

System.Int64

WriteValue (*System.UInt64*)

Writes a *System.UInt64* :param value: The

System.UInt64

WriteValue (*System.Single*)

Writes a *System.Single* :param value: The

System.Single

WriteValue (*System.Double*)

Writes a `System.Double` :param value: The
`System.Double`

WriteValue (*System.Boolean*)

Writes a `System.Boolean` :param value: The
`System.Boolean`

WriteValue (*System.Int16*)

Writes a `System.Int16` :param value: The
`System.Int16`

WriteValue (*System.UInt16*)

Writes a `System.UInt16` :param value: The
`System.UInt16`

WriteValue (*System.Char*)

Writes a `System.Char` :param value: The
`System.Char`

WriteValue (*System.Byte*)

Writes a `System.Byte` :param value: The
`System.Byte`

WriteValue (*System.SByte*)

Writes a `System.SByte` :param value: The
`System.SByte`

WriteValue (*System.Decimal*)

Writes a `System.Decimal` :param value: The
`System.Decimal`

WriteValue (*System.DateTime*)

Writes a `System.DateTime` :param value: The
`System.DateTime`

WriteValue (*System.DateTimeOffset*)

Writes a `System.DateTimeOffset` :param value: The
`System.DateTimeOffset`

WriteValue (*System.Byte[]*)

Writes a `System.Byte` :param value: The
`System.Byte`

WriteValue (*System.TimeSpan*)

Writes a `System.TimeSpan` :param value: The
`System.TimeSpan`

WriteValue (*System.Guid*)

Writes a `System.Guid` :param value: The

`System.Guid`

WriteValue (*System.Uri*)

Writes a `System.Uri` :param value: The

`System.Uri`

CurrentToken

Gets the `Newtonsoft.Json.Linq.JToken`

Token

Gets the token being written.

1.67 MergeArrayHandling

class `Newtonsoft.Json.Linq.MergeArrayHandling`

Specifies how JSON arrays are merged together.

Concat

Concatenate arrays.

Union

Union arrays, skipping items that already exist.

Replace

Replace all array items.

Merge

Merge array items together, matched by index.

1.68 MemberSerialization

class `Newtonsoft.Json.MemberSerialization`

Specifies the member serialization options for the `Newtonsoft.Json.JsonSerializer`

OptOut

All public members are serialized by default. Members can be excluded using `Newtonsoft.Json.JsonIgnoreAttribute` `NonSerializedAttribute`

OptIn

Only members must be marked with `Newtonsoft.Json.JsonPropertyAttribute` `System.Runtime.Serialization.DataMemberAttribute` `System.Runtime.Serialization.DataContractAttribute`

Fields

All public and private fields are serialized. Members can be excluded using `Newtonsoft.Json.JsonIgnoreAttribute` `NonSerializedAttribute` `SerializableAttribute` `Newtonsoft.Json.Serialization.DefaultContractResolver`

1.69 MetadataPropertyHandling

class `Newtonsoft.Json.MetadataPropertyHandling`

Specifies metadata property handling options for the `Newtonsoft.Json.JsonSerializer`

Default

Read metadata properties located at the start of a JSON object.

ReadAhead

Read metadata properties located anywhere in a JSON object. Note that this setting will impact performance.

Ignore

Do not try to read metadata properties.

1.70 MissingMemberHandling

class `Newtonsoft.Json.MissingMemberHandling`

Specifies missing member handling options for the `Newtonsoft.Json.JsonSerializer`

Ignore

Ignore a missing member and do not attempt to deserialize it.

Error

Throw a `Newtonsoft.Json.JsonSerializationException`

1.71 NullValueHandling

class `Newtonsoft.Json.NullValueHandling`

Specifies null value handling options for the `Newtonsoft.Json.JsonSerializer`

Include

Include null values when serializing and deserializing objects.

Ignore

Ignore null values when serializing and deserializing objects.

1.72 ObjectCreationHandling

class `Newtonsoft.Json.ObjectCreationHandling`

Specifies how object creation is handled by the `Newtonsoft.Json.JsonSerializer`

Auto

Reuse existing objects, create new objects when needed.

Reuse

Only reuse existing objects.

Replace

Always create new objects.

1.73 PreserveReferencesHandling

class `Newtonsoft.Json.PreserveReferencesHandling`

Specifies reference handling options for the `Newtonsoft.Json.JsonSerializer`

None

Do not preserve references when serializing types.

Objects

Preserve references when serializing into a JSON object structure.

Arrays

Preserve references when serializing into a JSON array structure.

All

Preserve references when serializing.

1.74 ReferenceLoopHandling

class `Newtonsoft.Json.ReferenceLoopHandling`

Specifies reference loop handling options for the `Newtonsoft.Json.JsonSerializer`

Error

Throw a `Newtonsoft.Json.JsonSerializationException`

Ignore

Ignore loop references and do not serialize.

Serialize

Serialize loop references.

1.75 Required

class `Newtonsoft.Json.Required`

Indicating whether a property is required.

Default

The property is not required. The default state.

AllowNull

The property must be defined in JSON but can be a null value.

Always

The property must be defined in JSON and cannot be a null value.

1.76 Extensions

class `Newtonsoft.Json.Schema.Extensions`

Contains the JSON schema extension methods.

IsValid (`Newtonsoft.Json.Linq.JToken`, `Newtonsoft.Json.Schema.JsonSchema`)

Determines whether the `Newtonsoft.Json.Linq.JToken` :param source: The source

`Newtonsoft.Json.Linq.JToken` :param schema: The schema to test with.

return `Newtonsoft.Json.Linq.JToken`

IsValid (`Newtonsoft.Json.Linq.JToken`, `Newtonsoft.Json.Schema.JsonSchema`, `System.Collections.Generic.IList<System.String>` @)

Determines whether the `Newtonsoft.Json.Linq.JToken` :param source: The source

`Newtonsoft.Json.Linq.JToken` :param schema: The schema to test with.

param errorMessages When this method returns, contains any error messages generated while validating.

return `Newtonsoft.Json.Linq.JToken`

Validate (`Newtonsoft.Json.Linq.JToken`, `Newtonsoft.Json.Schema.JsonSchema`)

Validates the specified `Newtonsoft.Json.Linq.JToken` :param source: The source

`Newtonsoft.Json.Linq.JToken` :param schema: The schema to test with.

Validate (`Newtonsoft.Json.Linq.JToken`, `Newtonsoft.Json.Schema.JsonSchema`, `Newtonsoft.Json.Schema.ValidationEventHandler`)

Validates the specified `Newtonsoft.Json.Linq.JToken` :param source: The source

`Newtonsoft.Json.Linq.JToken` :param schema: The schema to test with.

param validationEventHandler The validation event handler.

1.77 JsonSchema

class `Newtonsoft.Json.Schema.JsonSchema`

An in-memory representation of a JSON Schema.

Newtonsoft.Json.Schema.JsonSchema.#ctor

Initializes a new instance of the `Newtonsoft.Json.Schema.JsonSchema`

Read (`Newtonsoft.Json.JsonReader`)

Reads a `Newtonsoft.Json.Schema.JsonSchema` `Newtonsoft.Json.JsonReader` :param reader: The

`Newtonsoft.Json.JsonReader` :return: The `Newtonsoft.Json.Schema.JsonSchema`

Read (`Newtonsoft.Json.JsonReader`, `Newtonsoft.Json.Schema.JsonSchemaResolver`)

Reads a `Newtonsoft.Json.Schema.JsonSchema` `Newtonsoft.Json.JsonReader` :param reader: The

`Newtonsoft.Json.JsonReader` :param resolver: The `Newtonsoft.Json.Schema.JsonSchemaResolver`
:return: The `Newtonsoft.Json.Schema.JsonSchema`

Parse (*System.String*)

Load a `Newtonsoft.Json.Schema.JsonSchema` :param json: A
`System.String` :return: A `Newtonsoft.Json.Schema.JsonSchema`

Parse (*System.String, Newtonsoft.Json.Schema.JsonSchemaResolver*)

Parses the specified json. :param json: The json. :param resolver: The resolver. :return: A
`Newtonsoft.Json.Schema.JsonSchema`

WriteTo (*Newtonsoft.Json.JsonWriter*)

Writes this schema to a `Newtonsoft.Json.JsonWriter` :param writer: A
`Newtonsoft.Json.JsonWriter`

WriteTo (*Newtonsoft.Json.JsonWriter, Newtonsoft.Json.Schema.JsonSchemaResolver*)

Writes this schema to a `Newtonsoft.Json.JsonWriter`
`Newtonsoft.Json.Schema.JsonSchemaResolver` :param writer: A

`Newtonsoft.Json.JsonWriter` :param resolver: The resolver used.

ToString ()

Returns a `System.String` `System.Object` :return: A `System.String` `System.Object`

Id

Gets or sets the id.

Title

Gets or sets the title.

Required

Gets or sets whether the object is required.

ReadOnly

Gets or sets whether the object is read only.

Hidden

Gets or sets whether the object is visible to users.

Transient

Gets or sets whether the object is transient.

Description

Gets or sets the description of the object.

Type

Gets or sets the types of values allowed by the object.

Pattern

Gets or sets the pattern.

MinimumLength

Gets or sets the minimum length.

MaxLength

Gets or sets the maximum length.

DivisibleBy

Gets or sets a number that the value should be divisible by.

Minimum

Gets or sets the minimum.

Maximum

Gets or sets the maximum.

ExclusiveMinimum

Gets or sets a flag indicating whether the value can not equal the number defined by the “minimum” attribute.

ExclusiveMaximum

Gets or sets a flag indicating whether the value can not equal the number defined by the “maximum” attribute.

MinimumItems

Gets or sets the minimum number of items.

MaximumItems

Gets or sets the maximum number of items.

Items

Gets or sets the `Newtonsoft.Json.Schema.JsonSchema` items.

PositionalItemsValidation

Gets or sets a value indicating whether items in an array are validated using the `Newtonsoft.Json.Schema.JsonSchema.Items`.

AdditionalItems

Gets or sets the `Newtonsoft.Json.Schema.JsonSchema` additional items.

AllowAdditionalItems

Gets or sets a value indicating whether additional items are allowed.

UniqueItems

Gets or sets whether the array items must be unique.

Properties

Gets or sets the `Newtonsoft.Json.Schema.JsonSchema` properties.

AdditionalProperties

Gets or sets the `Newtonsoft.Json.Schema.JsonSchema` additional properties.

PatternProperties

Gets or sets the pattern properties.

AllowAdditionalProperties

Gets or sets a value indicating whether additional properties are allowed.

Requires

Gets or sets the required property if this property is present.

Enum

Gets or sets the a collection of valid enum values allowed.

Disallow

Gets or sets disallowed types.

Default

Gets or sets the default value.

Extends

Gets or sets the collection of `Newtonsoft.Json.Schema.JsonSchema`
`Newtonsoft.Json.Schema.JsonSchema`

Format

Gets or sets the format.

1.78 JsonSchemaException

class `Newtonsoft.Json.Schema.JsonSchemaException`

Returns detailed information about the schema exception.

`Newtonsoft.Json.Schema.JsonSchemaException.#ctor`

Initializes a new instance of the `Newtonsoft.Json.Schema.JsonSchemaException`

`Newtonsoft.Json.Schema.JsonSchemaException.#ctor(System.String)`

Initializes a new instance of the `Newtonsoft.Json.Schema.JsonSchemaException` :param message: The error message that explains the reason for the exception.

`Newtonsoft.Json.Schema.JsonSchemaException.#ctor(System.String, System.Exception)`

Initializes a new instance of the `Newtonsoft.Json.Schema.JsonSchemaException` :param message: The error message that explains the reason for the exception. :param innerException: The exception that is the cause of the current exception, or a null reference (Nothing in Visual Basic) if no inner exception is specified.

`LineNumber`

Gets the line number indicating where the error occurred.

`LinePosition`

Gets the line position indicating where the error occurred.

`Path`

Gets the path to the JSON where the error occurred.

1.79 JsonSchemaGenerator

class `Newtonsoft.Json.Schema.JsonSchemaGenerator`

Generates a `Newtonsoft.Json.Schema.JsonSchema` `System.Type`

Generate (*System.Type*)

Generate a `Newtonsoft.Json.Schema.JsonSchema` :param type: The type to generate a

`Newtonsoft.Json.Schema.JsonSchema` :return: A `Newtonsoft.Json.Schema.JsonSchema`

Generate (*System.Type, Newtonsoft.Json.Schema.JsonSchemaResolver*)

Generate a `Newtonsoft.Json.Schema.JsonSchema` :param type: The type to generate a

`Newtonsoft.Json.Schema.JsonSchema` :param resolver: The `Newtonsoft.Json.Schema.JsonSchemaResolver`

:return: A `Newtonsoft.Json.Schema.JsonSchema`

Generate (*System.Type, System.Boolean*)

Generate a `Newtonsoft.Json.Schema.JsonSchema` :param type: The type to generate a

`Newtonsoft.Json.Schema.JsonSchema` :param rootSchemaNullable: Specify whether the generated root `Newtonsoft.Json.Schema.JsonSchema` :return: A `Newtonsoft.Json.Schema.JsonSchema`

Generate (*System.Type, Newtonsoft.Json.Schema.JsonSchemaResolver, System.Boolean*)

Generate a `Newtonsoft.Json.Schema.JsonSchema` :param type: The type to generate a

`Newtonsoft.Json.Schema.JsonSchema` :param resolver: The `Newtonsoft.Json.Schema.JsonSchemaResolver`

:param rootSchemaNullable: Specify whether the generated root `Newtonsoft.Json.Schema.JsonSchema`

:return: A `Newtonsoft.Json.Schema.JsonSchema`

UndefinedSchemaIdHandling

Gets or sets how undefined schemas are handled by the serializer.

ContractResolver

Gets or sets the contract resolver.

1.80 JsonSchemaResolver

class `Newtonsoft.Json.Schema.JsonSchemaResolver`

Resolves `Newtonsoft.Json.Schema.JsonSchema`

Newtonsoft.Json.Schema.JsonSchemaResolver.#ctor

Initializes a new instance of the `Newtonsoft.Json.Schema.JsonSchemaResolver`

GetSchema (*System.String*)

Gets a `Newtonsoft.Json.Schema.JsonSchema` :param reference: The id. :return: A `Newtonsoft.Json.Schema.JsonSchema`

LoadedSchemas

Gets or sets the loaded schemas.

1.81 JsonSchemaType

class `Newtonsoft.Json.Schema.JsonSchemaType`

The value types allowed by the `Newtonsoft.Json.Schema.JsonSchema`

None

No type specified.

String

String type.

Float

Float type.

Integer

Integer type.

Boolean

Boolean type.

Object

Object type.

Array

Array type.

Null

Null type.

Any

Any type.

1.82 UndefinedSchemaIdHandling

class `Newtonsoft.Json.Schema.UndefinedSchemaIdHandling`

Specifies undefined schema Id handling options for the `Newtonsoft.Json.Schema.JsonSchemaGenerator`

None

Do not infer a schema Id.

