

# **XBinder**

---

XML Schema Compiler  
Version 2.2  
C JSON Runtime  
Reference Manual



The software described in this document is furnished under a license agreement and may be used only in accordance with the terms of this agreement.

### **Copyright Notice**

Copyright ©1997–2010 Objective Systems, Inc. All rights reserved.

This document may be distributed in any form, electronic or otherwise, provided that it is distributed in its entirety and that the copyright and this notice are included.

### **Author's Contact Information**

Comments, suggestions, and inquiries regarding XBinder may be submitted via electronic mail to [info@obj-sys.com](mailto:info@obj-sys.com).



# Contents

<b>1</b>	<b>Main Page</b>	<b>1</b>
<b>2</b>	<b>Module Index</b>	<b>2</b>
2.1	Modules . . . . .	2
<b>3</b>	<b>File Index</b>	<b>3</b>
3.1	File List . . . . .	3
<b>4</b>	<b>Module Documentation</b>	<b>4</b>
4.1	JSON encode functions. . . . .	4
4.1.1	Function Documentation . . . . .	6
4.1.1.1	rtJsonEncAnyAttr . . . . .	6
4.1.1.2	rtJsonEncBase64StrValue . . . . .	6
4.1.1.3	rtJsonEncBoolValue . . . . .	6
4.1.1.4	rtJsonEncDate . . . . .	7
4.1.1.5	rtJsonEncDateTime . . . . .	7
4.1.1.6	rtJsonEncDecimalValue . . . . .	7
4.1.1.7	rtJsonEncDoubleValue . . . . .	8
4.1.1.8	rtJsonEncFloatValue . . . . .	8
4.1.1.9	rtJsonEncGDay . . . . .	8
4.1.1.10	rtJsonEncGMonth . . . . .	9
4.1.1.11	rtJsonEncGMonthDay . . . . .	9
4.1.1.12	rtJsonEncGYear . . . . .	9
4.1.1.13	rtJsonEncGYearMonth . . . . .	10
4.1.1.14	rtJsonEncHexStr . . . . .	10
4.1.1.15	rtJsonEncIndent . . . . .	10
4.1.1.16	rtJsonEncInt64Value . . . . .	11
4.1.1.17	rtJsonEncIntValue . . . . .	11
4.1.1.18	rtJsonEncStringObject . . . . .	11

4.1.1.19	rtJsonEncStringObject2	12
4.1.1.20	rtJsonEncStringPair	12
4.1.1.21	rtJsonEncStringPair2	12
4.1.1.22	rtJsonEncStringValue	13
4.1.1.23	rtJsonEncStringValue2	13
4.1.1.24	rtJsonEncTime	13
4.1.1.25	rtJsonEncUInt64Value	14
4.1.1.26	rtJsonEncUIntValue	14
4.1.1.27	rtJsonEncUnicodeData	14
4.2	JSON decode functions.	15
4.2.1	Function Documentation	17
4.2.1.1	rtJsonDecAnyElem	17
4.2.1.2	rtJsonDecAnyType	17
4.2.1.3	rtJsonDecBase64Str	18
4.2.1.4	rtJsonDecBool	18
4.2.1.5	rtJsonDecDate	18
4.2.1.6	rtJsonDecDateTime	19
4.2.1.7	rtJsonDecDecimal	19
4.2.1.8	rtJsonDecDouble	19
4.2.1.9	rtJsonDecDynBase64Str	20
4.2.1.10	rtJsonDecDynHexStr	20
4.2.1.11	rtJsonDecGDay	20
4.2.1.12	rtJsonDecGMonth	21
4.2.1.13	rtJsonDecGMonthDay	21
4.2.1.14	rtJsonDecGYear	21
4.2.1.15	rtJsonDecGYearMonth	22
4.2.1.16	rtJsonDecHexStr	22
4.2.1.17	rtJsonDecInt16Value	22
4.2.1.18	rtJsonDecInt32Value	23
4.2.1.19	rtJsonDecInt64Value	23
4.2.1.20	rtJsonDecInt8Value	23
4.2.1.21	rtJsonDecMatchChar	24
4.2.1.22	rtJsonDecMatchObjectStart	24
4.2.1.23	rtJsonDecMatchToken	24
4.2.1.24	rtJsonDecMatchToken2	25
4.2.1.25	rtJsonDecNameValuePair	25
4.2.1.26	rtJsonDecPeekChar	25

4.2.1.27	rtJsonDecStringObject	26
4.2.1.28	rtJsonDecStringValue	26
4.2.1.29	rtJsonDecTime	26
4.2.1.30	rtJsonDecUInt16Value	27
4.2.1.31	rtJsonDecUInt32Value	27
4.2.1.32	rtJsonDecUInt64Value	27
4.2.1.33	rtJsonDecUInt8Value	28
4.2.1.34	rtJsonDecXmlStringValue	28
4.2.1.35	rtJsonGetElemIdx	29
<b>5</b>	<b>File Documentation</b>	<b>30</b>
5.1	osrtjson.h File Reference	30
5.1.1	Detailed Description	34

# Chapter 1

## Main Page

### C JSON Runtime Library Functions

The **C run-time JSON library** contains functions used to encode/decode data in Javascript object notation (JSON). These functions are identified by their *rtJson* prefixes.



# Chapter 2

## Module Index

### 2.1 Modules

Here is a list of all modules:

JSON encode functions. . . . .	4
JSON decode functions. . . . .	15

# Chapter 3

## File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">osrtjson.h</a> (JSON low-level C encode/decode functions) . . . . .	30
---	----

# Chapter 4

## Module Documentation

### 4.1 JSON encode functions.

#### Functions

- EXTERNJSON int [rtJsonEncAnyAttr](#) (OSCTXT \*pctx, const OSRTDList \*pvalue)  
*This function encodes a list of OSAnyAttr attributes in which the name and value are given as a UTF-8 string.*
- EXTERNJSON int [rtJsonEncIntValue](#) (OSCTXT \*pctx, OSINT32 value)  
*This function encodes a variable of the XSD integer type.*
- EXTERNJSON int [rtJsonEncInt64Value](#) (OSCTXT \*pctx, OSINT64 value)  
*This function encodes a variable of the XSD integer type.*
- EXTERNJSON int [rtJsonEncBase64StrValue](#) (OSCTXT \*pctx, OSUINT32 noctx, const OSOCTET \*value)  
*This function encodes a variable of the XSD base64Binary type.*
- EXTERNJSON int [rtJsonEncBoolValue](#) (OSCTXT \*pctx, OSBOOL value)  
*This function encodes a variable of the XSD boolean type.*
- EXTERNJSON int [rtJsonEncGYear](#) (OSCTXT \*pctx, const OSXSDDateTime \*pvalue)  
*This function encodes a numeric gYear value into a JSON string representation.*
- EXTERNJSON int [rtJsonEncGYearMonth](#) (OSCTXT \*pctx, const OSXSDDateTime \*pvalue)  
*This function encodes a numeric gYearMonth value into a JSON string representation.*
- EXTERNJSON int [rtJsonEncGMonth](#) (OSCTXT \*pctx, const OSXSDDateTime \*pvalue)  
*This function encodes a numeric gMonth value into a JSON string representation.*
- EXTERNJSON int [rtJsonEncGMonthDay](#) (OSCTXT \*pctx, const OSXSDDateTime \*pvalue)  
*This function encodes a numeric gMonthDay value into a JSON string representation.*
- EXTERNJSON int [rtJsonEncGDay](#) (OSCTXT \*pctx, const OSXSDDateTime \*pvalue)  
*This function encodes a numeric gDay value into a JSON string representation.*
- EXTERNJSON int [rtJsonEncDate](#) (OSCTXT \*pctx, const OSXSDDateTime \*pvalue)

*This function encodes a variable of the XSD 'date' type as a string.*

- EXTERNJSON int [rtJsonEncTime](#) (OSCTXT \*pctx, const OSXSDDateTime \*pvalue)

*This function encodes a variable of the XSD 'time' type as a JSON string.*

- EXTERNJSON int [rtJsonEncDateTime](#) (OSCTXT \*pctx, const OSXSDDateTime \*pvalue)

*This function encodes a numeric date/time value into a string representation.*

- EXTERNJSON int [rtJsonEncDecimalValue](#) (OSCTXT \*pctx, OSREAL value, const OSDecimalFmt \*pFmtSpec)

*This function encodes a value of the XSD decimal type.*

- EXTERNJSON int [rtJsonEncDoubleValue](#) (OSCTXT \*pctx, OSREAL value, const OSDoubleFmt \*pFmtSpec)

*This function encodes a value of the XSD double or float type.*

- EXTERNJSON int [rtJsonEncFloatValue](#) (OSCTXT \*pctx, OSREAL value, const OSDoubleFmt \*pFmtSpec)

*This function encodes a variable of the XSD float type.*

- EXTERNJSON int [rtJsonEncHexStr](#) (OSCTXT \*pctx, OSUINT32 noctx, const OSOCTET \*data)

*This function encodes a variable of the XSD hexBinary type.*

- EXTERNJSON int [rtJsonEncIndent](#) (OSCTXT \*pctx)

*This function adds indentation whitespace to the output stream.*

- EXTERNJSON int [rtJsonEncStringObject](#) (OSCTXT \*pctx, const OSUTF8CHAR \*name, const OSUTF8CHAR \*value)

*This function encodes a JSON object containing a string value.*

- EXTERNJSON int [rtJsonEncStringObject2](#) (OSCTXT \*pctx, const OSUTF8CHAR \*name, size\_t nameLen, const OSUTF8CHAR \*value, size\_t valueLen)

*This function encodes a JSON object containing a string value.*

- EXTERNJSON int [rtJsonEncStringPair](#) (OSCTXT \*pctx, const OSUTF8CHAR \*name, const OSUTF8CHAR \*value)

*This function encodes a name/value pair.*

- EXTERNJSON int [rtJsonEncStringPair2](#) (OSCTXT \*pctx, const OSUTF8CHAR \*name, size\_t nameLen, const OSUTF8CHAR \*value, size\_t valueLen)

*This function encodes a name/value pair.*

- EXTERNJSON int [rtJsonEncStringValue](#) (OSCTXT \*pctx, const OSUTF8CHAR \*value)

*This function encodes a variable of the XSD string type.*

- EXTERNJSON int [rtJsonEncStringValue2](#) (OSCTXT \*pctx, const OSUTF8CHAR \*value, size\_t valueLen)

*This function encodes a variable of the XSD string type.*

- EXTERNJSON int [rtJsonEncUnicodeData](#) (OSCTXT \*pctx, const OSUNICHAR \*value, OSUINT32 nchars)

*This function encodes a variable of Unicode value.*

- EXTERNJSON int [rtJsonEncUIntValue](#) (OSCTXT \*pctxt, OSUINT32 value)  
*This function encodes a variable of the XSD unsigned integer type.*
- EXTERNJSON int [rtJsonEncUInt64Value](#) (OSCTXT \*pctxt, OSUINT64 value)  
*This function encodes a variable of the XSD integer type.*

## 4.1.1 Function Documentation

### 4.1.1.1 EXTERNJSON int [rtJsonEncAnyAttr](#) (OSCTXT \*pctxt, const OSRTDList \*pvalue)

This function encodes a list of OSAnyAttr attributes in which the name and value are given as a UTF-8 string.

#### Parameters

- pctxt* Pointer to context block structure.
- pvalue* List of attributes.

#### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.1.1.2 EXTERNJSON int [rtJsonEncBase64StrValue](#) (OSCTXT \*pctxt, OSUINT32 nocts, const OSOCTET \*value)

This function encodes a variable of the XSD base64Binary type.

#### Parameters

- pctxt* Pointer to context block structure.
- nocts* Number of octets in the value string.
- value* Value to be encoded.

#### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.1.1.3 EXTERNJSON int [rtJsonEncBoolValue](#) (OSCTXT \*pctxt, OSBOOL value)

This function encodes a variable of the XSD boolean type.

#### Parameters

- pctxt* Pointer to context block structure.
- value* Boolean value to be encoded.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.1.1.4 EXTERNJSON int rtJsonEncDate (OSCTXT \* *pctxt*, const OSXSDDateTime \* *pvalue*)

This function encodes a variable of the XSD 'date' type as a string.

This version of the function is used to encode an OSXSDDateTime value into CCYY-MM-DD format.

## Parameters

*pctxt* Pointer to context block structure.

*pvalue* OSXSDDateTime type pointer points to a OSXSDDateTime value to be encoded.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.1.1.5 EXTERNJSON int rtJsonEncDateTime (OSCTXT \* *pctxt*, const OSXSDDateTime \* *pvalue*)

This function encodes a numeric date/time value into a string representation.

## Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to value to be encoded.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.1.1.6 EXTERNJSON int rtJsonEncDecimalValue (OSCTXT \* *pctxt*, OSREAL *value*, const OSDecimalFmt \* *pFmtSpec*)

This function encodes a value of the XSD decimal type.

## Parameters

*pctxt* Pointer to context block structure.

*value* Value to be encoded.

*pFmtSpec* Pointer to format specification structure.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.1.1.7 EXTERNJSON int rtJsonEncDoubleValue (OSCTXT \* *pctxt*, OSREAL *value*, const OSDoubleFmt \* *pFmtSpec*)

This function encodes a value of the XSD double or float type.

##### Parameters

- pctxt* Pointer to context block structure.
- value* Value to be encoded.
- pFmtSpec* Pointer to format specification structure.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.1.1.8 EXTERNJSON int rtJsonEncFloatValue (OSCTXT \* *pctxt*, OSREAL *value*, const OSDoubleFmt \* *pFmtSpec*)

This function encodes a variable of the XSD float type.

##### Parameters

- pctxt* Pointer to context block structure.
- value* Value to be encoded.
- pFmtSpec* Pointer to format specification structure.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.1.1.9 EXTERNJSON int rtJsonEncGDay (OSCTXT \* *pctxt*, const OSXSDDateTime \* *pvalue*)

This function encodes a numeric gDay value into a JSON string representation.

##### Parameters

- pctxt* Pointer to context block structure.
- pvalue* Pointer to value to be encoded.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.1.1.10 EXTERNJSON int rtJsonEncGMonth (OSCTXT \* *pctxt*, const OSXSDDateTime \* *pvalue*)

This function encodes a numeric gMonth value into a JSON string representation.

##### Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to value to be encoded.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.1.1.11 EXTERNJSON int rtJsonEncGMonthDay (OSCTXT \* *pctxt*, const OSXSDDateTime \* *pvalue*)

This function encodes a numeric gMonthDay value into a JSON string representation.

##### Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to value to be encoded.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.1.1.12 EXTERNJSON int rtJsonEncGYear (OSCTXT \* *pctxt*, const OSXSDDateTime \* *pvalue*)

This function encodes a numeric gYear value into a JSON string representation.

##### Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to value to be encoded.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.



#### 4.1.1.13 EXTERNJSON int rt.JsonEncGYearMonth (OSCTXT \* *pctxt*, const OSXSDDateTime \* *pvalue*)

This function encodes a numeric gYearMonth value into a JSON string representation.

##### Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to value to be encoded.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.1.1.14 EXTERNJSON int rt.JsonEncHexStr (OSCTXT \* *pctxt*, OSUINT32 *nocts*, const OSOCTET \* *data*)

This function encodes a variable of the XSD hexBinary type.

##### Parameters

*pctxt* Pointer to context block structure.

*nocts* Number of octets in the value string.

*data* Value to be encoded.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.1.1.15 EXTERNJSON int rt.JsonEncIndent (OSCTXT \* *pctxt*)

This function adds indentation whitespace to the output stream.

The amount of indentation to add is determined by the level member variable in the context structure and the OSXMLINDENT constant value.

##### Parameters

*pctxt* Pointer to context block structure.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.1.1.16 EXTERNJSON int rtJsonEncInt64Value (OSCTXT \* *pctxt*, OSINT64 *value*)

This function encodes a variable of the XSD integer type.