UseTypeName

Use the .NET type name as the schema Id.

UseAssemblyQualifiedName

Use the assembly qualified .NET type name as the schema Id.

1.83 ValidationEventArgs

class `Newtonsoft.Json.Schema.ValidationEventArgs`

Returns detailed information related to the `Newtonsoft.Json.Schema.ValidationEventHandler`

Exception

Gets the `Newtonsoft.Json.Schema.JsonSchemaException`

Path

Gets the path of the JSON location where the validation error occurred.

Message

Gets the text description corresponding to the validation error.

1.84 ValidationEventHandler

class `Newtonsoft.Json.Schema.ValidationEventHandler`

Represents the callback method that will handle JSON schema validation events and the `Newtonsoft.Json.Schema.ValidationEventArgs`

1.85 SerializationBinder

class `Newtonsoft.Json.SerializationBinder`

Allows users to control class loading and mandate what class to load.

BindToType (*System.String*, *System.String*)

When overridden in a derived class, controls the binding of a serialized object to a type. :param assemblyName: Specifies the

System.Reflection.Assembly :param typeName: Specifies the *System.Type* :return: The type of the object the formatter creates a new instance of.

BindToName (*System.Type*, *System.String@*, *System.String@*)

When overridden in a derived class, controls the binding of a serialized object to a type. :param serializedType: The type of the object the formatter creates a new instance of. :param assemblyName: Specifies the

System.Reflection.Assembly :param typeName: Specifies the *System.Type*

1.86 CamelCasePropertyNamesContractResolver

class `Newtonsoft.Json.Serialization.CamelCasePropertyNamesContractResolver`

Resolves member mappings for a type, camel casing property names.

1.87 DefaultContractResolver

class `Newtonsoft.Json.Serialization.DefaultContractResolver`

Used by `Newtonsoft.Json.JsonSerializer` `Newtonsoft.Json.Serialization.JsonContract`
`System.Type`

1.88 IContractResolver

class `Newtonsoft.Json.Serialization.IContractResolver`

Used by `Newtonsoft.Json.JsonSerializer` `Newtonsoft.Json.Serialization.JsonContract`
`System.Type`

ResolveContract (*System.Type*)

Resolves the contract for a given type. :param type: The type to resolve a contract for. :return: The contract for a given type.

Newtonsoft.Json.Serialization.DefaultContractResolver.#ctor

Initializes a new instance of the `Newtonsoft.Json.Serialization.DefaultContractResolver`

Newtonsoft.Json.Serialization.DefaultContractResolver.#ctor(System.Boolean)

Initializes a new instance of the `Newtonsoft.Json.Serialization.DefaultContractResolver` :param s
If set to

`Newtonsoft.Json.Serialization.DefaultContractResolver`

`Newtonsoft.Json.Serialization.DefaultContractResolver`

`Newtonsoft.Json.JsonSerializer`

`Newtonsoft.Json.Serialization.DefaultContractResolver.ResolveContract` (*System.Type*)

Resolves the contract for a given type. :param type: The type to resolve a contract for. :return: The contract for a given type.

`Newtonsoft.Json.Serialization.DefaultContractResolver.GetSerializableMembers` (*System.Type*)

Gets the serializable members for the type. :param objectType: The type to get serializable members for. :return: The serializable members for the type.

`Newtonsoft.Json.Serialization.DefaultContractResolver.CreateObjectContract` (*System.Type*)

Creates a `Newtonsoft.Json.Serialization.JsonObjectContract` :param object-Type: Type of the object. :return: A `Newtonsoft.Json.Serialization.JsonObjectContract`

`Newtonsoft.Json.Serialization.DefaultContractResolver.CreateConstructorParameters` (*System.Type*)

Creates the constructor parameters. :param constructor: The constructor to create properties for. :param memberProperties: The type's member properties. :return: Properties for the given `System.Reflection.ConstructorInfo`

`Newtonsoft.Json.Serialization.DefaultContractResolver.CreatePropertyFromConstructor`

Creates a `Newtonsoft.Json.Serialization.JsonProperty`
`System.Reflection.ParameterInfo` :param `matchingMemberProperty`: The
 matching member property. :param `parameterInfo`: The constructor param-
 eter. :return: A created `Newtonsoft.Json.Serialization.JsonProperty`
`System.Reflection.ParameterInfo`

`Newtonsoft.Json.Serialization.DefaultContractResolver.ResolveContractConverter` (*System.Type*)

Resolves the default `Newtonsoft.Json.JsonConverter` :param `objectType`: Type of the ob-
 ject. :return: The contract's default `Newtonsoft.Json.JsonConverter`

`Newtonsoft.Json.Serialization.DefaultContractResolver.CreateDictionaryContract` (*System.Type*)

Creates a `Newtonsoft.Json.Serialization.JsonDictionaryContract` :param ob-
 jectType: Type of the object. :return: A `Newtonsoft.Json.Serialization.JsonDictionaryContract`

`Newtonsoft.Json.Serialization.DefaultContractResolver.CreateArrayContract` (*System.Type*)

Creates a `Newtonsoft.Json.Serialization.JsonArrayContract` :param `objectType`:
 Type of the object. :return: A `Newtonsoft.Json.Serialization.JsonArrayContract`

`Newtonsoft.Json.Serialization.DefaultContractResolver.CreatePrimitiveContract` (*System.Type*)

Creates a `Newtonsoft.Json.Serialization.JsonPrimitiveContract` :param ob-
 jectType: Type of the object. :return: A `Newtonsoft.Json.Serialization.JsonPrimitiveContract`

`Newtonsoft.Json.Serialization.DefaultContractResolver.CreateLinqContract` (*System.Type*)

Creates a `Newtonsoft.Json.Serialization.JsonLinqContract` :param `objectType`:
 Type of the object. :return: A `Newtonsoft.Json.Serialization.JsonLinqContract`

`Newtonsoft.Json.Serialization.DefaultContractResolver.CreateDynamicContract` (*System.Type*)

Creates a `Newtonsoft.Json.Serialization.JsonDynamicContract` :param ob-
 jectType: Type of the object. :return: A `Newtonsoft.Json.Serialization.JsonDynamicContract`

`Newtonsoft.Json.Serialization.DefaultContractResolver.CreateStringContract` (*System.Type*)

Creates a `Newtonsoft.Json.Serialization.JsonStringContract` :param ob-
 jectType: Type of the object. :return: A `Newtonsoft.Json.Serialization.JsonStringContract`

`Newtonsoft.Json.Serialization.DefaultContractResolver.CreateContract` (*System.Type*)

Determines which contract type is created for the given type. :param `objectType`: Type of the object.
 :return: A `Newtonsoft.Json.Serialization.JsonContract`

`Newtonsoft.Json.Serialization.DefaultContractResolver.CreateProperties` (*System.Type*,
Newtonsoft.Json.MemberSerialization)

Creates properties for the given `Newtonsoft.Json.Serialization.JsonContract`
 :param `type`: The type to create properties for. :param `memberSerialization`:
 The member serialization mode for the type. :return: Properties for the given
`Newtonsoft.Json.Serialization.JsonContract`

`Newtonsoft.Json.Serialization.DefaultContractResolver.CreateMemberValueProvider` (*System.Type*)

Creates the `Newtonsoft.Json.Serialization.IValueProvider` :param member: The member. :return: The `Newtonsoft.Json.Serialization.IValueProvider`

`Newtonsoft.Json.Serialization.DefaultContractResolver.CreateProperty` (*System.Reflection.MemberInfo*, *Newtonsoft.Json.MemberSerialization*)

Creates a `Newtonsoft.Json.Serialization.JsonProperty` :param memberSerialization: The member's parent

`Newtonsoft.Json.MemberSerialization` :param member: The member to create a `Newtonsoft.Json.Serialization.JsonProperty` :return: A created `Newtonsoft.Json.Serialization.JsonProperty` `System.Reflection.MemberInfo`

`Newtonsoft.Json.Serialization.DefaultContractResolver.ResolvePropertyName` (*System.String*)

Resolves the name of the property. :param propertyName: Name of the property. :return: Name of the property.

`Newtonsoft.Json.Serialization.DefaultContractResolver.GetResolvedPropertyName` (*System.String*)

Gets the resolved name of the property. :param propertyName: Name of the property. :return: Name of the property.

`Newtonsoft.Json.Serialization.DefaultContractResolver.DynamicCodeGeneration`

Gets a value indicating whether members are being get and set using dynamic code generation.
This value is determined by the runtime permissions available.

`Newtonsoft.Json.Serialization.DefaultContractResolver.SerializeCompilerGeneratedMembers`

Gets or sets a value indicating whether compiler generated members should be serialized.

`Newtonsoft.Json.Serialization.CamelCasePropertyNamesContractResolver.#ctor`

Initializes a new instance of the `Newtonsoft.Json.Serialization.CamelCasePropertyNamesContractResolver`

`Newtonsoft.Json.Serialization.CamelCasePropertyNamesContractResolver.ResolvePropertyName`

Resolves the name of the property. :param propertyName: Name of the property. :return: The property name camel cased.

1.89 ExpressionValueProvider

class `Newtonsoft.Json.Serialization.ExpressionValueProvider`

Get and set values for a `System.Reflection.MemberInfo`

1.90 IValueProvider

class `Newtonsoft.Json.Serialization.IValueProvider`

Provides methods to get and set values.

SetValue (*System.Object*, *System.Object*)

Sets the value. :param target: The target to set the value on. :param value: The value to set on the target.

GetValue (*System.Object*)

Gets the value. :param target: The target to get the value from. :return: The value.

Newtonsoft.Json.Serialization.ExpressionValueProvider.#ctor (*System.Reflection.MemberInfo*)

Initializes a new instance of the *Newtonsoft.Json.Serialization.ExpressionValueProvider*
:param memberInfo: The member info.

Newtonsoft.Json.Serialization.ExpressionValueProvider.SetValue (*System.Object*,
System.Object)

Sets the value. :param target: The target to set the value on. :param value: The value to set on the target.

Newtonsoft.Json.Serialization.ExpressionValueProvider.GetValue (*System.Object*)

Gets the value. :param target: The target to get the value from. :return: The value.

1.91 IReferenceResolver

class *Newtonsoft.Json.Serialization.IReferenceResolver*

Used to resolve references when serializing and deserializing JSON by the *Newtonsoft.Json.JsonSerializer*

ResolveReference (*System.Object*, *System.String*)

Resolves a reference to its object. :param context: The serialization context. :param reference: The reference to resolve. :return: The object that

GetReference (*System.Object*, *System.Object*)

Gets the reference for the specified object. :param context: The serialization context. :param value: The object to get a reference for. :return: The reference to the object.

IsReferenced (*System.Object*, *System.Object*)

Determines whether the specified object is referenced. :param context: The serialization context. :param value: The object to test for a reference. :return:

AddReference (*System.Object*, *System.String*, *System.Object*)

Adds a reference to the specified object. :param context: The serialization context. :param reference: The reference. :param value: The object to reference.

1.92 DefaultSerializationBinder

class *Newtonsoft.Json.Serialization.DefaultSerializationBinder*

The default serialization binder used when resolving and loading classes from type names.

BindToType (*System.String*, *System.String*)

When overridden in a derived class, controls the binding of a serialized object to a type. :param assemblyName: Specifies the

System.Reflection.Assembly :param typeName: Specifies the *System.Type* :return: The type of the object the formatter creates a new instance of.

BindToName (*System.Type*, *System.String@*, *System.String@*)

When overridden in a derived class, controls the binding of a serialized object to a type. :param serializedType: The type of the object the formatter creates a new instance of. :param assemblyName: Specifies the

`System.Reflection.Assembly` :param typeName: Specifies the `System.Type`

1.93 ErrorContext

class `Newtonsoft.Json.Serialization.ErrorContext`

Provides information surrounding an error.

Error

Gets the error.

OriginalObject

Gets the original object that caused the error.

Member

Gets the member that caused the error.

Path

Gets the path of the JSON location where the error occurred.

Handled

Gets or sets a value indicating whether this `Newtonsoft.Json.Serialization.ErrorContext`

1.94 ErrorEventArgs

class `Newtonsoft.Json.Serialization.ErrorEventArgs`

Provides data for the Error event.

Newtonsoft.Json.Serialization.ErrorEventArgs.#ctor (`System.Object`, `Newtonsoft.Json.Seria`

Initializes a new instance of the `Newtonsoft.Json.Serialization.ErrorEventArgs` :param currentObject: The current object. :param errorContext: The error context.

CurrentObject

Gets the current object the error event is being raised against.

ErrorContext

Gets the error context.

1.95 IAttributeProvider

class `Newtonsoft.Json.Serialization.IAttributeProvider`

Provides methods to get attributes.

GetAttributes (`System.Boolean`)

Returns a collection of all of the attributes, or an empty collection if there are no attributes. :param inherit: When true, look up the hierarchy chain for the inherited custom attribute. :return: A collection of `System.Attribute`

GetAttributes (*System.Type, System.Boolean*)

Returns a collection of attributes, identified by type, or an empty collection if there are no attributes. :param attributeType: The type of the attributes. :param inherit: When true, look up the hierarchy chain for the inherited custom attribute. :return: A collection of `System.Attribute`

1.96 ITraceWriter

class `Newtonsoft.Json.Serialization.ITraceWriter`

Represents a trace writer.

Trace (*Newtonsoft.Json.TraceLevel, System.String, System.Exception*)

Writes the specified trace level, message and optional exception. :param level: The

Newtonsoft.Json.TraceLevel :param message: The trace message.

param ex The trace exception. This parameter is optional.

LevelFilter

Gets the `Newtonsoft.Json.TraceLevel` `Newtonsoft.Json.TraceLevel`

1.97 JsonArrayContract

class `Newtonsoft.Json.Serialization.JsonArrayContract`

Contract details for a `System.Type` `Newtonsoft.Json.JsonSerializer`

1.98 JsonContainerContract

class `Newtonsoft.Json.Serialization.JsonContainerContract`

Contract details for a `System.Type` `Newtonsoft.Json.JsonSerializer`

1.99 JsonContract

class `Newtonsoft.Json.Serialization.JsonContract`

Contract details for a `System.Type` `Newtonsoft.Json.JsonSerializer`

UnderlyingType

Gets the underlying type for the contract.

CreatedType

Gets or sets the type created during deserialization.

IsReference

Gets or sets whether this type contract is serialized as a reference.

Converter

Gets or sets the default `Newtonsoft.Json.JsonConverter`

OnDeserializedCallbacks

Gets or sets all methods called immediately after deserialization of the object.

OnDeserializingCallbacks

Gets or sets all methods called during deserialization of the object.

OnSerializedCallbacks

Gets or sets all methods called after serialization of the object graph.

OnSerializingCallbacks

Gets or sets all methods called before serialization of the object.

OnErrorCallbacks

Gets or sets all method called when an error is thrown during the serialization of the object.

OnDeserialized

Gets or sets the method called immediately after deserialization of the object.

OnDeserializing

Gets or sets the method called during deserialization of the object.