This version of the function is used for 64-bit integer values.

##### Parameters

*pctxt* Pointer to context block structure.

*value* Value to be encoded.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.1.1.17 EXTERNJSON int rtJsonEncIntValue (OSCTXT \* *pctxt*, OSINT32 *value*)

This function encodes a variable of the XSD integer type.

##### Parameters

*pctxt* Pointer to context block structure.

*value* Value to be encoded.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.1.1.18 EXTERNJSON int rtJsonEncStringObject (OSCTXT \* *pctxt*, const OSUTF8CHAR \* *name*, const OSUTF8CHAR \* *value*)

This function encodes a JSON object containing a string value.

##### Parameters

*pctxt* Pointer to context block structure.

*name* Name token to be encoded.

*value* Value as a character string to be encoded.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

**4.1.1.19 EXTERNJSON int rtJsonEncStringObject2 (OSCTXT \* *pctxt*, const OSUTF8CHAR \* *name*, size\_t *nameLen*, const OSUTF8CHAR \* *value*, size\_t *valueLen*)**

This function encodes a JSON object containing a string value.

**Parameters**

- pctxt* Pointer to context block structure.
- name* Name token to be encoded.
- nameLen* Length of the name token to be encoded.
- value* Value as a character string to be encoded.
- valueLen* Length of the value to be encoded.

**Returns**

- Completion status of operation:
- 0 = success,
  - negative return value is error.

**4.1.1.20 EXTERNJSON int rtJsonEncStringPair (OSCTXT \* *pctxt*, const OSUTF8CHAR \* *name*, const OSUTF8CHAR \* *value*)**

This function encodes a name/value pair.

The value is a character string.

**Parameters**

- pctxt* Pointer to context block structure.
- name* Name token to be encoded.
- value* Value as a character string to be encoded.

**Returns**

- Completion status of operation:
- 0 = success,
  - negative return value is error.

**4.1.1.21 EXTERNJSON int rtJsonEncStringPair2 (OSCTXT \* *pctxt*, const OSUTF8CHAR \* *name*, size\_t *nameLen*, const OSUTF8CHAR \* *value*, size\_t *valueLen*)**

This function encodes a name/value pair.

The value is a character string.

**Parameters**

- pctxt* Pointer to context block structure.
- name* Name token to be encoded.
- nameLen* Length of the name token to be encoded.

*value* Value as a character string to be encoded.

*valueLen* Length of the value to be encoded.

### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.1.1.22 EXTERNJSON int rtJsonEncStringValue (OSCTXT \* *pctxt*, const OSUTF8CHAR \* *value*)

This function encodes a variable of the XSD string type.

### Parameters

*pctxt* Pointer to context block structure.

*value* XML string value to be encoded.

### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.1.1.23 EXTERNJSON int rtJsonEncStringValue2 (OSCTXT \* *pctxt*, const OSUTF8CHAR \* *value*, size\_t *valueLen*)

This function encodes a variable of the XSD string type.

### Parameters

*pctxt* Pointer to context block structure.

*value* XML string value to be encoded.

*valueLen* Length of the XML string to be encoded.

### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.1.1.24 EXTERNJSON int rtJsonEncTime (OSCTXT \* *pctxt*, const OSXSDDateTime \* *pvalue*)

This function encodes a variable of the XSD 'time' type as a JSON string.

### Parameters

*pctxt* Pointer to context block structure.

*pvalue* OSXSDDateTime type pointer points to a OSXSDDateTime value to be encoded.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.1.1.25 EXTERNJSON int rt.JsonEncUInt64Value (OSCTXT \* *pctxt*, OSUINT64 *value*)

This function encodes a variable of the XSD integer type.

This version of the function is used when constraints cause an unsigned 64-bit integer variable to be used.

## Parameters

*pctxt* Pointer to context block structure.

*value* Value to be encoded.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.1.1.26 EXTERNJSON int rt.JsonEncUIntValue (OSCTXT \* *pctxt*, OSUINT32 *value*)

This function encodes a variable of the XSD unsigned integer type.

## Parameters

*pctxt* Pointer to context block structure.

*value* Value to be encoded.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.1.1.27 EXTERNJSON int rt.JsonEncUnicodeData (OSCTXT \* *pctxt*, const OSUNICHAR \* *value*, OSUINT32 *nchars*)

This function encodes a variable of Unicode value.

## Parameters

*pctxt* Pointer to context block structure.

*value* Unicode characters to be encoded.

*nchars* Number of Unicode characters to be encoded.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

## 4.2 JSON decode functions.

### Functions

- EXTERNJSON int [rtJsonDecAnyElem](#) (OSCTXT \*pctxt, OSUTF8CHAR \*\*ppvalue)  
*This function decodes an arbitrary block of JSON-encoded data into a string variable.*
- EXTERNJSON int [rtJsonDecAnyType](#) (OSCTXT \*pctxt, OSUTF8CHAR \*\*ppvalue)  
*This function decodes an arbitrary block of JSON-encoded data into a string variable.*
- EXTERNJSON int [rtJsonDecBase64Str](#) (OSCTXT \*pctxt, OSOCTET \*pvalue, OSUINT32 \*pnocets, size\_t bufsize)  
*This function decodes a contents of a Base64-encode binary string into a static memory structure.*
- EXTERNJSON int [rtJsonDecDynBase64Str](#) (OSCTXT \*pctxt, OSDynOctStr \*pvalue)  
*This function decodes a contents of a Base64-encode binary string.*
- EXTERNJSON int [rtJsonDecBool](#) (OSCTXT \*pctxt, OSBOOL \*pvalue)  
*This function decodes a variable of the boolean type.*
- EXTERNJSON int [rtJsonDecDate](#) (OSCTXT \*pctxt, OSXSDDateTime \*pvalue)  
*This function decodes a variable of the XSD 'date' type.*
- EXTERNJSON int [rtJsonDecTime](#) (OSCTXT \*pctxt, OSXSDDateTime \*pvalue)  
*This function decodes a variable of the XSD 'time' type.*
- EXTERNJSON int [rtJsonDecDateTime](#) (OSCTXT \*pctxt, OSXSDDateTime \*pvalue)  
*This function decodes a variable of the XSD 'dateTime' type.*
- EXTERNJSON int [rtJsonDecGYear](#) (OSCTXT \*pctxt, OSXSDDateTime \*pvalue)  
*This function decodes a variable of the XSD 'gYear' type.*
- EXTERNJSON int [rtJsonDecGYearMonth](#) (OSCTXT \*pctxt, OSXSDDateTime \*pvalue)  
*This function decodes a variable of the XSD 'gYearMonth' type.*
- EXTERNJSON int [rtJsonDecGMonth](#) (OSCTXT \*pctxt, OSXSDDateTime \*pvalue)  
*This function decodes a variable of the XSD 'gMonth' type.*
- EXTERNJSON int [rtJsonDecGMonthDay](#) (OSCTXT \*pctxt, OSXSDDateTime \*pvalue)  
*This function decodes a variable of the XSD 'gMonthDay' type.*
- EXTERNJSON int [rtJsonDecGDay](#) (OSCTXT \*pctxt, OSXSDDateTime \*pvalue)  
*This function decodes a variable of the XSD 'gDay' type.*
- EXTERNJSON int [rtJsonDecDecimal](#) (OSCTXT \*pctxt, OSREAL \*pvalue, int totalDigits, int fractionDigits)  
*This function decodes the contents of a decimal data type.*
- EXTERNJSON int [rtJsonDecDouble](#) (OSCTXT \*pctxt, OSREAL \*pvalue)  
*This function decodes the contents of a float or double data type.*