OnSerialized

Gets or sets the method called after serialization of the object graph.

OnSerializing

Gets or sets the method called before serialization of the object.

OnError

Gets or sets the method called when an error is thrown during the serialization of the object.

DefaultCreator

Gets or sets the default creator method used to create the object.

DefaultCreatorNonPublic

Gets or sets a value indicating whether the default creator is non public.

Newtonsoft.Json.Serialization.JsonContainerContract.#ctor(System.Type)

Initializes a new instance of the `Newtonsoft.Json.Serialization.JsonContainerContract`
:param underlyingType: The underlying type for the contract.

Newtonsoft.Json.Serialization.JsonContainerContract.ItemConverter

Gets or sets the default collection items `Newtonsoft.Json.JsonConverter`

Newtonsoft.Json.Serialization.JsonContainerContract.ItemIsReference

Gets or sets a value indicating whether the collection items preserve object references.

Newtonsoft.Json.Serialization.JsonContainerContract.ItemReferenceLoopHandling

Gets or sets the collection item reference loop handling.

Newtonsoft.Json.Serialization.JsonContainerContract.ItemTypeNameHandling

Gets or sets the collection item type name handling.

Newtonsoft.Json.Serialization.JsonArrayContract.#ctor(System.Type)

Initializes a new instance of the `Newtonsoft.Json.Serialization.JsonArrayContract` :param underlyingType: The underlying type for the contract.

`Newtonsoft.Json.Serialization.JsonArrayContract.CollectionItemType`

Gets the `System.Type` `System.Type`

`Newtonsoft.Json.Serialization.JsonArrayContract.IsMultidimensionalArray`

Gets a value indicating whether the collection type is a multidimensional array.

1.100 SerializationCallback

class `Newtonsoft.Json.Serialization.SerializationCallback`

Handles `Newtonsoft.Json.JsonSerializer` :param o: The object that raised the callback event.
:param context: The streaming context.

1.101 SerializationErrorCallback

class `Newtonsoft.Json.Serialization.SerializationErrorCallback`

Handles `Newtonsoft.Json.JsonSerializer` :param o: The object that raised the callback event.
:param context: The streaming context. :param errorContext: The error context.

1.102 ExtensionDataSetter

class `Newtonsoft.Json.Serialization.ExtensionDataSetter`

Sets extension data for an object during deserialization. :param o: The object to set extension data on. :param key: The extension data key. :param value: The extension data value.

1.103 ExtensionDataGetter

class `Newtonsoft.Json.Serialization.ExtensionDataGetter`

Gets extension data for an object during serialization. :param o: The object to set extension data on.

1.104 JsonDictionaryContract

class `Newtonsoft.Json.Serialization.JsonDictionaryContract`

Contract details for a `System.Type` `Newtonsoft.Json.JsonSerializer`

Newtonsoft.Json.Serialization.JsonDictionaryContract.#ctor(System.Type)

Initializes a new instance of the `Newtonsoft.Json.Serialization.JsonDictionaryContract`
:param underlyingType: The underlying type for the contract.

PropertyNameResolver

Gets or sets the property name resolver.

DictionaryKeyType

Gets the `System.Type` `System.Type`

DictionaryValueType

Gets the `System.Type` `System.Type`

1.105 JsonDynamicContract

class `Newtonsoft.Json.Serialization.JsonDynamicContract`

Contract details for a `System.Type` `Newtonsoft.Json.JsonSerializer`

Newtonsoft.Json.Serialization.JsonDynamicContract.#ctor(System.Type)

Initializes a new instance of the `Newtonsoft.Json.Serialization.JsonDynamicContract`
:param underlyingType: The underlying type for the contract.

Properties

Gets the object's properties.

PropertyNameResolver

Gets or sets the property name resolver.

1.106 JsonLinqContract

class `Newtonsoft.Json.Serialization.JsonLinqContract`

Contract details for a `System.Type` `Newtonsoft.Json.JsonSerializer`

Newtonsoft.Json.Serialization.JsonLinqContract.#ctor(System.Type)

Initializes a new instance of the `Newtonsoft.Json.Serialization.JsonLinqContract` :param
underlyingType: The underlying type for the contract.

1.107 JsonObjectContract

class `Newtonsoft.Json.Serialization.JsonObjectContract`

Contract details for a `System.Type` `Newtonsoft.Json.JsonSerializer`

Newtonsoft.Json.Serialization.JsonObjectContract.#ctor(System.Type)

Initializes a new instance of the `Newtonsoft.Json.Serialization.JsonObjectContract` :param
underlyingType: The underlying type for the contract.

MemberSerialization

Gets or sets the object member serialization.

ItemRequired

Gets or sets a value that indicates whether the object's properties are required.

Properties

Gets the object's properties.

ConstructorParameters

Gets the constructor parameters required for any non-default constructor

CreatorParameters

Gets a collection of `Newtonsoft.Json.Serialization.JsonProperty`
`Newtonsoft.Json.Serialization.JsonObjectContract.OverrideCreator`

OverrideConstructor

Gets or sets the override constructor used to create the object. This is set when a constructor is marked up using the `JsonConstructor` attribute.

ParametrizedConstructor

Gets or sets the parametrized constructor used to create the object.

OverrideCreator

Gets or sets the function used to create the object. When set this function will override `Newtonsoft.Json.Serialization.JsonContract.DefaultCreator`
`Newtonsoft.Json.Serialization.JsonObjectContract.CreatorParameters`

ExtensionDataSetter

Gets or sets the extension data setter.

ExtensionDataGetter

Gets or sets the extension data getter.

1.108 JsonPrimitiveContract

class `Newtonsoft.Json.Serialization.JsonPrimitiveContract`

Contract details for a `System.Type` `Newtonsoft.Json.JsonSerializer`

`Newtonsoft.Json.Serialization.JsonPrimitiveContract.#ctor(System.Type)`

Initializes a new instance of the `Newtonsoft.Json.Serialization.JsonPrimitiveContract`
:param underlyingType: The underlying type for the contract.

1.109 JsonProperty

class `Newtonsoft.Json.Serialization.JsonProperty`

Maps a JSON property to a .NET member or constructor parameter.

`ToString()`

Returns a `System.String` :return: A `System.String`

`PropertyName`

Gets or sets the name of the property.

`DeclaringType`

Gets or sets the type that declared this property.

`Order`

Gets or sets the order of serialization and deserialization of a member.

UnderlyingName

Gets or sets the name of the underlying member or parameter.

ValueProvider

Gets the `Newtonsoft.Json.Serialization.IValueProvider` `Newtonsoft.Json.Serialization.JsonProperty`
`Newtonsoft.Json.Serialization.IValueProvider` `Newtonsoft.Json.Serialization.JsonProperty`

AttributeProvider

Gets or sets the `Newtonsoft.Json.Serialization.IAttributeProvider`
`Newtonsoft.Json.Serialization.IAttributeProvider`

PropertyType

Gets or sets the type of the property.

Converter

Gets or sets the `Newtonsoft.Json.JsonConverter`

MemberConverter

Gets or sets the member converter.

Ignored

Gets or sets a value indicating whether this `Newtonsoft.Json.Serialization.JsonProperty`

Readable

Gets or sets a value indicating whether this `Newtonsoft.Json.Serialization.JsonProperty`

Writable

Gets or sets a value indicating whether this `Newtonsoft.Json.Serialization.JsonProperty`

HasMemberAttribute

Gets or sets a value indicating whether this `Newtonsoft.Json.Serialization.JsonProperty`

DefaultValue

Gets the default value.

Required

Gets or sets a value indicating whether this `Newtonsoft.Json.Serialization.JsonProperty`
`Newtonsoft.Json.Serialization.JsonProperty`

IsReference

Gets or sets a value indicating whether this property preserves object references.

NullValueHandling

Gets or sets the property null value handling.

DefaultValueHandling

Gets or sets the property default value handling.

ReferenceLoopHandling

Gets or sets the property reference loop handling.

ObjectCreationHandling

Gets or sets the property object creation handling.

TypeNameHandling

Gets or sets or sets the type name handling.

ShouldSerialize

Gets or sets a predicate used to determine whether the property should be serialize.

GetIsSpecified

Gets or sets a predicate used to determine whether the property should be serialized.

SetIsSpecified

Gets or sets an action used to set whether the property has been deserialized.

ItemConverter

Gets or sets the converter used when serializing the property's collection items.

ItemIsReference

Gets or sets whether this property's collection items are serialized as a reference.

ItemTypeNameHandling

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

ItemReferenceLoopHandling

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

1.110 JsonPropertyCollection

class `Newtonsoft.Json.Serialization.JsonPropertyCollection`

A collection of `Newtonsoft.Json.Serialization.JsonProperty`

Newtonsoft.Json.Serialization.JsonPropertyCollection.#ctor(System.Type)

Initializes a new instance of the `Newtonsoft.Json.Serialization.JsonPropertyCollection`
:param type: The type.

GetKeyForItem(Newtonsoft.Json.Serialization.JsonProperty)

When implemented in a derived class, extracts the key from the specified element. :param item: The element from which to extract the key. :return: The key for the specified element.

AddProperty(Newtonsoft.Json.Serialization.JsonProperty)

Adds a `Newtonsoft.Json.Serialization.JsonProperty` :param property: The property to add to the collection.

GetClosestMatchProperty(System.String)

Gets the closest matching `Newtonsoft.Json.Serialization.JsonProperty` :param propertyName: Name of the property. :return: A matching property if found.

GetProperty(System.String, System.StringComparison)

Gets a property by property name. :param propertyName: The name of the property to get. :param comparisonType: Type property name string comparison. :return: A matching property if found.

1.111 JsonStringContract

class `Newtonsoft.Json.Serialization.JsonStringContract`

Contract details for a `System.Type` `Newtonsoft.Json.JsonSerializer`

Newtonsoft.Json.Serialization.JsonStringContract.#ctor (`System.Type`)

Initializes a new instance of the `Newtonsoft.Json.Serialization.JsonStringContract` :param underlyingType: The underlying type for the contract.

Newtonsoft.Json.Serialization.JsonTypeReflector.CreateJsonConverterInstance (`System.Type`, `System.Object[]`)

Lookup and create an instance of the `JsonConverter` type described by the argument. :param converterType: The `JsonConverter` type to create. :param converterArgs: Optional arguments to pass to an initializing constructor of the `JsonConverter`.

If null, the default constructor is used.

Newtonsoft.Json.Serialization.JsonTypeReflector.GetJsonConverterCreator (`System.Type`)

Create a factory function that can be used to create instances of a `JsonConverter` described by the argument type. The returned function can then be used to either invoke the converter's default ctor, or any parameterized constructors by way of an object array.

1.112 MemoryTraceWriter

class `Newtonsoft.Json.Serialization.MemoryTraceWriter`

Represents a trace writer that writes to memory. When the trace message limit is reached then old trace messages will be removed as new messages are added.

Newtonsoft.Json.Serialization.MemoryTraceWriter.#ctor

Initializes a new instance of the `Newtonsoft.Json.Serialization.MemoryTraceWriter`

Trace (`Newtonsoft.Json.TraceLevel`, `System.String`, `System.Exception`)

Writes the specified trace level, message and optional exception. :param level: The

Newtonsoft.Json.TraceLevel :param message: The trace message.

param ex The trace exception. This parameter is optional.

GetTraceMessages ()

Returns an enumeration of the most recent trace messages. :return: An enumeration of the most recent trace messages.

ToString ()

Returns a `System.String` :return: A `System.String`

LevelFilter

Gets the `Newtonsoft.Json.TraceLevel` `Newtonsoft.Json.TraceLevel`

1.113 ObjectConstructor`1

Newtonsoft.Json.Serialization.ObjectConstructor`1

Represents a method that constructs an object.

1.114 OnErrorAttribute

class **Newtonsoft.Json.Serialization.OnErrorAttribute**

When applied to a method, specifies that the method is called when an error occurs serializing an object.

1.115 ReflectionAttributeProvider

class **Newtonsoft.Json.Serialization.ReflectionAttributeProvider**

Provides methods to get attributes from a `System.Type` `System.Reflection.MemberInfo` `System.Reflection.ParameterInfo` `System.Reflection.Assembly`

Newtonsoft.Json.Serialization.ReflectionAttributeProvider.#ctor(System.Object)

Initializes a new instance of the `Newtonsoft.Json.Serialization.ReflectionAttributeProvider`

GetAttributes (*System.Boolean*)

Returns a collection of all of the attributes, or an empty collection if there are no attributes. :param inherit: When true, look up the hierarchy chain for the inherited custom attribute. :return: A collection of `System.Attribute`

GetAttributes (*System.Type, System.Boolean*)

Returns a collection of attributes, identified by type, or an empty collection if there are no attributes. :param attributeType: The type of the attributes. :param inherit: When true, look up the hierarchy chain for the inherited custom attribute. :return: A collection of `System.Attribute`

1.116 ReflectionValueProvider

class **Newtonsoft.Json.Serialization.ReflectionValueProvider**

Get and set values for a `System.Reflection.MemberInfo`

Newtonsoft.Json.Serialization.ReflectionValueProvider.#ctor(System.Reflection.MemberInfo)

Initializes a new instance of the `Newtonsoft.Json.Serialization.ReflectionValueProvider`
:param memberInfo: The member info.

SetValue (*System.Object, System.Object*)

Sets the value. :param target: The target to set the value on. :param value: The value to set on the target.

GetValue (*System.Object*)

Gets the value. :param target: The target to get the value from. :return: The value.

1.117 StringEscapeHandling

class `Newtonsoft.Json.StringEscapeHandling`

Specifies how strings are escaped when writing JSON text.

Default

Only control characters (e.g. newline) are escaped.

EscapeNonAscii

All non-ASCII and control characters (e.g. newline) are escaped.

EscapeHtml

HTML (<, >, &, ', ") and control characters (e.g. newline) are escaped.

1.118 TraceLevel

class `Newtonsoft.Json.TraceLevel`

Specifies what messages to output for the `Newtonsoft.Json.Serialization.ITraceWriter`

Off

Output no tracing and debugging messages.

Error

Output error-handling messages.

Warning

Output warnings and error-handling messages.

Info

Output informational messages, warnings, and error-handling messages.

Verbose

Output all debugging and tracing messages.

1.119 TypeNameHandling

class `Newtonsoft.Json.TypeNameHandling`

Specifies type name handling options for the `Newtonsoft.Json.JsonSerializer`

None

Do not include the .NET type name when serializing types.

Objects

Include the .NET type name when serializing into a JSON object structure.

Arrays

Include the .NET type name when serializing into a JSON array structure.

All

Always include the .NET type name when serializing.

Auto

Include the .NET type name when the type of the object being serialized is not the same as its declared type.

Newtonsoft.Json.Utilities.CollectionUtils.IsNullOrEmpty`1(System.Collections.Generic.IEnumerable<T>)

Determines whether the collection is null or empty. :param collection: The collection. :return:

Newtonsoft.Json.Utilities.CollectionUtils.AddRange`1(System.Collections.Generic.IList<T>, IEnumerable<T>)

Adds the elements of the specified collection to the specified generic IList. :param initial: The list to add to. :param collection: The collection of elements to add.

Newtonsoft.Json.Utilities.CollectionUtils.IndexOf`1(System.Collections.Generic.IEnumerable<T>, T, IEqualityComparer<T>)

Returns the index of the first occurrence in a sequence by using a specified IEqualityComparer. :param list: A sequence in which to locate a value. :param value: The object to locate in the sequence :param comparer: An equality comparer to compare values. :return: The zero-based index of the first occurrence of value within the entire sequence, if found; otherwise, 1.

Newtonsoft.Json.Utilities.ConvertUtils.ConvertOrCast (*System.Object*, *System.Globalization.CultureInfo*, *System.Type*)

Converts the value to the specified type. If the value is unable to be converted, the value is checked whether it assignable to the specified type. :param initialValue: The value to convert.

Parameters

- **culture** – The culture to use when converting.
- **targetType** – The type to convert or cast the value to.

Returns The converted type. If conversion was unsuccessful, the initial value is returned if assignable to the target type.

Newtonsoft.Json.Utilities.DynamicProxyMetaObject`1.CallMethodWithResult(System.String, IEnumerable<System.Object>, System.Type, System.Reflection.MethodInfo, System.Collections.Generic.IDictionary<System.String, System.Object>)

Helper method for generating a MetaObject which calls a specific method on Dynamic that returns a result

Newtonsoft.Json.Utilities.DynamicProxyMetaObject`1.CallMethodReturnLast(System.String, IEnumerable<System.Object>, System.Type, System.Reflection.MethodInfo, System.Collections.Generic.IDictionary<System.String, System.Object>)

Helper method for generating a MetaObject which calls a specific method on Dynamic, but uses one of the arguments for the result.

Newtonsoft.Json.Utilities.DynamicProxyMetaObject`1.CallMethodNoResult(System.String, IEnumerable<System.Object>, System.Type, System.Reflection.MethodInfo, System.Collections.Generic.IDictionary<System.String, System.Object>)

Helper method for generating a MetaObject which calls a specific method on Dynamic, but uses one of the arguments for the result.

Newtonsoft.Json.Utilities.DynamicProxyMetaObject`1.GetRestrictions

Returns a Restrictions object which includes our current restrictions merged with a restriction limiting our type

Newtonsoft.Json.Utilities.EnumUtils.GetNamesAndValues`1

Gets a dictionary of the names and values of an Enum type.

Newtonsoft.Json.Utilities.EnumUtils.GetNamesAndValues`1(System.Type)

Gets a dictionary of the names and values of an Enum type. :param enumType: The enum type to get names and values for.

`Newtonsoft.Json.Utilities.ReflectionUtils.GetCollectionItemType (System.Type)`

Gets the type of the typed collection's items. :param type: The type. :return: The type of the typed collection's items.

`Newtonsoft.Json.Utilities.ReflectionUtils.GetMemberUnderlyingType (System.Reflection.MemberInfo)`

Gets the member's underlying type. :param member: The member. :return: The underlying type of the member.

`Newtonsoft.Json.Utilities.ReflectionUtils.IsIndexedProperty (System.Reflection.MemberInfo)`

Determines whether the member is an indexed property. :param member: The member. :return:

`Newtonsoft.Json.Utilities.ReflectionUtils.IsIndexedProperty (System.Reflection.PropertyInfo)`

Determines whether the property is an indexed property. :param property: The property. :return:

`Newtonsoft.Json.Utilities.ReflectionUtils.GetMemberValue (System.Reflection.MemberInfo, System.Object)`

Gets the member's value on the object. :param member: The member. :param target: The target object. :return: The member's value on the object.

`Newtonsoft.Json.Utilities.ReflectionUtils.SetMemberValue (System.Reflection.MemberInfo, System.Object, System.Object)`

Sets the member's value on the target object. :param member: The member. :param target: The target. :param value: The value.

`Newtonsoft.Json.Utilities.ReflectionUtils.CanReadMemberValue (System.Reflection.MemberInfo, System.Boolean)`

Determines whether the specified MemberInfo can be read. :param member: The MemberInfo to determine whether can be read. :param nonPublic: if set to :return:

`Newtonsoft.Json.Utilities.ReflectionUtils.CanSetMemberValue (System.Reflection.MemberInfo, System.Boolean, System.Boolean)`

Determines whether the specified MemberInfo can be set. :param member: The MemberInfo to determine whether can be set. :param nonPublic: if set to :param canSetReadOnly: if set to :return:

1.120 StringBuffer

class `Newtonsoft.Json.Utilities.StringBuffer`

Builds a string. Unlike StringBuilder this class lets you reuse it's internal buffer.

`Newtonsoft.Json.Utilities.StringUtils.IsWhiteSpace (System.String)`

Determines whether the string is all white space. Empty string will return false. :param s: The string to test whether it is all white space. :return:

`Newtonsoft.Json.Utilities.StringUtils.NullEmptyString (System.String)`

Nulls an empty string. :param s: The string. :return: Null if the string was null, otherwise the string unchanged.

1.121 WriteState

class `Newtonsoft.Json.WriteState`

Specifies the state of the `Newtonsoft.Json.JsonWriter`

Error

An exception has been thrown, which has left the `Newtonsoft.Json.JsonWriter`
`Newtonsoft.Json.JsonWriter.Close` `Newtonsoft.Json.JsonWriter`
`Newtonsoft.Json.JsonWriter` `System.InvalidOperationException`

Closed

The `Newtonsoft.Json.JsonWriter.Close`

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.

This is an autogenerated index file.

Please create a `/Users/eric/projects/sphinx-markdown-test/index.rst` or
`/Users/eric/projects/sphinx-markdown-test/README.rst` file with your own content.

Here is the `Newtonsoft.Json.Bson.BsonObjectId`

A

- Add() (Newtonsoft.Json.Linq.JObject method), 68
- Add() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JArray method), 64
- Add() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JContainer method), 62
- AddAfterSelf() (Newtonsoft.Json.Linq.JToken method), 51
- AddAnnotation() (Newtonsoft.Json.Linq.JToken method), 61
- AddBeforeSelf() (Newtonsoft.Json.Linq.JToken method), 51
- AddFirst() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JContainer method), 62
- AdditionalItems (Newtonsoft.Json.Schema.JsonSchema attribute), 83
- AdditionalProperties (Newtonsoft.Json.Schema.JsonSchema attribute), 83
- AddProperty() (Newtonsoft.Json.Serialization.JsonPropertyCollection method), 99
- AddReference() (Newtonsoft.Json.Serialization.IReferenceResolver method), 91
- AfterSelf() (Newtonsoft.Json.Linq.JToken method), 52
- All (Newtonsoft.Json.PreserveReferencesHandling attribute), 80
- All (Newtonsoft.Json.TypeNameHandling attribute), 102
- AllowAdditionalItems (Newtonsoft.Json.Schema.JsonSchema attribute), 83
- AllowAdditionalProperties (Newtonsoft.Json.Schema.JsonSchema attribute), 84
- AllowIntegerValues (Newtonsoft.Json.Converters.StringEnumConverter attribute), 20
- AllowNonPublicDefaultConstructor (Newtonsoft.Json.ConstructorHandling attribute), 14
- AllowNull (Newtonsoft.Json.Required attribute), 80
- AllowNullItems (Newtonsoft.Json.JsonContainerAttribute.Newtonsoft.Json.JsonArrayAttribute), 25
- Always (Newtonsoft.Json.Required attribute), 80
- Ancestors() (Newtonsoft.Json.Linq.JToken method), 51
- AncestorsAndSelf() (Newtonsoft.Json.Linq.JToken method), 51
- Annotation() (Newtonsoft.Json.Linq.JToken method), 61
- Annotations() (Newtonsoft.Json.Linq.JToken method), 61
- Any (Newtonsoft.Json.Schema.JsonSchemaType attribute), 86
- Array (Newtonsoft.Json.JsonReader.State attribute), 4
- Array (Newtonsoft.Json.Linq.JTokenType attribute), 74
- Array (Newtonsoft.Json.Schema.JsonSchemaType attribute), 86
- Array (Newtonsoft.Json.WriteState attribute), 105
- Arrays (Newtonsoft.Json.PreserveReferencesHandling attribute), 80
- Arrays (Newtonsoft.Json.TypeNameHandling attribute), 102
- ArrayStart (Newtonsoft.Json.JsonReader.State attribute), 3
- AsJEnumerable() (Newtonsoft.Json.Linq.Extensions method), 50
- AttributeProvider (Newtonsoft.Json.Serialization.JsonProperty attribute), 98
- Auto (Newtonsoft.Json.ObjectCreationHandling attribute), 79
- Auto (Newtonsoft.Json.TypeNameHandling attribute), 102

B

- BeforeSelf() (Newtonsoft.Json.Linq.JToken method), 52
- Binder (Newtonsoft.Json.JsonSerializer attribute), 38
- Binder (Newtonsoft.Json.JsonSerializerSettings attribute), 41
- BindToName() (Newtonsoft.Json.Serialization.DefaultSerializationBinder attribute), 107

method), 91

BindToName() (Newtonsoft.Json.SerializationBinder method), 87

BindToType() (Newtonsoft.Json.Serialization.DefaultSerializationBinder method), 91

BindToType() (Newtonsoft.Json.SerializationBinder method), 87

Boolean (Newtonsoft.Json.JsonToken attribute), 47

Boolean (Newtonsoft.Json.Linq.JTokenType attribute), 74

Boolean (Newtonsoft.Json.Schema.JsonSchemaType attribute), 86

Bytes (Newtonsoft.Json.JsonToken attribute), 47

Bytes (Newtonsoft.Json.Linq.JTokenType attribute), 75

C

CamelCaseText (Newtonsoft.Json.Converters.StringEnumConverter attribute), 20

CanConvert() (Newtonsoft.Json.Converters.DateTimeConverterBase method), 16

CanConvert() (Newtonsoft.Json.Converters.DiscriminatedUnionConverter method), 17

CanConvert() (Newtonsoft.Json.Converters.ExpandoObjectConverter method), 17

CanConvert() (Newtonsoft.Json.Converters.KeyValuePairConverter method), 19

CanConvert() (Newtonsoft.Json.Converters.RegexConverter method), 19

CanConvert() (Newtonsoft.Json.Converters.StringEnumConverter method), 20

CanConvert() (Newtonsoft.Json.Converters.VersionConverter method), 20

CanConvert() (Newtonsoft.Json.Converters.XmlNodeConverter method), 21

CanConvert() (Newtonsoft.Json.JsonConverter method), 15

CanConvert() (Newtonsoft.Json.JsonConverter Newtonsoft.Json.Converters.BsonObjectConverter method), 15

CanRead (Newtonsoft.Json.JsonConverter attribute), 15

CanReadMemberValue() (Newtonsoft.Json.TypeNameHandling.Newtonsoft.Json.Utilities.ReflectionUtilities method), 104

CanSetMemberValue() (Newtonsoft.Json.TypeNameHandling.Newtonsoft.Json.Utilities.ReflectionUtilities method), 104

CanWrite (Newtonsoft.Json.Converters.ExpandoObjectConverter attribute), 17

CanWrite (Newtonsoft.Json.JsonConverter attribute), 15

CheckAdditionalContent (Newtonsoft.Json.JsonSerializer attribute), 40

CheckAdditionalContent (Newtonsoft.Json.JsonSerializerSettings attribute), 42

Children() (Newtonsoft.Json.Linq.JToken method), 52

Children() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JContainer method), 62

ChildrenTokens (Newtonsoft.Json.Linq.JConstructor attribute), 66

ChildrenTokens (Newtonsoft.Json.Linq.JObject attribute), 69

ChildrenTokens (Newtonsoft.Json.Linq.JProperty attribute), 70

ChildrenTokens (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JArray attribute), 65

ChildrenTokens (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JContainer attribute), 63

Clear() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JArray method), 65

Close() (Newtonsoft.Json.JsonReader method), 2

Close() (Newtonsoft.Json.JsonReader.State.Newtonsoft.Json.Bson.BsonReader method), 5

Close() (Newtonsoft.Json.JsonTextReader method), 42

Close() (Newtonsoft.Json.JsonTextWriter method), 43

Close() (Newtonsoft.Json.JsonWriter method), 5

Close() (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter method), 12

Close() (Newtonsoft.Json.Linq.JTokenWriter method), 75

Closed (Newtonsoft.Json.JsonReader.State attribute), 4

Closed (Newtonsoft.Json.WriteState attribute), 105

CloseInput (Newtonsoft.Json.JsonReader attribute), 2

CloseOutput (Newtonsoft.Json.JsonWriter attribute), 10

CollectionItemType (Newtonsoft.Json.Serialization.JsonContract.Newtonsoft.Json.Serialization attribute), 95