- EXTERNJSON int [rtJsonDecHexStr](#) (OSCTXT \*pctxt, OSOCTET \*pvalue, OSUINT32 \*pnocts, size\_t bufsize)  
*This function decodes the contents of a hexBinary string into a static memory structure.*
- EXTERNJSON int [rtJsonDecDynHexStr](#) (OSCTXT \*pctxt, OSDynOctStr \*pvalue)  
*This function decodes a contents of a hexBinary string.*
- EXTERNJSON int [rtJsonDecInt8Value](#) (OSCTXT \*pctxt, OSINT8 \*pvalue)  
*This function decodes the contents of an 8-bit integer data type (i.e.*
- EXTERNJSON int [rtJsonDecInt16Value](#) (OSCTXT \*pctxt, OSINT16 \*pvalue)  
*This function decodes the contents of a 16-bit integer data type.*
- EXTERNJSON int [rtJsonDecInt32Value](#) (OSCTXT \*pctxt, OSINT32 \*pvalue)  
*This function decodes the contents of a 32-bit integer data type.*
- EXTERNJSON int [rtJsonDecInt64Value](#) (OSCTXT \*pctxt, OSINT64 \*pvalue)  
*This function decodes the contents of a 64-bit integer data type.*
- EXTERNJSON int [rtJsonDecUInt8Value](#) (OSCTXT \*pctxt, OSUINT8 \*pvalue)  
*This function decodes the contents of an unsigned 8-bit integer data type (i.e.*
- EXTERNJSON int [rtJsonDecUInt16Value](#) (OSCTXT \*pctxt, OSUINT16 \*pvalue)  
*This function decodes the contents of an unsigned 16-bit integer data type.*
- EXTERNJSON int [rtJsonDecUInt32Value](#) (OSCTXT \*pctxt, OSUINT32 \*pvalue)  
*This function decodes the contents of an unsigned 32-bit integer data type.*
- EXTERNJSON int [rtJsonDecUInt64Value](#) (OSCTXT \*pctxt, OSUINT64 \*pvalue)  
*This function decodes the contents of an unsigned 64-bit integer data type.*
- EXTERNJSON int [rtJsonDecMatchChar](#) (OSCTXT \*pctxt, OSUTF8CHAR ch)  
*This function parses the next byte and matches it with given character.*
- EXTERNJSON int [rtJsonDecMatchObjectStart](#) (OSCTXT \*pctxt, const OSUTF8NameAndLen \*nameArray, size\_t numNames)  
*This function parses the next object start tag that matches with one of the names in the given name array.*
- EXTERNJSON int [rtJsonDecMatchToken](#) (OSCTXT \*pctxt, const OSUTF8CHAR \*token)  
*This function decodes a JSON string and matches with a given token.*
- EXTERNJSON int [rtJsonDecMatchToken2](#) (OSCTXT \*pctxt, const OSUTF8CHAR \*token, size\_t tokenLen)  
*This function decodes a JSON string and matches with a given token.*
- EXTERNJSON int [rtJsonDecNameValuePair](#) (OSCTXT \*pctxt, OSUTF8NVP \*pvalue)  
*This function decodes a name/value pair.*
- EXTERNJSON int [rtJsonDecPeekChar](#) (OSCTXT \*pctxt, OSUTF8CHAR \*pch)  
*This function read the next character.*

- EXTERNJSON int [rtJsonDecStringObject](#) (OSCTXT \*pctx, const OSUTF8CHAR \*name, OSUTF8CHAR \*\*ppvalue)

*This function decodes the contents of a JSON object containing a character string.*

- EXTERNJSON int [rtJsonDecStringValue](#) (OSCTXT \*pctx, OSUTF8CHAR \*\*ppvalue)

*This function decodes the contents of a string data type.*

- EXTERNJSON int [rtJsonDecXmlStringValue](#) (OSCTXT \*pctx, OSXMLSTRING \*pvalue)

*This function decodes the contents of an XML string data type.*

- EXTERNJSON size\_t [rtJsonGetElemIdx](#) (OSCTXT \*pctx, const OSUTF8NameAndLen nameArray[], size\_t nrows)

*This function parses the next start tag and finds the index of the element name in the descriptor table.*

## 4.2.1 Function Documentation

### 4.2.1.1 EXTERNJSON int [rtJsonDecAnyElem](#) (OSCTXT \*pctx, OSUTF8CHAR \*\*ppvalue)

This function decodes an arbitrary block of JSON-encoded data into a string variable.

In this case, the expected format is element name : JSON encoded data.

#### Parameters

*pctx* A pointer to a context structure. This provides a storage area for the function to store all working variables that must be maintained between function calls.

*ppvalue* A pointer to a variable to receive the decoded JSON text.

#### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.2 EXTERNJSON int [rtJsonDecAnyType](#) (OSCTXT \*pctx, OSUTF8CHAR \*\*ppvalue)

This function decodes an arbitrary block of JSON-encoded data into a string variable.

In this case, the expected format is a complete JSON encoded data fragment.

#### Parameters

*pctx* A pointer to a context structure. This provides a storage area for the function to store all working variables that must be maintained between function calls.

*ppvalue* A pointer to a variable to receive the decoded JSON text.

#### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.



#### 4.2.1.3 EXTERNJSON int rtJsonDecBase64Str (OSCTXT \* *pctxt*, OSOCTET \* *pvalue*, OSUINT32 \* *pnocts*, size\_t *bufsize*)

This function decodes a contents of a Base64-encode binary string into a static memory structure.

The octet string must be Base64 encoded. This function call is used to decode a sized base64Binary string production.

##### Parameters

*pctxt* A pointer to a context structure. This provides a storage area for the function to store all working variables that must be maintained between function calls.

*pvalue* A pointer to a variable to receive the decoded bit string. This is assumed to be a static array large enough to hold the number of octets specified in the bufsize input parameter.

*pnocts* A pointer to an integer value to receive the decoded number of octets.

*bufsize* The size (in octets) of the sized octet string. An error will occur if the number of octets in the decoded string is larger than this value.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.2.1.4 EXTERNJSON int rtJsonDecBool (OSCTXT \* *pctxt*, OSBOOL \* *pvalue*)

This function decodes a variable of the boolean type.

##### Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to a variable to receive the decoded boolean value.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.2.1.5 EXTERNJSON int rtJsonDecDate (OSCTXT \* *pctxt*, OSXSDDateTime \* *pvalue*)

This function decodes a variable of the XSD 'date' type.

Input is expected to be a string of characters returned by a JSON parser. The string should have CCYY-MM-DD format.

##### Parameters

*pctxt* Pointer to context block structure.

*pvalue* OSXSDDateTime type pointer points to a OSXSDDateTime value to receive decoded result.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.2.1.6 EXTERNJSON int rtJsonDecDateTime (OSCTXT \* *pctxt*, OSXSDDateTime \* *pvalue*)

This function decodes a variable of the XSD 'dateTime' type.

Input is expected to be a string of characters returned by an JSON parser.

##### Parameters

*pctxt* Pointer to context block structure.

*pvalue* OSXSDDateTime type pointer points to a OSXSDDateTime value to receive decoded result.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.2.1.7 EXTERNJSON int rtJsonDecDecimal (OSCTXT \* *pctxt*, OSREAL \* *pvalue*, int *totalDigits*, int *fractionDigits*)

This function decodes the contents of a decimal data type.

Input is expected to be a string of OSUTF8CHAR characters returned by a JSON parser.

##### Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to 64-bit double value to receive decoded result.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.2.1.8 EXTERNJSON int rtJsonDecDouble (OSCTXT \* *pctxt*, OSREAL \* *pvalue*)

This function decodes the contents of a float or double data type.

Input is expected to be a string of OSUTF8CHAR characters returned by a JSON parser.

##### Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to 64-bit double value to receive decoded result.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.2.1.9 EXTERNJSON int rtJsonDecDynBase64Str (OSCTXT \* *pctxt*, OSDynOctStr \* *pvalue*)

This function decodes a contents of a Base64-encode binary string.

The octet string must be Base64 encoded. This function will allocate dynamic memory to store the decoded result.

##### Parameters

*pctxt* A pointer to a context structure. This provides a storage area for the function to store all working variables that must be maintained between function calls.