Comment (Newtonsoft.Json.JsonToken attribute), 46

Comment (Newtonsoft.Json.Linq.JTokenType attribute), 74

CompareTo() (Newtonsoft.Json.Linq.JValue method), 72

Complete (Newtonsoft.Json.JsonReader.State attribute), 3

Concat (Newtonsoft.Json.Linq.MergeArrayHandling attribute), 78

Constructor (Newtonsoft.Json.JsonReader.State attribute), 4

Constructor (Newtonsoft.Json.Linq.JTokenType attribute), 4

- attribute), 18
- Culture (Newtonsoft.Json.JsonReader attribute), 3
- Culture (Newtonsoft.Json.JsonSerializer attribute), 40
- Culture (Newtonsoft.Json.JsonSerializerSettings attribute), 42
- Culture (Newtonsoft.Json.JsonWriter attribute), 11
- CurrentObject (Newtonsoft.Json.Serialization.ErrorEventArgs attribute), 92
- CurrentState (Newtonsoft.Json.JsonReader attribute), 2
- CurrentToken (Newtonsoft.Json.Linq.JTokenReader attribute), 74
- CurrentToken (Newtonsoft.Json.Linq.JTokenWriter attribute), 78
- D**
- Date (Newtonsoft.Json.JsonToken attribute), 47
- Date (Newtonsoft.Json.Linq.JTokenType attribute), 75
- DateFormatHandling (Newtonsoft.Json.JsonSerializer attribute), 39
- DateFormatHandling (Newtonsoft.Json.JsonSerializerSettings attribute), 41
- DateFormatHandling (Newtonsoft.Json.JsonWriter attribute), 11
- DateFormatString (Newtonsoft.Json.JsonReader attribute), 3
- DateFormatString (Newtonsoft.Json.JsonSerializer attribute), 40
- DateFormatString (Newtonsoft.Json.JsonSerializerSettings attribute), 41
- DateFormatString (Newtonsoft.Json.JsonWriter attribute), 11
- DateParseHandling (Newtonsoft.Json.JsonReader attribute), 2
- DateParseHandling (Newtonsoft.Json.JsonSerializer attribute), 39
- DateParseHandling (Newtonsoft.Json.JsonSerializerSettings attribute), 41
- DateTime (Newtonsoft.Json.DateParseHandling attribute), 22
- DateTimeFormat (Newtonsoft.Json.Converters.IsoDateTimeConverter attribute), 18
- DateTimeKindHandling (Newtonsoft.Json.JsonReader.State.Newtonsoft.Json.Bson.BsonReader attribute), 5
- DateTimeKindHandling (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter attribute), 14
- DateTimeOffset (Newtonsoft.Json.DateParseHandling attribute), 22
- DateTimeStyles (Newtonsoft.Json.Converters.IsoDateTimeConverter attribute), 18
- DateTimeZoneHandling (Newtonsoft.Json.JsonReader attribute), 2
- DateTimeZoneHandling (Newtonsoft.Json.JsonSerializer attribute), 39
- DateTimeZoneHandling (Newtonsoft.Json.JsonSerializerSettings attribute), 41
- DateTimeZoneHandling (Newtonsoft.Json.JsonWriter attribute), 11
- Decimal (Newtonsoft.Json.FloatParseHandling attribute), 23
- DeclaringType (Newtonsoft.Json.Serialization.JsonProperty attribute), 97
- DeepClone() (Newtonsoft.Json.Linq.JToken method), 60
- DeepEquals() (Newtonsoft.Json.Linq.JToken method), 51
- Default (Newtonsoft.Json.ConstructorHandling attribute), 14
- Default (Newtonsoft.Json.MetadataPropertyHandling attribute), 79
- Default (Newtonsoft.Json.Required attribute), 80
- Default (Newtonsoft.Json.Schema.JsonSchema attribute), 84
- Default (Newtonsoft.Json.StringEscapeHandling attribute), 102
- DefaultCreator (Newtonsoft.Json.Serialization.JsonContract attribute), 94
- DefaultCreatorNonPublic (Newtonsoft.Json.Serialization.JsonContract attribute), 94
- DefaultSettings (Newtonsoft.Json.JsonConvert attribute), 32
- DefaultValue (Newtonsoft.Json.FloatFormatHandling attribute), 23
- DefaultValue (Newtonsoft.Json.Serialization.JsonProperty attribute), 98
- DefaultValueHandling (Newtonsoft.Json.JsonPropertyAttribute attribute), 35
- DefaultValueHandling (Newtonsoft.Json.JsonSerializer attribute), 39
- DefaultValueHandling (Newtonsoft.Json.JsonSerializerSettings attribute), 40
- DefaultValueHandling (Newtonsoft.Json.Serialization.JsonProperty attribute), 98
- Depth (Newtonsoft.Json.JsonReader attribute), 3
- Depth (Newtonsoft.Json.JsonValidatingReader attribute),

- 48
- Descendants() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JContainer attribute), 62
- DescendantsAndSelf() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JContainer attribute), 62
- Description (Newtonsoft.Json.JsonContainerAttribute attribute), 25
- Description (Newtonsoft.Json.Schema.JsonSchema attribute), 82
- Deserialize() (Newtonsoft.Json.JsonSerializer method), 37, 38
- DeserializeObject() (Newtonsoft.Json.JsonConvert method), 29, 30
- DeserializeObjectAsync() (Newtonsoft.Json.JsonConvert method), 31
- DeserializeRootElementName (Newtonsoft.Json.Converters.XmlNodeConverter attribute), 21
- DeserializeXmlNode() (Newtonsoft.Json.JsonConvert method), 32
- DictionaryKeyType (Newtonsoft.Json.Serialization.JsonDictionaryContract attribute), 95
- DictionaryValueType (Newtonsoft.Json.Serialization.JsonDictionaryContract attribute), 96
- Disallow (Newtonsoft.Json.Schema.JsonSchema attribute), 84
- Dispose() (Newtonsoft.Json.JsonReader method), 2
- DivisibleBy (Newtonsoft.Json.Schema.JsonSchema attribute), 83
- Double (Newtonsoft.Json.FloatParseHandling attribute), 23
- DynamicCodeGeneration (Newtonsoft.Json.Serialization.IContractResolver.Newtonsoft.Json.Serialization.DefaultContractResolver attribute), 90
- ## E
- EndArray (Newtonsoft.Json.JsonToken attribute), 47
- EndConstructor (Newtonsoft.Json.JsonToken attribute), 47
- EndObject (Newtonsoft.Json.JsonToken attribute), 47
- Enum (Newtonsoft.Json.Schema.JsonSchema attribute), 84
- EqualityComparer (Newtonsoft.Json.Linq.JToken attribute), 61
- Equals() (Newtonsoft.Json.Linq.JTokenEqualityComparer method), 73
- Equals() (Newtonsoft.Json.Linq.JValue method), 72
- Error (Newtonsoft.Json.JsonReader.State attribute), 4
- Error (Newtonsoft.Json.JsonSerializerSettings attribute), 41
- Error (Newtonsoft.Json.MissingMemberHandling attribute), 79
- Error (Newtonsoft.Json.ReferenceLoopHandling attribute), 80
- Error (Newtonsoft.Json.Serialization.ErrorContext attribute), 92
- Error (Newtonsoft.Json.TraceLevel attribute), 102
- Error (Newtonsoft.Json.WriteState attribute), 105
- ErrorContext (Newtonsoft.Json.Serialization.ErrorEventArgs attribute), 92
- EscapeHtml (Newtonsoft.Json.StringEscapeHandling attribute), 102
- EscapeNonAscii (Newtonsoft.Json.StringEscapeHandling attribute), 102
- Exception (Newtonsoft.Json.Schema.ValidationEventArgs attribute), 87
- ExclusiveMaximum (Newtonsoft.Json.Schema.JsonSchema attribute), 83
- ExclusiveMinimum (Newtonsoft.Json.Schema.JsonSchema attribute), 83
- Extends (Newtonsoft.Json.Schema.JsonSchema attribute), 84
- ExtensionDataGetter (Newtonsoft.Json.Serialization.JsonObjectContract attribute), 97
- ExtensionDataSetter (Newtonsoft.Json.Serialization.JsonObjectContract attribute), 97
- ## F
- False (Newtonsoft.Json.JsonConvert attribute), 26
- Fields (Newtonsoft.Json.MemberSerialization attribute), 84
- Finished (Newtonsoft.Json.JsonReader.State attribute), 4
- First (Newtonsoft.Json.Linq.JToken attribute), 62
- First (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JContainer attribute), 63
- Float (Newtonsoft.Json.JsonToken attribute), 46
- Float (Newtonsoft.Json.Linq.JTokenType attribute), 74
- Float (Newtonsoft.Json.Schema.JsonSchemaType attribute), 86
- FloatFormatHandling (Newtonsoft.Json.JsonSerializer attribute), 39
- FloatFormatHandling (Newtonsoft.Json.JsonSerializerSettings attribute), 41
- FloatFormatHandling (Newtonsoft.Json.JsonWriter attribute), 11
- FloatParseHandling (Newtonsoft.Json.JsonReader attribute), 3

FloatParseHandling (Newtonsoft.Json.JsonSerializer attribute), 39

FloatParseHandling (Newtonsoft.Json.JsonSerializerSettings attribute), 41

Flush() (Newtonsoft.Json.JsonTextWriter method), 43

Flush() (Newtonsoft.Json.JsonWriter method), 5

Flush() (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter method), 11

Flush() (Newtonsoft.Json.Linq.JTokenWriter method), 75

Format (Newtonsoft.Json.Schema.JsonSchema attribute), 84

Formatting (Newtonsoft.Json.JsonSerializer attribute), 39

Formatting (Newtonsoft.Json.JsonSerializerSettings attribute), 41

Formatting (Newtonsoft.Json.JsonWriter attribute), 10

FromObject() (Newtonsoft.Json.Linq.JObject method), 68

FromObject() (Newtonsoft.Json.Linq.JToken method), 59

FromObject() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JArray method), 64

Full (System.Runtime.Serialization.Formatters.FormatterAssemblyStyle attribute), 24

G

Generate() (Newtonsoft.Json.Schema.JsonSchemaGenerator method), 85

GetAttributes() (Newtonsoft.Json.Serialization.IAttributeProvider method), 92, 93

GetAttributes() (Newtonsoft.Json.Serialization.ReflectionAttributeProvider method), 101

GetClosestMatchProperty() (Newtonsoft.Json.Serialization.JsonPropertyCollection method), 99

GetCollectionItemType() (Newtonsoft.Json.TypeNameHandling.Newtonsoft.Json.Utilities.ReflectionUtilities method), 103

GetEnumerator() (Newtonsoft.Json.Linq.JObject method), 69

GetEnumerator() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JArray method), 64

GetHashCode() (Newtonsoft.Json.Linq.JTokenEqualityComparer method), 73

GetHashCode() (Newtonsoft.Json.Linq.JValue method), 72

GetIsSpecified (Newtonsoft.Json.Serialization.JsonProperty attribute), 99

GetJsonConverterCreator() (Newtonsoft.Json.Serialization.JsonStringContract.Newtonsoft.Json.Serialization.JsonPropertyCollection method), 100

GetKeyForItem() (Newtonsoft.Json.Serialization.JsonPropertyCollection method), 99

GetMemberUnderlyingType() (Newtonsoft.Json.TypeNameHandling.Newtonsoft.Json.Utilities.ReflectionUtilities method), 104

GetMemberValue() (Newtonsoft.Json.TypeNameHandling.Newtonsoft.Json.Utilities.ReflectionUtilities method), 104

GetMetaObject() (Newtonsoft.Json.Linq.JObject method), 69

GetMetaObject() (Newtonsoft.Json.Linq.JToken method), 60

GetMetaObject() (Newtonsoft.Json.Linq.JValue method), 72

GetProperty() (Newtonsoft.Json.Serialization.JsonPropertyCollection method), 99

GetReference() (Newtonsoft.Json.Serialization.IReferenceResolver method), 91

GetResolvedPropertyName() (Newtonsoft.Json.Serialization.IContractResolver.Newtonsoft.Json.Serialization.JsonPropertyCollection method), 90

GetSchema() (Newtonsoft.Json.JsonConverter method), 15

GetSchema() (Newtonsoft.Json.Schema.JsonSchemaResolver method), 85

GetSerializableMembers() (Newtonsoft.Json.Serialization.IContractResolver.Newtonsoft.Json.Serialization.JsonPropertyCollection method), 88

GetTraceMessages() (Newtonsoft.Json.Serialization.MemoryTraceWriter method), 100

GetValue() (Newtonsoft.Json.Linq.JObject method), 68

GetValue() (Newtonsoft.Json.Serialization.IValueProvider method), 90

GetValue() (Newtonsoft.Json.Serialization.IValueProvider.Newtonsoft.Json.Serialization.JsonPropertyCollection method), 91

GetValue() (Newtonsoft.Json.Serialization.ReflectionValueProvider method), 101

Guid (Newtonsoft.Json.Linq.JTokenType attribute), 75

H

Handled (Newtonsoft.Json.Serialization.ErrorContext attribute), 92

HasLineInfo() (Newtonsoft.Json.IJsonLineInfo method), 24

HasLineInfo() (Newtonsoft.Json.JsonTextReader method), 42

HasMemberAttribute (Newtonsoft.Json.Serialization.JsonProperty attribute), 98	method), 21
HasValues (Newtonsoft.Json.Linq.JToken attribute), 61	DateFormat (Newtonsoft.Json.DateFormatHandling attribute), 21
HasValues (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JConstructor attribute), 63	IsReadOnly (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JArray attribute), 65
HasValues (Newtonsoft.Json.Linq.JValue attribute), 72	IsReference (Newtonsoft.Json.JsonContainerAttribute attribute), 25
Hidden (Newtonsoft.Json.Schema.JsonSchema attribute), 82	IsReference (Newtonsoft.Json.JsonPropertyAttribute attribute), 35
I	IsReference (Newtonsoft.Json.Serialization.JsonContract attribute), 93
Id (Newtonsoft.Json.JsonContainerAttribute attribute), 25	IsReference (Newtonsoft.Json.Serialization.JsonProperty attribute), 98
Id (Newtonsoft.Json.Schema.JsonSchema attribute), 82	IsReferenced() (Newtonsoft.Json.Serialization.IReferenceResolver method), 91
Ignore (Newtonsoft.Json.DefaultValueHandling attribute), 23	IsValid() (Newtonsoft.Json.Schema.Extensions method), 81
Ignore (Newtonsoft.Json.MetadataPropertyHandling attribute), 79	IsWhiteSpace() (Newtonsoft.Json.Utilities.StringBuffer.Newtonsoft.Json.Utilities.StringU method), 104
Ignore (Newtonsoft.Json.MissingMemberHandling attribute), 79	Item (Newtonsoft.Json.Linq.JConstructor attribute), 66
Ignore (Newtonsoft.Json.NullValueHandling attribute), 79	Item (Newtonsoft.Json.Linq.JObject attribute), 69
Ignore (Newtonsoft.Json.ReferenceLoopHandling attribute), 80	Item (Newtonsoft.Json.Linq.JToken attribute), 62
IgnoreAndPopulate (Newtonsoft.Json.DefaultValueHandling attribute), 23	Item (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JArray attribute), 65
Ignored (Newtonsoft.Json.Serialization.JsonProperty attribute), 98	ItemConverter (Newtonsoft.Json.Serialization.JsonContract.Newtonsoft.Json.Serialization attribute), 94
Include (Newtonsoft.Json.DefaultValueHandling attribute), 22	ItemConverter (Newtonsoft.Json.Serialization.JsonProperty attribute), 99
Include (Newtonsoft.Json.NullValueHandling attribute), 79	ItemConverterParameters (Newtonsoft.Json.JsonContainerAttribute attribute), 25
Indentation (Newtonsoft.Json.JsonTextWriter attribute), 46	ItemConverterParameters (Newtonsoft.Json.JsonPropertyAttribute attribute), 35
IndentChar (Newtonsoft.Json.JsonTextWriter attribute), 46	ItemConverterType (Newtonsoft.Json.JsonContainerAttribute attribute), 25
Indented (Newtonsoft.Json.Formatting attribute), 24	ItemConverterType (Newtonsoft.Json.JsonPropertyAttribute attribute), 35
IndexOf() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JArray method), 64	ItemIsReference (Newtonsoft.Json.JsonContainerAttribute attribute), 25
Info (Newtonsoft.Json.TraceLevel attribute), 102	ItemIsReference (Newtonsoft.Json.JsonPropertyAttribute attribute), 35
Insert() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JArray method), 64	ItemIsReference (Newtonsoft.Json.JsonContainerAttribute attribute), 25
Integer (Newtonsoft.Json.JsonToken attribute), 46	ItemIsReference (Newtonsoft.Json.JsonPropertyAttribute attribute), 35
Integer (Newtonsoft.Json.Linq.JTokenType attribute), 74	ItemIsReference (Newtonsoft.Json.JsonContainerAttribute attribute), 25
Integer (Newtonsoft.Json.Schema.JsonSchemaType attribute), 86	ItemIsReference (Newtonsoft.Json.JsonPropertyAttribute attribute), 35
IsIndexedProperty() (Newtonsoft.Json.TypeNameHandling.Newtonsoft.Json.Utilities method), 104	ItemIsReference (Newtonsoft.Json.JsonContainerAttribute attribute), 25
IsMultidimensionalArray (Newtonsoft.Json.Serialization.JsonContract.Newtonsoft.Json.Serialization attribute), 95	ItemIsReference (Newtonsoft.Json.JsonPropertyAttribute attribute), 35
IsNamespaceAttribute() (Newtonsoft.Json.Converters.XmlNodeConverter	ItemIsReference (Newtonsoft.Json.JsonContainerAttribute attribute), 25