*pvalue* A pointer to a dynamic octet string structure to receive the decoded octet string. Dynamic memory is allocated for the string using the rtxMemAlloc function.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.2.1.10 EXTERNJSON int rtJsonDecDynHexStr (OSCTXT \* *pctxt*, OSDynOctStr \* *pvalue*)

This function decodes a contents of a hexBinary string.

This function will allocate dynamic memory to store the decoded result. Input is expected to be a string of OS-UTF8CHAR characters returned by a JSON parser.

##### Parameters

*pctxt* A pointer to a context structure. This provides a storage area for the function to store all working variables that must be maintained between function calls.

*pvalue* A pointer to a dynamic octet string structure to receive the decoded octet string. Dynamic memory is allocated to hold the string using the rtxMemAlloc function.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.2.1.11 EXTERNJSON int rtJsonDecGDay (OSCTXT \* *pctxt*, OSXSDDateTime \* *pvalue*)

This function decodes a variable of the XSD 'gDay' type.

Input is expected to be a string of characters returned by a JSON parser. The string should have ---DD[+hh:mm|Z] format.

##### Parameters

*pctxt* Pointer to context block structure.

*pvalue* OSXSDDateTime type pointer points to a OSXSDDateTime value to receive decoded result.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.2.1.12 EXTERNJSON int rtJsonDecGMonth (OSCTXT \* *pctxt*, OSXSDDateTime \* *pvalue*)

This function decodes a variable of the XSD 'gMonth' type.

Input is expected to be a string of characters returned by a JSON parser. The string should have --MM[+hh:mm|Z] format.

##### Parameters

*pctxt* Pointer to context block structure.

*pvalue* OSXSDDateTime type pointer points to a OSXSDDateTime value to receive decoded result.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.2.1.13 EXTERNJSON int rtJsonDecGMonthDay (OSCTXT \* *pctxt*, OSXSDDateTime \* *pvalue*)

This function decodes a variable of the XSD 'gMonthDay' type.

Input is expected to be a string of characters returned by a JSON parser. The string should have --MM-DD[+hh:mm|Z] format.

##### Parameters

*pctxt* Pointer to context block structure.

*pvalue* OSXSDDateTime type pointer points to a OSXSDDateTime value to receive decoded result.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.2.1.14 EXTERNJSON int rtJsonDecGYear (OSCTXT \* *pctxt*, OSXSDDateTime \* *pvalue*)

This function decodes a variable of the XSD 'gYear' type.

Input is expected to be a string of characters returned by a JSON parser. The string should have CCYY[+hh:mm|Z] format.

##### Parameters

*pctxt* Pointer to context block structure.

*pvalue* OSXSDDateTime type pointer points to a OSXSDDateTime value to receive decoded result.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.2.1.15 EXTERNJSON int rtJsonDecGYearMonth (OSCTXT \* *pctxt*, OSXSDDateTime \* *pvalue*)

This function decodes a variable of the XSD 'gYearMonth' type.

Input is expected to be a string of characters returned by a JSON parser. The string should have CCYY-MM[-hh:mm|Z] format.

##### Parameters

*pctxt* Pointer to context block structure.

*pvalue* OSXSDDateTime type pointer points to a OSXSDDateTime value to receive decoded result.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.2.1.16 EXTERNJSON int rtJsonDecHexStr (OSCTXT \* *pctxt*, OSOCTET \* *pvalue*, OSUINT32 \* *pnocts*, size\_t *bufsize*)

This function decodes the contents of a hexBinary string into a static memory structure.

Input is expected to be a string of OSUTF8CHAR characters returned by a JSON parser.

##### Parameters

*pctxt* A pointer to a context structure. This provides a storage area for the function to store all working variables that must be maintained between function calls.

*pvalue* A pointer to a variable to receive the decoded bit string. This is assumed to be a static array large enough to hold the number of octets specified in the bufsize input parameter.

*pnocts* A pointer to an integer value to receive the decoded number of octets.

*bufsize* The size (in octets) of the sized octet string. An error will occur if the number of octets in the decoded string is larger than this value.

##### Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

#### 4.2.1.17 EXTERNJSON int rtJsonDecInt16Value (OSCTXT \* *pctxt*, OSINT16 \* *pvalue*)

This function decodes the contents of a 16-bit integer data type.

Input is expected to be a string of OSUTF8CHAR characters returned by a JSON parser.

##### Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to 16-bit integer value to receive decoded result.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.18 EXTERNJSON int rtJsonDecInt32Value (OSCTXT \* *pctxt*, OSINT32 \* *pvalue*)

This function decodes the contents of a 32-bit integer data type.

Input is expected to be a string of OSUTF8CHAR characters returned by a JSON parser.

## Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to 32-bit integer value to receive decoded result.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.19 EXTERNJSON int rtJsonDecInt64Value (OSCTXT \* *pctxt*, OSINT64 \* *pvalue*)

This function decodes the contents of a 64-bit integer data type.

Input is expected to be a string of OSUTF8CHAR characters returned by a JSON parser.

## Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to 64-bit integer value to receive decoded result.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.20 EXTERNJSON int rtJsonDecInt8Value (OSCTXT \* *pctxt*, OSINT8 \* *pvalue*)

This function decodes the contents of an 8-bit integer data type (i.e.

a signed byte type in the range of -128 to 127). Input is expected to be a string of OSUTF8CHAR characters returned by a JSON parser.

## Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to 8-bit integer value to receive decoded result.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.21 EXTERNJSON int rtJsonDecMatchChar (OSCTXT \* *pctxt*, OSUTF8CHAR *ch*)

This function parses the next byte and matches it with given character.

## Parameters

*pctxt* Pointer to context block structure.

*ch* The character to be matched.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.22 EXTERNJSON int rtJsonDecMatchObjectStart (OSCTXT \* *pctxt*, const OSUTF8NameAndLen \* *nameArray*, size\_t *numNames*)

This function parses the next object start tag that matches with one of the names in the given name array.

It also matches the the leading '{' of the object and ':' character after the name.

## Parameters

*pctxt* Pointer to context block structure.

*nameArray* Array of names to be matched.

*numNames* Number of names in the name array

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.23 EXTERNJSON int rtJsonDecMatchToken (OSCTXT \* *pctxt*, const OSUTF8CHAR \* *token*)

This function decodes a JSON string and matches with a given token.

## Parameters

*pctxt* A pointer to a context structure. This provides a storage area for the function to store all working variables that must be maintained between function calls.

*token* The token to be matched.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.24 EXTERNJSON int rtJsonDecMatchToken2 (OSCTXT \* *pctxt*, const OSUTF8CHAR \* *token*, size\_t *tokenLen*)

This function decodes a JSON string and matches with a given token.

## Parameters

*pctxt* A pointer to a context structure. This provides a storage area for the function to store all working variables that must be maintained between function calls.

*token* The token to be matched.

*tokenLen* The length of the token to be matched.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.25 EXTERNJSON int rtJsonDecNameValuePair (OSCTXT \* *pctxt*, OSUTF8NVP \* *pvalue*)

This function decodes a name/value pair.

## Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to an structure to receive the decoded name and value.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.26 EXTERNJSON int rtJsonDecPeekChar (OSCTXT \* *pctxt*, OSUTF8CHAR \* *pch*)

This function read the next character.

It does not decode the value.

## Parameters

*pctxt* Pointer to OSCTXT structure

*pch* A pointer to a variable to receive the next character.



## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.27 EXTERNJSON int rtJsonDecStringObject (OSCTXT \* *pctxt*, const OSUTF8CHAR \* *name*, OSUTF8CHAR \*\* *ppvalue*)

This function decodes the contents of a JSON object containing a character string.

Input is expected to be a complete JSON object returned by a JSON parser.

## Parameters

*pctxt* Pointer to context block structure.

*name* The name token.

*ppvalue* Pointer to a string structure to receive the decoded string. Memory is allocated for the string using the run-time memory manager.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.28 EXTERNJSON int rtJsonDecStringValue (OSCTXT \* *pctxt*, OSUTF8CHAR \*\* *ppvalue*)

This function decodes the contents of a string data type.