99		LinePosition (Newtonsoft.Json.IJsonLineInfo attribute), 24
ItemReferenceLoopHandling (Newtonsoft.Json.JsonContainerAttribute attribute), 25		LinePosition (Newtonsoft.Json.JsonReaderException attribute), 36
ItemReferenceLoopHandling (Newtonsoft.Json.JsonPropertyAttribute attribute), 35		LinePosition (Newtonsoft.Json.JsonTextReader attribute), 43
ItemReferenceLoopHandling (Newtonsoft.Json.Serialization.JsonContract attribute), 94		LinePosition (Newtonsoft.Json.Schema.JsonSchemaException attribute), 84
ItemReferenceLoopHandling (Newtonsoft.Json.Serialization.JsonProperty attribute), 99		Load() (Newtonsoft.Json.Linq.JObject method), 68
ItemRequired (Newtonsoft.Json.JsonObjectAttribute attribute), 34		Load() (Newtonsoft.Json.Linq.JProperty method), 70
ItemRequired (Newtonsoft.Json.Serialization.JsonObjectContract attribute), 96		Load() (Newtonsoft.Json.Linq.JToken method), 60
Items (Newtonsoft.Json.Schema.JsonSchema attribute), 83		Load() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JArray method), 63
ItemTypeNameHandling (Newtonsoft.Json.JsonContainerAttribute attribute), 25		LoadedSchemas (Newtonsoft.Json.Schema.JsonSchemaResolver attribute), 85
ItemTypeNameHandling (Newtonsoft.Json.JsonPropertyAttribute attribute), 35		Local (Newtonsoft.Json.DateTimeZoneHandling attribute), 22
ItemTypeNameHandling (Newtonsoft.Json.Serialization.JsonContract attribute), 94		
ItemTypeNameHandling (Newtonsoft.Json.Serialization.JsonProperty attribute), 99		
J		M
JsonNet35BinaryCompatibility (Newtonsoft.Json.JsonReader.State.Newtonsoft.Json.Bson.BsonReader attribute), 5		MaxDepth (Newtonsoft.Json.JsonReader attribute), 3
L		MaxDepth (Newtonsoft.Json.JsonSerializer attribute), 40
Last (Newtonsoft.Json.Linq.JToken attribute), 62		MaxDepth (Newtonsoft.Json.JsonSerializerSettings attribute), 41
Last (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JContainer attribute), 63		Maximum (Newtonsoft.Json.Schema.JsonSchema attribute), 83
LevelFilter (Newtonsoft.Json.Serialization.ITraceWriter attribute), 93		MaximumLength (Newtonsoft.Json.Schema.JsonSchema attribute), 83
LevelFilter (Newtonsoft.Json.Serialization.MemoryTraceWriter attribute), 100		Member (Newtonsoft.Json.Serialization.ErrorContext attribute), 92
LineNumber (Newtonsoft.Json.IJsonLineInfo attribute), 24		MemberConverter (Newtonsoft.Json.Serialization.JsonProperty attribute), 92
LineNumber (Newtonsoft.Json.JsonReaderException attribute), 36		MemberSerialization (Newtonsoft.Json.JsonObjectAttribute attribute), 34
LineNumber (Newtonsoft.Json.JsonTextReader attribute), 42		MemberSerialization (Newtonsoft.Json.Serialization.JsonObjectContract attribute), 96
LineNumber (Newtonsoft.Json.Schema.JsonSchemaException attribute), 84		Merge (Newtonsoft.Json.Linq.MergeArrayHandling attribute), 78
		Merge() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JContainer method), 63
		MergeArrayHandling (Newtonsoft.Json.Linq.JsonMergeSettings attribute), 73
		Message (Newtonsoft.Json.Schema.ValidationEventArgs attribute), 87
		MetadataPropertyHandling (Newtonsoft.Json.JsonSerializer attribute), 39

- [MetadataPropertyHandling](#) (Newtonsoft.Json.JsonSerializerSettings attribute), [41](#)
[MicrosoftDateFormat](#) (Newtonsoft.Json.DateFormatHandling attribute), [22](#)
[Minimum](#) (Newtonsoft.Json.Schema.JsonSchema attribute), [83](#)
[MinimumItems](#) (Newtonsoft.Json.Schema.JsonSchema attribute), [83](#)
[MinimumLength](#) (Newtonsoft.Json.Schema.JsonSchema attribute), [82](#)
[MissingMemberHandling](#) (Newtonsoft.Json.JsonSerializer attribute), [39](#)
[MissingMemberHandling](#) (Newtonsoft.Json.JsonSerializerSettings attribute), [40](#)
- ## N
- [Name](#) (Newtonsoft.Json.Linq.JConstructor attribute), [66](#)
[Name](#) (Newtonsoft.Json.Linq.JProperty attribute), [70](#)
[NaN](#) (Newtonsoft.Json.JsonConvert attribute), [26](#)
[NegativeInfinity](#) (Newtonsoft.Json.JsonConvert attribute), [26](#)
[Newtonsoft.Json.Bson.BsonObjectId](#) (built-in class), [1](#)
[Newtonsoft.Json.Bson.BsonReader](#) (built-in class), [1](#)
[Newtonsoft.Json.Bson.BsonWriter](#) (built-in class), [5](#)
[Newtonsoft.Json.ConstructorHandling](#) (built-in class), [14](#)
[Newtonsoft.Json.Converters.BsonObjectIdConverter](#) (built-in class), [14](#)
[Newtonsoft.Json.Converters.DateTimeConverterBase](#) (built-in class), [16](#)
[Newtonsoft.Json.Converters.DiscriminatedUnionConverter](#) (built-in class), [16](#)
[Newtonsoft.Json.Converters.ExpandoObjectConverter](#) (built-in class), [17](#)
[Newtonsoft.Json.Converters.IsoDateTimeConverter](#) (built-in class), [17](#)
[Newtonsoft.Json.Converters.JavaScriptDateTimeConverter](#) (built-in class), [18](#)
[Newtonsoft.Json.Converters.KeyValuePairConverter](#) (built-in class), [18](#)
[Newtonsoft.Json.Converters.RegexConverter](#) (built-in class), [19](#)
[Newtonsoft.Json.Converters.StringEnumConverter](#) (built-in class), [19](#)
[Newtonsoft.Json.Converters.VersionConverter](#) (built-in class), [20](#)
[Newtonsoft.Json.Converters.XmlNodeConverter](#) (built-in class), [21](#)
[Newtonsoft.Json.DateFormatHandling](#) (built-in class), [21](#)
[Newtonsoft.Json.DateParseHandling](#) (built-in class), [22](#)
[Newtonsoft.Json.DateTimeZoneHandling](#) (built-in class), [22](#)
[Newtonsoft.Json.DefaultValueHandling](#) (built-in class), [22](#)
[Newtonsoft.Json.FloatFormatHandling](#) (built-in class), [23](#)
[Newtonsoft.Json.FloatParseHandling](#) (built-in class), [23](#)
[Newtonsoft.Json.Formatting](#) (built-in class), [24](#)
[Newtonsoft.Json.IJsonLineInfo](#) (built-in class), [24](#)
[Newtonsoft.Json.JsonArrayAttribute](#) (built-in class), [24](#)
[Newtonsoft.Json.JsonConstructorAttribute](#) (built-in class), [25](#)
[Newtonsoft.Json.JsonContainerAttribute](#) (built-in class), [24](#)
[Newtonsoft.Json.JsonConvert](#) (built-in class), [26](#)
[Newtonsoft.Json.JsonConverter](#) (built-in class), [14](#)
[Newtonsoft.Json.JsonConverterAttribute](#) (built-in class), [32](#)
[Newtonsoft.Json.JsonConverterCollection](#) (built-in class), [33](#)
[Newtonsoft.Json.JsonDictionaryAttribute](#) (built-in class), [33](#)
[Newtonsoft.Json.JsonException](#) (built-in class), [33](#)
[Newtonsoft.Json.JsonExtensionDataAttribute](#) (built-in class), [34](#)
[Newtonsoft.Json.JsonIgnoreAttribute](#) (built-in class), [34](#)
[Newtonsoft.Json.JsonObjectAttribute](#) (built-in class), [34](#)
[Newtonsoft.Json.JsonPropertyAttribute](#) (built-in class), [35](#)
[Newtonsoft.Json.JsonReader](#) (built-in class), [1](#)
[Newtonsoft.Json.JsonReader.State](#) (built-in class), [3](#)
[Newtonsoft.Json.JsonReaderException](#) (built-in class), [36](#)
[Newtonsoft.Json.JsonSerializationException](#) (built-in class), [36](#)
[Newtonsoft.Json.JsonSerializer](#) (built-in class), [37](#)
[Newtonsoft.Json.JsonSerializerSettings](#) (built-in class), [40](#)
[Newtonsoft.Json.JsonTextReader](#) (built-in class), [42](#)
[Newtonsoft.Json.JsonTextWriter](#) (built-in class), [43](#)
[Newtonsoft.Json.JsonToken](#) (built-in class), [46](#)
[Newtonsoft.Json.JsonValidatingReader](#) (built-in class), [47](#)
[Newtonsoft.Json.JsonWriter](#) (built-in class), [5](#)
[Newtonsoft.Json.JsonWriterException](#) (built-in class), [48](#)
[Newtonsoft.Json.Linq.Extensions](#) (built-in class), [49](#)
[Newtonsoft.Json.Linq.JArray](#) (built-in class), [51](#)
[Newtonsoft.Json.Linq.JConstructor](#) (built-in class), [65](#)
[Newtonsoft.Json.Linq.JContainer](#) (built-in class), [51](#)
[Newtonsoft.Json.Linq.JObject](#) (built-in class), [67](#)
[Newtonsoft.Json.Linq.JProperty](#) (built-in class), [69](#)
[Newtonsoft.Json.Linq.JRaw](#) (built-in class), [70](#)
[Newtonsoft.Json.Linq.JsonMergeSettings](#) (built-in class), [73](#)
[Newtonsoft.Json.Linq.JToken](#) (built-in class), [51](#)
[Newtonsoft.Json.Linq.JTokenEqualityComparer](#) (built-in class), [73](#)
[Newtonsoft.Json.Linq.JTokenReader](#) (built-in class), [73](#)

Newtonsoft.Json.Linq.JTokenType (built-in class), 74	Newtonsoft.Json.Serialization.IReferenceResolver (built-in class), 91
Newtonsoft.Json.Linq.JTokenWriter (built-in class), 75	Newtonsoft.Json.Serialization.ITraceWriter (built-in class), 93
Newtonsoft.Json.Linq.JValue (built-in class), 70	Newtonsoft.Json.Serialization.IValueProvider (built-in class), 90
Newtonsoft.Json.Linq.MergeArrayHandling (built-in class), 78	Newtonsoft.Json.Serialization.JsonArrayContract (built-in class), 93
Newtonsoft.Json.MemberSerialization (built-in class), 78	Newtonsoft.Json.Serialization.JsonContainerContract (built-in class), 93
Newtonsoft.Json.MetadataPropertyHandling (built-in class), 79	Newtonsoft.Json.Serialization.JsonContract (built-in class), 93
Newtonsoft.Json.MissingMemberHandling (built-in class), 79	Newtonsoft.Json.Serialization.JsonDictionaryContract (built-in class), 95
Newtonsoft.Json.NullValueHandling (built-in class), 79	Newtonsoft.Json.Serialization.JsonDynamicContract (built-in class), 96
Newtonsoft.Json.ObjectCreationHandling (built-in class), 79	Newtonsoft.Json.Serialization.JsonLinqContract (built-in class), 96
Newtonsoft.Json.PreserveReferencesHandling (built-in class), 80	Newtonsoft.Json.Serialization.JsonObjectContract (built-in class), 96
Newtonsoft.Json.ReferenceLoopHandling (built-in class), 80	Newtonsoft.Json.Serialization.JsonPrimitiveContract (built-in class), 97
Newtonsoft.Json.Required (built-in class), 80	Newtonsoft.Json.Serialization.JsonProperty (built-in class), 97
Newtonsoft.Json.Schema.Extensions (built-in class), 81	Newtonsoft.Json.Serialization.JsonPropertyCollection (built-in class), 99
Newtonsoft.Json.Schema.JsonSchema (built-in class), 81	Newtonsoft.Json.Serialization.JsonStringContract (built-in class), 100
Newtonsoft.Json.Schema.JsonSchemaException (built-in class), 84	Newtonsoft.Json.Serialization.MemoryTraceWriter (built-in class), 100
Newtonsoft.Json.Schema.JsonSchemaGenerator (built-in class), 85	Newtonsoft.Json.Serialization.OnErrorAttribute (built-in class), 101
Newtonsoft.Json.Schema.JsonSchemaResolver (built-in class), 85	Newtonsoft.Json.Serialization.ReflectionAttributeProvider (built-in class), 101
Newtonsoft.Json.Schema.JsonSchemaType (built-in class), 86	Newtonsoft.Json.Serialization.ReflectionValueProvider (built-in class), 101
Newtonsoft.Json.Schema.UndefinedSchemaIdHandling (built-in class), 86	Newtonsoft.Json.Serialization.SerializationCallback (built-in class), 95
Newtonsoft.Json.Schema.ValidationEventArgs (built-in class), 87	Newtonsoft.Json.Serialization.SerializationErrorCallback (built-in class), 95
Newtonsoft.Json.Schema.ValidationEventHandler (built-in class), 87	Newtonsoft.Json.SerializationBinder (built-in class), 87
Newtonsoft.Json.Serialization.CamelCasePropertyNamesContractResolver (built-in class), 87	Newtonsoft.Json.StringEscapeHandling (built-in class), 102
Newtonsoft.Json.Serialization.DefaultContractResolver (built-in class), 88	Newtonsoft.Json.TraceLevel (built-in class), 102
Newtonsoft.Json.Serialization.DefaultSerializationBinder (built-in class), 91	Newtonsoft.Json.TypeNameHandling (built-in class), 102
Newtonsoft.Json.Serialization.ErrorContext (built-in class), 92	Newtonsoft.Json.Utilities.StringBuffer (built-in class), 104
Newtonsoft.Json.Serialization.ErrorEventArgs (built-in class), 92	Newtonsoft.Json.WriteState (built-in class), 105
Newtonsoft.Json.Serialization.ExpressionValueProvider (built-in class), 90	Next (Newtonsoft.Json.Linq.JToken attribute), 61
Newtonsoft.Json.Serialization.ExtensionDataGetter (built-in class), 95	None (Newtonsoft.Json.DateParseHandling attribute), 22
Newtonsoft.Json.Serialization.ExtensionDataSetter (built-in class), 95	None (Newtonsoft.Json.Formatting attribute), 24
Newtonsoft.Json.Serialization.IAttributeProvider (built-in class), 92	None (Newtonsoft.Json.JsonToken attribute), 46
Newtonsoft.Json.Serialization.IContractResolver (built-in class), 88	None (Newtonsoft.Json.Linq.JTokenType attribute), 74
	None (Newtonsoft.Json.PreserveReferencesHandling at-