This type contains a pointer to a UTF-8 character string. Input is expected to be a string of UTF-8 characters returned by a JSON parser.

## Parameters

*pctxt* Pointer to context block structure.

*ppvalue* Pointer to a string structure to receive the decoded string. Memory is allocated for the string using the run-time memory manager.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.29 EXTERNJSON int rtJsonDecTime (OSCTXT \* *pctxt*, OSXSDDateTime \* *pvalue*)

This function decodes a variable of the XSD 'time' type.

Input is expected to be a string of characters returned by a JSON parser. The string should have one of following formats:

(1) hh-mm-ss.ss used if *tz\_flag* = false (2) hh-mm-ss.ssZ used if *tz\_flag* = false and *tzo* = 0 (3) hh-mm-ss.ss+HH:MM if *tz\_flag* = false and *tzo* > 0 (4) hh-mm-ss.ss-HH:MM-HH:MM if *tz\_flag* = false and *tzo* < 0

## Parameters

*pctxt* Pointer to context block structure.

*pvalue* OSXSDDateTime type pointer points to a OSXSDDateTime value to receive decoded result.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.30 EXTERNJSON int rtJsonDecUInt16Value (OSCTXT \* *pctxt*, OSUINT16 \* *pvalue*)

This function decodes the contents of an unsigned 16-bit integer data type.

Input is expected to be a string of OSUTF8CHAR characters returned by a JSON parser.

## Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to unsigned 16-bit integer value to receive decoded result.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.31 EXTERNJSON int rtJsonDecUInt32Value (OSCTXT \* *pctxt*, OSUINT32 \* *pvalue*)

This function decodes the contents of an unsigned 32-bit integer data type.

Input is expected to be a string of OSUTF8CHAR characters returned by a JSON parser.

## Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to unsigned 32-bit integer value to receive decoded result.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.32 EXTERNJSON int rtJsonDecUInt64Value (OSCTXT \* *pctxt*, OSUINT64 \* *pvalue*)

This function decodes the contents of an unsigned 64-bit integer data type.

Input is expected to be a string of OSUTF8CHAR characters returned by a JSON parser.

## Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to unsigned 64-bit integer value to receive decoded result.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.33 EXTERNJSON int rtJsonDecUInt8Value (OSCTXT \* *pctxt*, OSUINT8 \* *pvalue*)

This function decodes the contents of an unsigned 8-bit integer data type (i.e.

a signed byte type in the range of 0 to 255). Input is expected to be a string of OSUTF8CHAR characters returned by a JSON parser.

## Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to unsigned 8-bit integer value to receive decoded result.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

### 4.2.1.34 EXTERNJSON int rtJsonDecXmlStringValue (OSCTXT \* *pctxt*, OSXMLSTRING \* *pvalue*)

This function decodes the contents of an XML string data type.

This type contains a pointer to a UTF-8 character string plus flags that can be set to alter the encoding of the string (for example, the cdata flag allows the string to be encoded in a CDATA section). Input is expected to be a string of UTF-8 characters returned by a JSON parser.

## Parameters

*pctxt* Pointer to context block structure.

*pvalue* Pointer to an XML string structure to receive the decoded string. Memory is allocated for the string using the run-time memory manager.

## Returns

Completion status of operation:

- 0 = success,
- negative return value is error.

**4.2.1.35 EXTERNJSON size\_t rtJsonGetElemIdx (OSCTXT \* *pctxt*, const OSUTF8NameAndLen *nameArray*[], size\_t *nrows*)**

This function parses the next start tag and finds the index of the element name in the descriptor table.

**Parameters**

- pctxt* Pointer to context block structure.
- nameArray* Elements descriptor table.
- nrows* Number of descriptors in table.

**Returns**

Completion status of operation:

- positive or zero value is element identifier,
- negative return value is error.

# Chapter 5

## File Documentation

### 5.1 osrtjson.h File Reference

JSON low-level C encode/decode functions.

```
#include "rtxsrc/osMacros.h"
#include "rtxsrc/rtxCommonDefs.h"
#include "rtxsrc/rtxError.h"
#include "rtxmlsrc/osrtxml.h"
```

#### Functions

- EXTERNJSON int [rtJsonEncAnyAttr](#) (OSCTXT \*pctxt, const OSRTDList \*pvalue)  
*This function encodes a list of OSAnyAttr attributes in which the name and value are given as a UTF-8 string.*
- EXTERNJSON int [rtJsonEncIntValue](#) (OSCTXT \*pctxt, OSINT32 value)  
*This function encodes a variable of the XSD integer type.*
- EXTERNJSON int [rtJsonEncInt64Value](#) (OSCTXT \*pctxt, OSINT64 value)  
*This function encodes a variable of the XSD integer type.*
- EXTERNJSON int [rtJsonEncBase64StrValue](#) (OSCTXT \*pctxt, OSUINT32 nocts, const OSOCTET \*value)  
*This function encodes a variable of the XSD base64Binary type.*
- EXTERNJSON int [rtJsonEncBoolValue](#) (OSCTXT \*pctxt, OSBOOL value)  
*This function encodes a variable of the XSD boolean type.*
- EXTERNJSON int [rtJsonEncGYear](#) (OSCTXT \*pctxt, const OSXSDDateTime \*pvalue)  
*This function encodes a numeric gYear value into a JSON string representation.*
- EXTERNJSON int [rtJsonEncGYearMonth](#) (OSCTXT \*pctxt, const OSXSDDateTime \*pvalue)  
*This function encodes a numeric gYearMonth value into a JSON string representation.*
- EXTERNJSON int [rtJsonEncGMonth](#) (OSCTXT \*pctxt, const OSXSDDateTime \*pvalue)  
*This function encodes a numeric gMonth value into a JSON string representation.*

- EXTERNJSON int [rtJsonEncGMonthDay](#) (OSCTXT \*pctx, const OSXSDDateTime \*pvalue)  
*This function encodes a numeric gMonthDay value into a JSON string representation.*
- EXTERNJSON int [rtJsonEncGDay](#) (OSCTXT \*pctx, const OSXSDDateTime \*pvalue)  
*This function encodes a numeric gDay value into a JSON string representation.*
- EXTERNJSON int [rtJsonEncDate](#) (OSCTXT \*pctx, const OSXSDDateTime \*pvalue)  
*This function encodes a variable of the XSD 'date' type as a string.*
- EXTERNJSON int [rtJsonEncTime](#) (OSCTXT \*pctx, const OSXSDDateTime \*pvalue)  
*This function encodes a variable of the XSD 'time' type as a JSON string.*
- EXTERNJSON int [rtJsonEncDateTime](#) (OSCTXT \*pctx, const OSXSDDateTime \*pvalue)  
*This function encodes a numeric date/time value into a string representation.*
- EXTERNJSON int [rtJsonEncDecimalValue](#) (OSCTXT \*pctx, OSREAL value, const OSDecimalFmt \*pFmtSpec)  
*This function encodes a value of the XSD decimal type.*
- EXTERNJSON int [rtJsonEncDoubleValue](#) (OSCTXT \*pctx, OSREAL value, const OSDoubleFmt \*pFmtSpec)  
*This function encodes a value of the XSD double or float type.*
- EXTERNJSON int [rtJsonEncFloatValue](#) (OSCTXT \*pctx, OSREAL value, const OSDoubleFmt \*pFmtSpec)  
*This function encodes a variable of the XSD float type.*
- EXTERNJSON int [rtJsonEncHexStr](#) (OSCTXT \*pctx, OSUINT32 noctx, const OSOCTET \*data)  
*This function encodes a variable of the XSD hexBinary type.*
- EXTERNJSON int [rtJsonEncIndent](#) (OSCTXT \*pctx)  
*This function adds indentation whitespace to the output stream.*
- EXTERNJSON int [rtJsonEncStringObject](#) (OSCTXT \*pctx, const OSUTF8CHAR \*name, const OSUTF8CHAR \*value)  
*This function encodes a JSON object containing a string value.*
- EXTERNJSON int [rtJsonEncStringObject2](#) (OSCTXT \*pctx, const OSUTF8CHAR \*name, size\_t nameLen, const OSUTF8CHAR \*value, size\_t valueLen)  
*This function encodes a JSON object containing a string value.*
- EXTERNJSON int [rtJsonEncStringPair](#) (OSCTXT \*pctx, const OSUTF8CHAR \*name, const OSUTF8CHAR \*value)  
*This function encodes a name/value pair.*
- EXTERNJSON int [rtJsonEncStringPair2](#) (OSCTXT \*pctx, const OSUTF8CHAR \*name, size\_t nameLen, const OSUTF8CHAR \*value, size\_t valueLen)  
*This function encodes a name/value pair.*
- EXTERNJSON int [rtJsonEncStringValue](#) (OSCTXT \*pctx, const OSUTF8CHAR \*value)

*This function encodes a variable of the XSD string type.*

- EXTERNJSON int [rtJsonEncStringValue2](#) (OSCTXT \*pctx, const OSUTF8CHAR \*value, size\_t valueLen)

*This function encodes a variable of the XSD string type.*

- EXTERNJSON int [rtJsonEncUnicodeData](#) (OSCTXT \*pctx, const OSUNICHAR \*value, OSUINT32 nchars)

*This function encodes a variable of Unicode value.*

- EXTERNJSON int [rtJsonEncUIntValue](#) (OSCTXT \*pctx, OSUINT32 value)

*This function encodes a variable of the XSD unsigned integer type.*

- EXTERNJSON int [rtJsonEncUInt64Value](#) (OSCTXT \*pctx, OSUINT64 value)

*This function encodes a variable of the XSD integer type.*

- EXTERNJSON int [rtJsonDecAnyElem](#) (OSCTXT \*pctx, OSUTF8CHAR \*\*ppvalue)

*This function decodes an arbitrary block of JSON-encoded data into a string variable.*

- EXTERNJSON int [rtJsonDecAnyType](#) (OSCTXT \*pctx, OSUTF8CHAR \*\*ppvalue)

*This function decodes an arbitrary block of JSON-encoded data into a string variable.*

- EXTERNJSON int [rtJsonDecBase64Str](#) (OSCTXT \*pctx, OSOCTET \*pvalue, OSUINT32 \*pnoc, size\_t bufsize)

*This function decodes a contents of a Base64-encode binary string into a static memory structure.*

- EXTERNJSON int [rtJsonDecDynBase64Str](#) (OSCTXT \*pctx, OSDynOctStr \*pvalue)

*This function decodes a contents of a Base64-encode binary string.*

- EXTERNJSON int [rtJsonDecBool](#) (OSCTXT \*pctx, OSBOOL \*pvalue)

*This function decodes a variable of the boolean type.*

- EXTERNJSON int [rtJsonDecDate](#) (OSCTXT \*pctx, OSXSDDateTime \*pvalue)

*This function decodes a variable of the XSD 'date' type.*

- EXTERNJSON int [rtJsonDecTime](#) (OSCTXT \*pctx, OSXSDDateTime \*pvalue)

*This function decodes a variable of the XSD 'time' type.*

- EXTERNJSON int [rtJsonDecDateTime](#) (OSCTXT \*pctx, OSXSDDateTime \*pvalue)

*This function decodes a variable of the XSD 'dateTime' type.*

- EXTERNJSON int [rtJsonDecGYear](#) (OSCTXT \*pctx, OSXSDDateTime \*pvalue)

*This function decodes a variable of the XSD 'gYear' type.*

- EXTERNJSON int [rtJsonDecGYearMonth](#) (OSCTXT \*pctx, OSXSDDateTime \*pvalue)

*This function decodes a variable of the XSD 'gYearMonth' type.*

- EXTERNJSON int [rtJsonDecGMonth](#) (OSCTXT \*pctx, OSXSDDateTime \*pvalue)

*This function decodes a variable of the XSD 'gMonth' type.*

- EXTERNJSON int [rtJsonDecGMonthDay](#) (OSCTXT \*pctx, OSXSDDateTime \*pvalue)

*This function decodes a variable of the XSD 'gMonthDay' type.*

- EXTERNJSON int [rtJsonDecGDay](#) (OSCTXT \*pctxt, OSXSDDateTime \*pvalue)  
*This function decodes a variable of the XSD 'gDay' type.*
- EXTERNJSON int [rtJsonDecDecimal](#) (OSCTXT \*pctxt, OSREAL \*pvalue, int totalDigits, int fractionDigits)  
*This function decodes the contents of a decimal data type.*
- EXTERNJSON int [rtJsonDecDouble](#) (OSCTXT \*pctxt, OSREAL \*pvalue)  
*This function decodes the contents of a float or double data type.*
- EXTERNJSON int [rtJsonDecHexStr](#) (OSCTXT \*pctxt, OSOCTET \*pvalue, OSUINT32 \*pnocts, size\_t bufsize)  
*This function decodes the contents of a hexBinary string into a static memory structure.*
- EXTERNJSON int [rtJsonDecDynHexStr](#) (OSCTXT \*pctxt, OSDynOctStr \*pvalue)  
*This function decodes a contents of a hexBinary string.*
- EXTERNJSON int [rtJsonDecInt8Value](#) (OSCTXT \*pctxt, OSINT8 \*pvalue)  
*This function decodes the contents of an 8-bit integer data type (i.e.*
- EXTERNJSON int [rtJsonDecInt16Value](#) (OSCTXT \*pctxt, OSINT16 \*pvalue)  
*This function decodes the contents of a 16-bit integer data type.*
- EXTERNJSON int [rtJsonDecInt32Value](#) (OSCTXT \*pctxt, OSINT32 \*pvalue)  
*This function decodes the contents of a 32-bit integer data type.*
- EXTERNJSON int [rtJsonDecInt64Value](#) (OSCTXT \*pctxt, OSINT64 \*pvalue)  
*This function decodes the contents of a 64-bit integer data type.*
- EXTERNJSON int [rtJsonDecUInt8Value](#) (OSCTXT \*pctxt, OSUINT8 \*pvalue)  
*This function decodes the contents of an unsigned 8-bit integer data type (i.e.*
- EXTERNJSON int [rtJsonDecUInt16Value](#) (OSCTXT \*pctxt, OSUINT16 \*pvalue)  
*This function decodes the contents of an unsigned 16-bit integer data type.*
- EXTERNJSON int [rtJsonDecUInt32Value](#) (OSCTXT \*pctxt, OSUINT32 \*pvalue)  
*This function decodes the contents of an unsigned 32-bit integer data type.*
- EXTERNJSON int [rtJsonDecUInt64Value](#) (OSCTXT \*pctxt, OSUINT64 \*pvalue)  
*This function decodes the contents of an unsigned 64-bit integer data type.*
- EXTERNJSON int [rtJsonDecMatchChar](#) (OSCTXT \*pctxt, OSUTF8CHAR ch)  
*This function parses the next byte and matches it with given character.*
- EXTERNJSON int [rtJsonDecMatchObjectStart](#) (OSCTXT \*pctxt, const OSUTF8NameAndLen \*nameArray, size\_t numNames)  
*This function parses the next object start tag that matches with one of the names in the given name array.*
- EXTERNJSON int [rtJsonDecMatchToken](#) (OSCTXT \*pctxt, const OSUTF8CHAR \*token)



*This function decodes a JSON string and matches with a given token.*

- EXTERNJSON int [rtJsonDecMatchToken2](#) (OSCTXT \*pctx, const OSUTF8CHAR \*token, size\_t tokenLen)

*This function decodes a JSON string and matches with a given token.*

- EXTERNJSON int [rtJsonDecNameValuePair](#) (OSCTXT \*pctx, OSUTF8NVP \*pvalue)

*This function decodes a name/value pair.*

- EXTERNJSON int [rtJsonDecPeekChar](#) (OSCTXT \*pctx, OSUTF8CHAR \*pch)

*This function read the next character.*

- EXTERNJSON int [rtJsonDecStringObject](#) (OSCTXT \*pctx, const OSUTF8CHAR \*name, OSUTF8CHAR \*\*ppvalue)

*This function decodes the contents of a JSON object containing a character string.*

- EXTERNJSON int [rtJsonDecStringValue](#) (OSCTXT \*pctx, OSUTF8CHAR \*\*ppvalue)

*This function decodes the contents of a string data type.*

- EXTERNJSON int [rtJsonDecXmlStringValue](#) (OSCTXT \*pctx, OSXMLSTRING \*pvalue)

*This function decodes the contents of an XML string data type.*

- EXTERNJSON size\_t [rtJsonGetElemIdx](#) (OSCTXT \*pctx, const OSUTF8NameAndLen nameArray[], size\_t nrows)

*This function parses the next start tag and finds the index of the element name in the descriptor table.*

### 5.1.1 Detailed Description

JSON low-level C encode/decode functions.