- tribute), 80
- None (Newtonsoft.Json.Schema.JsonSchemaType attribute), 86
- None (Newtonsoft.Json.Schema.UndefinedSchemaIdHandling attribute), 86
- None (Newtonsoft.Json.TypeNameHandling attribute), 102
- Null (Newtonsoft.Json.JsonConvert attribute), 26
- Null (Newtonsoft.Json.JsonToken attribute), 47
- Null (Newtonsoft.Json.Linq.JTokenType attribute), 75
- Null (Newtonsoft.Json.Schema.JsonSchemaType attribute), 86
- NullOrEmptyString() (Newtonsoft.Json.Utilities.StringBuffer.Newtonsoft.Json.Utilities.StringBuffer method), 104
- NullValueHandling (Newtonsoft.Json.JsonPropertyAttribute attribute), 35
- NullValueHandling (Newtonsoft.Json.JsonSerializer attribute), 39
- NullValueHandling (Newtonsoft.Json.JsonSerializerSettings attribute), 40
- NullValueHandling (Newtonsoft.Json.Serialization.JsonProperty attribute), 98
- O**
- Object (Newtonsoft.Json.JsonReader.State attribute), 3
- Object (Newtonsoft.Json.Linq.JTokenType attribute), 74
- Object (Newtonsoft.Json.Schema.JsonSchemaType attribute), 86
- Object (Newtonsoft.Json.WriteState attribute), 105
- ObjectCreationHandling (Newtonsoft.Json.JsonPropertyAttribute attribute), 35
- ObjectCreationHandling (Newtonsoft.Json.JsonSerializer attribute), 39
- ObjectCreationHandling (Newtonsoft.Json.JsonSerializerSettings attribute), 40
- ObjectCreationHandling (Newtonsoft.Json.Serialization.JsonProperty attribute), 98
- Objects (Newtonsoft.Json.PreserveReferencesHandling attribute), 80
- Objects (Newtonsoft.Json.TypeNameHandling attribute), 102
- ObjectStart (Newtonsoft.Json.JsonReader.State attribute), 3
- Off (Newtonsoft.Json.TraceLevel attribute), 102
- OmitRootObject (Newtonsoft.Json.Converters.XmlNodeConverter attribute), 21
- OnCollectionChanged() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JContainer method), 62
- OnDeserialized (Newtonsoft.Json.Serialization.JsonContract attribute), 94
- OnDeserializedCallbacks (Newtonsoft.Json.Serialization.JsonContract attribute), 94
- OnDeserializing (Newtonsoft.Json.Serialization.JsonContract attribute), 94
- OnDeserializingCallbacks (Newtonsoft.Json.Serialization.JsonContract attribute), 94
- OnError (Newtonsoft.Json.Serialization.JsonContract attribute), 94
- OnErrorCallbacks (Newtonsoft.Json.Serialization.JsonContract attribute), 94
- OnPropertyChanged() (Newtonsoft.Json.Linq.JObject method), 69
- OnSerialized (Newtonsoft.Json.Serialization.JsonContract attribute), 94
- OnSerializedCallbacks (Newtonsoft.Json.Serialization.JsonContract attribute), 94
- OnSerializing (Newtonsoft.Json.Serialization.JsonContract attribute), 94
- OnSerializingCallbacks (Newtonsoft.Json.Serialization.JsonContract attribute), 94
- OptIn (Newtonsoft.Json.MemberSerialization attribute), 78
- OptOut (Newtonsoft.Json.MemberSerialization attribute), 78
- Order (Newtonsoft.Json.JsonPropertyAttribute attribute), 35
- Order (Newtonsoft.Json.Serialization.JsonProperty attribute), 97
- OriginalObject (Newtonsoft.Json.Serialization.ErrorContext attribute), 92
- OverrideConstructor (Newtonsoft.Json.Serialization.JsonObjectContract attribute), 97
- OverrideCreator (Newtonsoft.Json.Serialization.JsonObjectContract attribute), 97
- P**
- ParametrizedConstructor (Newtonsoft.Json.Serialization.JsonObjectContract attribute), 97

- attribute), 97
 - Parent (Newtonsoft.Json.Linq.JToken attribute), 61
 - Parse() (Newtonsoft.Json.Linq.JObject method), 68
 - Parse() (Newtonsoft.Json.Linq.JToken method), 60
 - Parse() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JArray method), 64
 - Parse() (Newtonsoft.Json.Schema.JsonSchema method), 82
 - Path (Newtonsoft.Json.JsonReader attribute), 3
 - Path (Newtonsoft.Json.JsonReaderException attribute), 36
 - Path (Newtonsoft.Json.JsonValidatingReader attribute), 48
 - Path (Newtonsoft.Json.JsonWriter attribute), 10
 - Path (Newtonsoft.Json.JsonWriterException attribute), 49
 - Path (Newtonsoft.Json.Linq.JToken attribute), 62
 - Path (Newtonsoft.Json.Linq.JTokenReader attribute), 74
 - Path (Newtonsoft.Json.Schema.JsonSchemaException attribute), 84
 - Path (Newtonsoft.Json.Schema.ValidationEventArgs attribute), 87
 - Path (Newtonsoft.Json.Serialization.ErrorContext attribute), 92
 - Pattern (Newtonsoft.Json.Schema.JsonSchema attribute), 82
 - PatternProperties (Newtonsoft.Json.Schema.JsonSchema attribute), 83
 - Populate (Newtonsoft.Json.DefaultValueHandling attribute), 23
 - Populate() (Newtonsoft.Json.JsonSerializer method), 37
 - PopulateObject() (Newtonsoft.Json.JsonConvert method), 31
 - PopulateObjectAsync() (Newtonsoft.Json.JsonConvert method), 31
 - PositionalItemsValidation (Newtonsoft.Json.Schema.JsonSchema attribute), 83
 - PositiveInfinity (Newtonsoft.Json.JsonConvert attribute), 26
 - PostValue (Newtonsoft.Json.JsonReader.State attribute), 4
 - PreserveReferencesHandling (Newtonsoft.Json.JsonSerializer attribute), 38
 - PreserveReferencesHandling (Newtonsoft.Json.JsonSerializerSettings attribute), 40
 - Previous (Newtonsoft.Json.Linq.JToken attribute), 62
 - Properties (Newtonsoft.Json.Schema.JsonSchema attribute), 83
 - Properties (Newtonsoft.Json.Serialization.JsonDynamicContract attribute), 96
 - Properties (Newtonsoft.Json.Serialization.JsonObjectContract attribute), 96
 - Properties() (Newtonsoft.Json.Linq.Extensions method), 49
 - Properties() (Newtonsoft.Json.Linq.JObject method), 67
 - Property (Newtonsoft.Json.JsonReader.State attribute), 3
 - Property (Newtonsoft.Json.Linq.JTokenType attribute), 74
 - Property (Newtonsoft.Json.WriteState attribute), 105
 - Property() (Newtonsoft.Json.Linq.JObject method), 67
 - PropertyName (Newtonsoft.Json.JsonPropertyAttribute attribute), 35
 - PropertyName (Newtonsoft.Json.JsonToken attribute), 46
 - PropertyName (Newtonsoft.Json.Serialization.JsonProperty attribute), 97
 - PropertyNameResolver (Newtonsoft.Json.Serialization.JsonDictionaryContract attribute), 95
 - PropertyNameResolver (Newtonsoft.Json.Serialization.JsonDynamicContract attribute), 96
 - PropertyType (Newtonsoft.Json.Serialization.JsonProperty attribute), 98
 - PropertyValues() (Newtonsoft.Json.Linq.JObject method), 68
- ## Q
- QuoteChar (Newtonsoft.Json.JsonReader attribute), 2
 - QuoteChar (Newtonsoft.Json.JsonTextWriter attribute), 46
 - QuoteChar (Newtonsoft.Json.JsonValidatingReader attribute), 48
 - QuoteName (Newtonsoft.Json.JsonTextWriter attribute), 46
- ## R
- Raw (Newtonsoft.Json.JsonToken attribute), 46
 - Raw (Newtonsoft.Json.Linq.JTokenType attribute), 75
 - Read() (Newtonsoft.Json.JsonReader method), 1
 - Read() (Newtonsoft.Json.JsonReader.State.Newtonsoft.Json.Bson.BsonReader method), 5
 - Read() (Newtonsoft.Json.JsonTextReader method), 42
 - Read() (Newtonsoft.Json.JsonValidatingReader method), 48
 - Read() (Newtonsoft.Json.Linq.JTokenReader method), 74
 - Read() (Newtonsoft.Json.Schema.JsonSchema method), 81
 - Readable (Newtonsoft.Json.Serialization.JsonProperty attribute), 98
 - ReadAhead (Newtonsoft.Json.MetadataPropertyHandling attribute), 79
 - ReadAsBytes() (Newtonsoft.Json.JsonReader method), 2
 - ReadAsBytes() (Newtonsoft.Json.JsonReader.State.Newtonsoft.Json.Bson.BsonReader method), 4

ReadAsBytes()	(Newtonsoft.Json.JsonTextReader method), 42	soft.Json.JsonReader.State.Newtonsoft.Json.Bson.BsonReader method), 4
ReadAsBytes()	(Newtonsoft.Json.JsonValidatingReader method), 47	ReadAsString() (Newtonsoft.Json.JsonTextReader method), 42
ReadAsBytes()	(Newtonsoft.Json.Linq.JTokenReader method), 73	ReadAsString() (Newtonsoft.Json.JsonValidatingReader method), 47
ReadAsDateTime()	(Newtonsoft.Json.JsonReader method), 2	ReadAsString() (Newtonsoft.Json.Linq.JTokenReader method), 73
ReadAsDateTime()	(Newtonsoft.Json.JsonReader.State.Newtonsoft.Json.Bson.BsonReader attribute), 34	ReadData (Newtonsoft.Json.JsonExtensionDataAttribute attribute), 48
ReadAsDateTime()	(Newtonsoft.Json.JsonTextReader method), 42	ReadFrom() (Newtonsoft.Json.Linq.JToken method), 60
ReadAsDateTime()	(Newtonsoft.Json.JsonValidatingReader method), 47	ReadJson() (Newtonsoft.Json.Converters.DiscriminatedUnionConverter method), 16
ReadAsDateTime()	(Newtonsoft.Json.Linq.JTokenReader method), 74	ReadJson() (Newtonsoft.Json.Converters.ExpandoObjectConverter method), 17
ReadAsDateTimeOffset()	(Newtonsoft.Json.JsonReader method), 2	ReadJson() (Newtonsoft.Json.Converters.IsoDateTimeConverter method), 18
ReadAsDateTimeOffset()	(Newtonsoft.Json.JsonReader.State.Newtonsoft.Json.Bson.BsonReader attribute), 5	ReadJson() (Newtonsoft.Json.Converters.JavaScriptDateTimeConverter method), 18
ReadAsDateTimeOffset()	(Newtonsoft.Json.JsonTextReader method), 42	ReadJson() (Newtonsoft.Json.Converters.KeyValuePairConverter method), 19
ReadAsDateTimeOffset()	(Newtonsoft.Json.JsonValidatingReader method), 48	ReadJson() (Newtonsoft.Json.Converters.RegexConverter method), 19
ReadAsDateTimeOffset()	(Newtonsoft.Json.Linq.JTokenReader method), 74	ReadJson() (Newtonsoft.Json.Converters.StringEnumConverter method), 20
ReadAsDecimal()	(Newtonsoft.Json.JsonReader method), 2	ReadJson() (Newtonsoft.Json.Converters.VersionConverter method), 20
ReadAsDecimal()	(Newtonsoft.Json.JsonReader.State.Newtonsoft.Json.Bson.BsonReader attribute), 4	ReadJson() (Newtonsoft.Json.Converters.XmlNodeConverter method), 21
ReadAsDecimal()	(Newtonsoft.Json.JsonTextReader method), 42	ReadJson() (Newtonsoft.Json.JsonConverter method), 14
ReadAsDecimal()	(Newtonsoft.Json.JsonValidatingReader method), 47	ReadJson() (Newtonsoft.Json.JsonConverter.Newtonsoft.Json.Converters.BsonReader attribute), 15
ReadAsDecimal()	(Newtonsoft.Json.Linq.JTokenReader method), 73	ReadOnly (Newtonsoft.Json.Schema.JsonSchema attribute), 82
ReadAsInt32()	(Newtonsoft.Json.JsonReader method), 1	ReadRootValueAsArray (Newtonsoft.Json.JsonReader.State.Newtonsoft.Json.Bson.BsonReader attribute), 5
ReadAsInt32()	(Newtonsoft.Json.JsonReader.State.Newtonsoft.Json.Bson.BsonReader attribute), 4	ReferenceLoopHandling (Newtonsoft.Json.JsonPropertyAttribute attribute), 35
ReadAsInt32()	(Newtonsoft.Json.JsonTextReader method), 42	ReferenceLoopHandling (Newtonsoft.Json.JsonSerializer attribute), 39
ReadAsInt32()	(Newtonsoft.Json.JsonValidatingReader method), 47	ReferenceLoopHandling (Newtonsoft.Json.JsonSerializerSettings attribute), 40
ReadAsInt32()	(Newtonsoft.Json.Linq.JTokenReader method), 73	ReferenceLoopHandling (Newtonsoft.Json.Serialization.JsonProperty attribute), 98
ReadAsString()	(Newtonsoft.Json.JsonReader method), 1	ReferenceResolver (Newtonsoft.Json.JsonSerializer attribute), 38
ReadAsString()	(Newtonsoft.Json.JsonReader.State.Newtonsoft.Json.Bson.BsonReader attribute), 4	ReferenceResolver (Newtonsoft.Json.JsonSerializerSettings attribute),

41	Remove() (Newtonsoft.Json.Linq.IObject method), 68	SelectToken() (Newtonsoft.Json.Linq.JToken method), 60
	Remove() (Newtonsoft.Json.Linq.JToken method), 52	SelectTokens() (Newtonsoft.Json.Linq.JToken method), 60
	Remove() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JArray method), 65	Serialize (Newtonsoft.Json.ReferenceLoopHandling attribute), 80
	RemoveAll() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JContainer method), 63	Serialize() (Newtonsoft.Json.JsonSerializer method), 38
	RemoveAnnotations() (Newtonsoft.Json.Linq.JToken method), 61	SerializeCompilerGeneratedMembers (Newtonsoft.Json.Serialization.IContractResolver.Newtonsoft.Json.Serialization.JsonConverter attribute), 90
	RemoveAt() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JContainer method), 64	SerializeObject() (Newtonsoft.Json.JsonConvert method), 28, 29
	Replace (Newtonsoft.Json.Linq.MergeArrayHandling attribute), 78	SerializeObjectAsync() (Newtonsoft.Json.JsonConvert method), 29
	Replace (Newtonsoft.Json.ObjectCreationHandling attribute), 79	SerializeXmlNode() (Newtonsoft.Json.JsonConvert method), 32
	Replace() (Newtonsoft.Json.Linq.JToken method), 52	SetIsSpecified (Newtonsoft.Json.Serialization.JsonProperty attribute), 99
	ReplaceAll() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JContainer method), 63	SetMemberValue() (Newtonsoft.Json.TypeNameHandling.Newtonsoft.Json.Utilities.ReflectionHelper method), 104
	Required (Newtonsoft.Json.JsonPropertyAttribute attribute), 35	SetStateBasedOnCurrent() (Newtonsoft.Json.JsonReader method), 2
	Required (Newtonsoft.Json.Schema.JsonSchema attribute), 82	SetToken() (Newtonsoft.Json.JsonReader method), 2
	Required (Newtonsoft.Json.Serialization.JsonProperty attribute), 98	SetValue() (Newtonsoft.Json.Serialization.IValueProvider method), 90
	Requires (Newtonsoft.Json.Schema.JsonSchema attribute), 84	SetValue() (Newtonsoft.Json.Serialization.IValueProvider.Newtonsoft.Json.JsonReader method), 91
	ResolveContract() (Newtonsoft.Json.Serialization.IContractResolver method), 88	SetValue() (Newtonsoft.Json.Serialization.ReflectionValueProvider method), 101
	ResolveContract() (Newtonsoft.Json.Serialization.IContractResolver.Newtonsoft.Json.Serialization.JsonConverter method), 88	SoftWriteSerializer (Newtonsoft.Json.DefaultContractResolver attribute), 10
	ResolveContractConverter() (Newtonsoft.Json.Serialization.IContractResolver.Newtonsoft.Json.Serialization.JsonConverter method), 89	ShouldSerialize (Newtonsoft.Json.Serialization.IContractResolver attribute), 99
	ResolvePropertyName() (Newtonsoft.Json.Serialization.IContractResolver.Newtonsoft.Json.Serialization.JsonConverter method), 90	Simple (System.Runtime.Serialization.Formatters.FormatterAssemblyStyle attribute), 2
	ResolvePropertyName() (Newtonsoft.Json.Serialization.IContractResolver.Newtonsoft.Json.Serialization.JsonConverter method), 90	Skip() (Newtonsoft.Json.JsonReader method), 2
	ResolveReference() (Newtonsoft.Json.Serialization.IReferenceResolver method), 91	Start (Newtonsoft.Json.JsonReader.State attribute), 3
	Reuse (Newtonsoft.Json.ObjectCreationHandling attribute), 79	StartArray (Newtonsoft.Json.JsonToken attribute), 46
	Root (Newtonsoft.Json.Linq.JToken attribute), 61	StartConstructor (Newtonsoft.Json.JsonToken attribute), 46
	RoundtripKind (Newtonsoft.Json.DateTimeZoneHandling attribute), 22	StartObject (Newtonsoft.Json.JsonToken attribute), 46
		String (Newtonsoft.Json.FloatFormatHandling attribute), 23
		String (Newtonsoft.Json.Linq.JTokenType attribute), 74
		String (Newtonsoft.Json.Schema.JsonSchemaType attribute), 86
		StringEscapeHandling (Newtonsoft.Json.JsonSerializer attribute), 40
S		
	Schema (Newtonsoft.Json.JsonValidatingReader attribute), 41	StringEscapeHandling (Newtonsoft.Json.JsonSerializer attribute), 40

- soft.Json.JsonSerializerSettings attribute), 42
- StringEscapeHandling (Newtonsoft.Json.JsonWriter attribute), 11
- SupportMultipleContent (Newtonsoft.Json.JsonReader attribute), 2
- Symbol (Newtonsoft.Json.FloatFormatHandling attribute), 23
- System.Runtime.Serialization.FormatterAssemblyStyle (built-in class), 23
- ## T
- TimeSpan (Newtonsoft.Json.Linq.JTokenType attribute), 75
- Title (Newtonsoft.Json.JsonContainerAttribute attribute), 25
- Title (Newtonsoft.Json.Schema.JsonSchema attribute), 82
- Token (Newtonsoft.Json.Linq.JTokenWriter attribute), 78
- TokenType (Newtonsoft.Json.JsonReader attribute), 3
- TokenType (Newtonsoft.Json.JsonValidatingReader attribute), 48
- ToObject() (Newtonsoft.Json.Linq.JToken method), 59
- Top (Newtonsoft.Json.JsonWriter attribute), 10
- ToString() (Newtonsoft.Json.JsonConvert method), 26–28
- ToString() (Newtonsoft.Json.Linq.JToken method), 52
- ToString() (Newtonsoft.Json.Linq.JValue method), 72
- ToString() (Newtonsoft.Json.Schema.JsonSchema method), 82
- ToString() (Newtonsoft.Json.Serialization.JsonProperty method), 97
- ToString() (Newtonsoft.Json.Serialization.MemoryTraceWriter method), 100
- Trace() (Newtonsoft.Json.Serialization.ITraceWriter method), 93
- Trace() (Newtonsoft.Json.Serialization.MemoryTraceWriter method), 100
- TraceWriter (Newtonsoft.Json.JsonSerializer attribute), 38
- TraceWriter (Newtonsoft.Json.JsonSerializerSettings attribute), 41
- Transient (Newtonsoft.Json.Schema.JsonSchema attribute), 82
- True (Newtonsoft.Json.JsonConvert attribute), 26
- TryGetValue() (Newtonsoft.Json.Linq.JObject method), 68, 69
- Type (Newtonsoft.Json.Linq.JConstructor attribute), 66
- Type (Newtonsoft.Json.Linq.JObject attribute), 69
- Type (Newtonsoft.Json.Linq.JProperty attribute), 70
- Type (Newtonsoft.Json.Linq.JToken attribute), 61
- Type (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JTokenType attribute), 65
- Type (Newtonsoft.Json.Linq.JValue attribute), 72
- Type (Newtonsoft.Json.Schema.JsonSchema attribute), 82
- TypeNameAssemblyFormat (Newtonsoft.Json.JsonSerializer attribute), 38
- TypeNameAssemblyFormat (Newtonsoft.Json.JsonSerializerSettings attribute), 41
- TypeNameHandling (Newtonsoft.Json.JsonPropertyAttribute attribute), 35
- TypeNameHandling (Newtonsoft.Json.JsonSerializer attribute), 38
- TypeNameHandling (Newtonsoft.Json.JsonSerializerSettings attribute), 40
- TypeNameHandling (Newtonsoft.Json.Serialization.JsonProperty attribute), 99
- ## U
- Undefined (Newtonsoft.Json.JsonConvert attribute), 26
- Undefined (Newtonsoft.Json.JsonToken attribute), 47
- Undefined (Newtonsoft.Json.Linq.JTokenType attribute), 75
- UndefinedSchemaIdHandling (Newtonsoft.Json.Schema.JsonSchemaGenerator attribute), 85
- UnderlyingName (Newtonsoft.Json.Serialization.JsonProperty attribute), 97
- UnderlyingType (Newtonsoft.Json.Serialization.JsonContract attribute), 93
- Union (Newtonsoft.Json.Linq.MergeArrayHandling attribute), 78
- UniqueItems (Newtonsoft.Json.Schema.JsonSchema attribute), 83
- Unspecified (Newtonsoft.Json.DateTimeZoneHandling attribute), 22
- Uri (Newtonsoft.Json.Linq.JTokenType attribute), 75
- UseAssemblyQualifiedName (Newtonsoft.Json.Schema.UndefinedSchemaIdHandling attribute), 86
- UseTypeName (Newtonsoft.Json.Schema.UndefinedSchemaIdHandling attribute), 86
- Utc (Newtonsoft.Json.DateTimeZoneHandling attribute), 22
- ## V
- Validate() (Newtonsoft.Json.Schema.Extensions method), 81
- Value (Newtonsoft.Json.Bson.BsonObjectId attribute), 1
- Value (Newtonsoft.Json.JsonReader attribute), 3

Value (Newtonsoft.Json.JsonValidatingReader attribute), 48
 Value (Newtonsoft.Json.Linq.JProperty attribute), 70
 Value (Newtonsoft.Json.Linq.JValue attribute), 72
 ValueProvider (Newtonsoft.Json.Serialization.JsonProperty attribute), 98
 Values() (Newtonsoft.Json.Linq.Extensions method), 49, 50
 ValueType (Newtonsoft.Json.JsonReader attribute), 3
 ValueType (Newtonsoft.Json.JsonValidatingReader attribute), 48
 Verbose (Newtonsoft.Json.TraceLevel attribute), 102

W

Warning (Newtonsoft.Json.TraceLevel attribute), 102
 Writable (Newtonsoft.Json.Serialization.JsonProperty attribute), 98
 WriteArrayAttribute (Newtonsoft.Json.Converters.XmlNodeConverter attribute), 21
 WriteComment() (Newtonsoft.Json.JsonTextWriter method), 46
 WriteComment() (Newtonsoft.Json.JsonWriter method), 10
 WriteComment() (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter method), 11
 WriteComment() (Newtonsoft.Json.Linq.JTokenWriter method), 76
 WriteData (Newtonsoft.Json.JsonExtensionDataAttribute attribute), 34
 WriteEnd() (Newtonsoft.Json.JsonTextWriter method), 43
 WriteEnd() (Newtonsoft.Json.JsonWriter method), 6
 WriteEnd() (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter method), 11
 WriteEnd() (Newtonsoft.Json.Linq.JTokenWriter method), 76
 WriteEndArray() (Newtonsoft.Json.JsonWriter method), 6
 WriteEndConstructor() (Newtonsoft.Json.JsonWriter method), 6
 WriteEndObject() (Newtonsoft.Json.JsonWriter method), 5
 WriteIndent() (Newtonsoft.Json.JsonTextWriter method), 43
 WriteIndent() (Newtonsoft.Json.JsonWriter method), 6
 WriteIndentSpace() (Newtonsoft.Json.JsonTextWriter method), 43
 WriteIndentSpace() (Newtonsoft.Json.JsonWriter method), 7
 WriteJson() (Newtonsoft.Json.Converters.DiscriminatedUnionConverter method), 16

WriteJson() (Newtonsoft.Json.Converters.ExpandoObjectConverter method), 17
 WriteJson() (Newtonsoft.Json.Converters.IsoDateTimeConverter method), 17
 WriteJson() (Newtonsoft.Json.Converters.JavaScriptDateTimeConverter method), 18
 WriteJson() (Newtonsoft.Json.Converters.KeyValuePairConverter method), 18
 WriteJson() (Newtonsoft.Json.Converters.RegexConverter method), 19
 WriteJson() (Newtonsoft.Json.Converters.StringEnumConverter method), 20
 WriteJson() (Newtonsoft.Json.Converters.VersionConverter method), 20
 WriteJson() (Newtonsoft.Json.Converters.XmlNodeConverter method), 21
 WriteJson() (Newtonsoft.Json.JsonConverter method), 14
 WriteJson() (Newtonsoft.Json.JsonConverter.Newtonsoft.Json.Converters.BsonWriter method), 15
 WriteNull() (Newtonsoft.Json.JsonTextWriter method), 44
 WriteNull() (Newtonsoft.Json.JsonWriter method), 7
 WriteNull() (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter method), 12
 WriteNull() (Newtonsoft.Json.Linq.JTokenWriter method), 76
 WriteObjectId() (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter method), 14
 WritePropertyName() (Newtonsoft.Json.JsonTextWriter method), 43
 WritePropertyName() (Newtonsoft.Json.JsonWriter method), 6
 WritePropertyName() (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter method), 12
 WritePropertyName() (Newtonsoft.Json.Linq.JTokenWriter method), 76
 WriteRaw() (Newtonsoft.Json.JsonTextWriter method), 44
 WriteRaw() (Newtonsoft.Json.JsonWriter method), 7
 WriteRaw() (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter method), 11
 WriteRaw() (Newtonsoft.Json.Linq.JTokenWriter method), 76
 WriteRawValue() (Newtonsoft.Json.JsonWriter method), 7
 WriteRawValue() (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter method), 11
 WriteRegex() (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter method), 14
 WriteStartArray() (Newtonsoft.Json.JsonTextWriter method), 16

method), 43

WriteStartArray() (Newtonsoft.Json.JsonWriter method), 6

WriteStartArray() (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter method), 11

WriteStartArray() (Newtonsoft.Json.Linq.JTokenWriter method), 75

WriteStartConstructor() (Newtonsoft.Json.JsonTextWriter method), 43

WriteStartConstructor() (Newtonsoft.Json.JsonWriter method), 6

WriteStartConstructor() (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter method), 11

WriteStartConstructor() (Newtonsoft.Json.Linq.JTokenWriter method), 75

WriteStartObject() (Newtonsoft.Json.JsonTextWriter method), 43

WriteStartObject() (Newtonsoft.Json.JsonWriter method), 5

WriteStartObject() (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter method), 11

WriteStartObject() (Newtonsoft.Json.Linq.JTokenWriter method), 75

WriteState (Newtonsoft.Json.JsonWriter attribute), 10

WriteTo() (Newtonsoft.Json.Linq.JConstructor method), 66

WriteTo() (Newtonsoft.Json.Linq.JObject method), 68

WriteTo() (Newtonsoft.Json.Linq.JProperty method), 69

WriteTo() (Newtonsoft.Json.Linq.JToken method), 52

WriteTo() (Newtonsoft.Json.Linq.JToken.Newtonsoft.Json.Linq.JArray method), 64

WriteTo() (Newtonsoft.Json.Linq.JValue method), 71

WriteTo() (Newtonsoft.Json.Schema.JsonSchema method), 82

WriteToken() (Newtonsoft.Json.JsonWriter method), 6

WriteUndefined() (Newtonsoft.Json.JsonTextWriter method), 44

WriteUndefined() (Newtonsoft.Json.JsonWriter method), 7

WriteUndefined() (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter method), 12

WriteUndefined() (Newtonsoft.Json.Linq.JTokenWriter method), 76

WriteValue() (Newtonsoft.Json.JsonTextWriter method), 43–45

WriteValue() (Newtonsoft.Json.JsonWriter method), 7–10

WriteValue() (Newtonsoft.Json.JsonWriter.Newtonsoft.Json.Bson.BsonWriter method), 12, 13

WriteValue() (Newtonsoft.Json.Linq.JTokenWriter method), 76–78

WriteValueDelimiter() (Newtonsoft.Json.JsonTextWriter method), 43

WriteValueDelimiter() (Newtonsoft.Json.JsonWriter method), 7

WriteWhitespace() (Newtonsoft.Json.JsonTextWriter method), 46

WriteWhitespace() (Newtonsoft.Json.JsonWriter method), 10