Definition in file [osrtjson.h](#).

# Index

JSON decode functions., 15

JSON encode functions., 4

osrtjson.h, 30

rtJsonDec

rtJsonDecAnyElem, 17

rtJsonDecAnyType, 17

rtJsonDecBase64Str, 19

rtJsonDecBool, 18

rtJsonDecDate, 18

rtJsonDecDateTime, 18

rtJsonDecDecimal, 19

rtJsonDecDouble, 19

rtJsonDecDynBase64Str, 19

rtJsonDecDynHexStr, 20

rtJsonDecGDay, 20

rtJsonDecGMonth, 20

rtJsonDecGMonthDay, 21

rtJsonDecGYear, 21

rtJsonDecGYearMonth, 21

rtJsonDecHexStr, 22

rtJsonDecInt16Value, 22

rtJsonDecInt32Value, 23

rtJsonDecInt64Value, 23

rtJsonDecInt8Value, 23

rtJsonDecMatchChar, 24

rtJsonDecMatchObjectStart, 24

rtJsonDecMatchToken, 24

rtJsonDecMatchToken2, 25

rtJsonDecNameValuePair, 25

rtJsonDecPeekChar, 25

rtJsonDecStringObject, 26

rtJsonDecStringValue, 26

rtJsonDecTime, 26

rtJsonDecUInt16Value, 27

rtJsonDecUInt32Value, 27

rtJsonDecUInt64Value, 27

rtJsonDecUInt8Value, 28

rtJsonDecXmlStringValue, 28

rtJsonGetElemIdx, 28

rtJsonDecAnyElem

rtJsonDec, 17

rtJsonDecAnyType

rtJsonDec, 17

rtJsonDecBase64Str

rtJsonDec, 17

rtJsonDecBool

rtJsonDec, 18

rtJsonDecDate

rtJsonDec, 18

rtJsonDecDateTime

rtJsonDec, 18

rtJsonDecDecimal

rtJsonDec, 19

rtJsonDecDouble

rtJsonDec, 19

rtJsonDecDynBase64Str

rtJsonDec, 19

rtJsonDecDynHexStr

rtJsonDec, 20

rtJsonDecGDay

rtJsonDec, 20

rtJsonDecGMonth

rtJsonDec, 20

rtJsonDecGMonthDay

rtJsonDec, 21

rtJsonDecGYear

rtJsonDec, 21

rtJsonDecGYearMonth

rtJsonDec, 21

rtJsonDecHexStr

rtJsonDec, 22

rtJsonDecInt16Value

rtJsonDec, 22

rtJsonDecInt32Value

rtJsonDec, 23

rtJsonDecInt64Value

rtJsonDec, 23

rtJsonDecInt8Value

rtJsonDec, 23

rtJsonDecMatchChar

rtJsonDec, 24

rtJsonDecMatchObjectStart

rtJsonDec, 24

rtJsonDecMatchToken

rtJsonDec, 24

rtJsonDecMatchToken2

rtJsonDec, 25

rtJsonDecNameValuePair

- rtJsonDec, [25](#)
- rtJsonDecPeekChar
  - rtJsonDec, [25](#)
- rtJsonDecStringObject
  - rtJsonDec, [26](#)
- rtJsonDecStringValue
  - rtJsonDec, [26](#)
- rtJsonDecTime
  - rtJsonDec, [26](#)
- rtJsonDecUInt16Value
  - rtJsonDec, [27](#)
- rtJsonDecUInt32Value
  - rtJsonDec, [27](#)
- rtJsonDecUInt64Value
  - rtJsonDec, [27](#)
- rtJsonDecUInt8Value
  - rtJsonDec, [28](#)
- rtJsonDecXmlStringValue
  - rtJsonDec, [28](#)
- rtJsonEnc
  - rtJsonEncAnyAttr, [6](#)
  - rtJsonEncBase64StrValue, [6](#)
  - rtJsonEncBoolValue, [6](#)
  - rtJsonEncDate, [7](#)
  - rtJsonEncDateTime, [7](#)
  - rtJsonEncDecimalValue, [7](#)
  - rtJsonEncDoubleValue, [7](#)
  - rtJsonEncFloatValue, [8](#)
  - rtJsonEncGDay, [8](#)
  - rtJsonEncGMonth, [8](#)
  - rtJsonEncGMonthDay, [9](#)
  - rtJsonEncGYear, [9](#)
  - rtJsonEncGYearMonth, [9](#)
  - rtJsonEncHexStr, [10](#)
  - rtJsonEncIndent, [10](#)
  - rtJsonEncInt64Value, [10](#)
  - rtJsonEncIntValue, [11](#)
  - rtJsonEncStringObject, [11](#)
  - rtJsonEncStringObject2, [11](#)
  - rtJsonEncStringPair, [12](#)
  - rtJsonEncStringPair2, [12](#)
  - rtJsonEncStringValue, [13](#)
  - rtJsonEncStringValue2, [13](#)
  - rtJsonEncTime, [13](#)
  - rtJsonEncUInt64Value, [14](#)
  - rtJsonEncUIntValue, [14](#)
  - rtJsonEncUnicodeData, [14](#)
- rtJsonEncAnyAttr
  - rtJsonEnc, [6](#)
- rtJsonEncBase64StrValue
  - rtJsonEnc, [6](#)
- rtJsonEncBoolValue
  - rtJsonEnc, [6](#)
- rtJsonEncDate
  - rtJsonEnc, [7](#)
- rtJsonEncDateTime
  - rtJsonEnc, [7](#)
- rtJsonEncDecimalValue
  - rtJsonEnc, [7](#)
- rtJsonEncDoubleValue
  - rtJsonEnc, [7](#)
- rtJsonEncFloatValue
  - rtJsonEnc, [8](#)
- rtJsonEncGDay
  - rtJsonEnc, [8](#)
- rtJsonEncGMonth
  - rtJsonEnc, [8](#)
- rtJsonEncGMonthDay
  - rtJsonEnc, [9](#)
- rtJsonEncGYear
  - rtJsonEnc, [9](#)
- rtJsonEncGYearMonth
  - rtJsonEnc, [9](#)
- rtJsonEncHexStr
  - rtJsonEnc, [10](#)
- rtJsonEncIndent
  - rtJsonEnc, [10](#)
- rtJsonEncInt64Value
  - rtJsonEnc, [10](#)
- rtJsonEncIntValue
  - rtJsonEnc, [11](#)
- rtJsonEncStringObject
  - rtJsonEnc, [11](#)
- rtJsonEncStringObject2
  - rtJsonEnc, [11](#)
- rtJsonEncStringPair
  - rtJsonEnc, [12](#)
- rtJsonEncStringPair2
  - rtJsonEnc, [12](#)
- rtJsonEncStringValue
  - rtJsonEnc, [13](#)
- rtJsonEncStringValue2
  - rtJsonEnc, [13](#)
- rtJsonEncTime
  - rtJsonEnc, [13](#)
- rtJsonEncUInt64Value
  - rtJsonEnc, [14](#)
- rtJsonEncUIntValue
  - rtJsonEnc, [14](#)
- rtJsonEncUnicodeData
  - rtJsonEnc, [14](#)
- rtJsonGetElemIdx
  - rtJsonDec, [28](#)