ー JSON機能補足 ー

NTTアドバンステクノロジ株式会社

はじめに

■ ユーザライブラリとして追加されたJSON機能の詳細動作を記載します。

・関連するユーザライブラリは下記のとおりです。

98_構造データ関連/01_JSON/

JSONファイル配列サイズ.ums7
JSONファイル配列読み取り.ums7
JSON変数ファイル保存.ums7
JSON変数新規オブジェクト.ums7
JSON変数新規配列.ums7
JSON変数要素追加.ums7
JSON変数読み取り.ums7
JSON変数配列サイズ.ums7
JSON変数配列要素追加.ums7
JSON変数配列読み取り.ums7
JSON形式書き込み.ums7
JSON形式読み取り.ums7

商標について

本書において以下に記載された名称、およびその他記載されている会社名、製品名は、各社の登録商標または商標です。なお、本文中ではTM、®、©マークは省略しています。

- WinActorはNTTアドバンステクノロジ株式会社の登録商標です。
- Microsoft、Windows※1、Microsoft Edge、Excel、VBScript※2は、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。

  ※1　Windowsの正式名称は、Microsoft Windows Operating Systemです。

  ※2　VBScriptの正式名称は、Microsoft Visual Basic Scripting Editionです。
- その他の記載されている会社名、製品名は各社の商標または登録商標です。

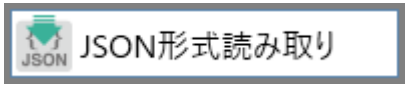
本書に関する注意

- 本書および提供するソフトウェア類に付された著作権表示「Copyright © 2013-2025 NTT, Inc. & NTT ADVANCED TECHNOLOGY CORPORATION」の変更、削除をすることはできません。
  本書の著作権はNTT株式会社及びNTTアドバンステクノロジ株式会社に帰属します。
- 本書では、Windowsの操作方法や機能を理解されていることを前提として説明しています。
  本書に記載されていないことについては、Microsoftが提供しているドキュメントなどをご覧ください。

JSON形式読み取り　読み取り結果(1/2)

JSONの各データ型が、「JSON形式読み取り」を使ってどのように読み取れるかを記載します。

「JSON形式読み取り」プロパティでは「JSON」と読み取り結果を格納する「変数」を指定します。
「型」はWinActorが自動判定します。



JSON形式読み取り　読み取り結果(2/2)

表．key の読み取り結果

No	JSON（※1）	型（※2）	読み取り結果（※3）
1	{ key : 123 }	整数値	123
2	{ key : 12.3 }	小数値	12.3
3	{ key : “123” }	文字列	“123”
4	{ key : “null” }	文字列	“null”
5	{ key : “” }	文字列	“”
6	{ key : { sub : 123 } }	オブジェクト	{ sub : 123 }
7	{ key : [1, 2, 3] }	配列	[1, 2, 3]
8	{ key : true }	真偽値	true
9	{ key : null }	Null値	null

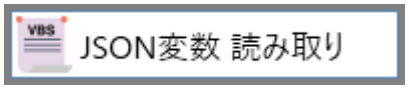
※1　読み取り対象のJSONです。
※2　JSONを読み取るときにWinActorが内部で自動判別する型です。判別結果は変数に保持されません。
※3　読み取り結果として指定した変数に格納される値です。

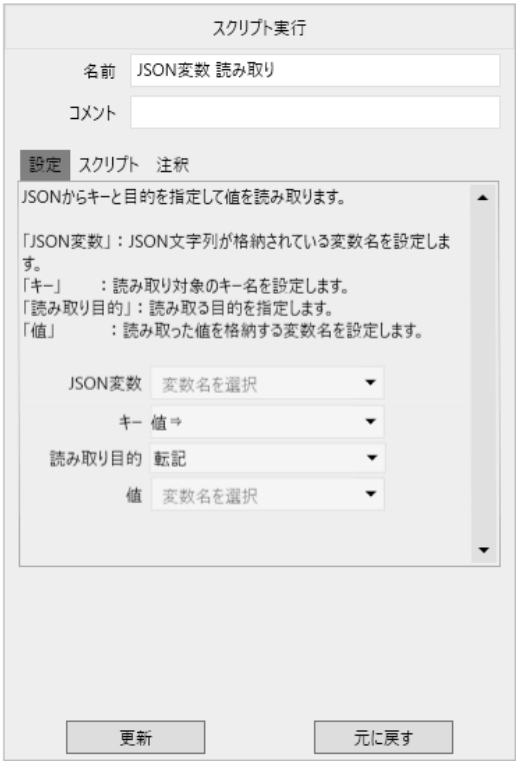
注意：
No3,No4 文字列は “” 付きで読み取られます。
No9 null値は null として読み取られます。

JSON変数読み取り　読み取り結果(1/4)

「JSON 変数読み取り」では、従来動作の**「値参照」**目的の読み取りに加え、**「転記」**目的の読み取り、**「型判定」**目的の読み取りが選択できるようになりました。

「JSON変数 読み取り」プロパティでは読み取り対象の「JSON」、
読み取り対象の「キー」、「読み取り目的」、
読み取り結果を格納する「変数」を指定します。





JSON変数読み取り　読み取り結果(2/4)

表．key の読み取り結果（値参照目的選択時）

No	JSON（※1）	型（※2）	読み取り結果（※3）
1	{ key : 123 }	整数値	123
2	{ key : 12.3 }	小数値	12.3
3	{ key : “123” }	文字列	“123”
4	{ key : “null” }	文字列	“null”
5	{ key : “” }	文字列	“”
6	{ key : { sub : 123 } }	オブジェクト	{ sub : 123 }
7	{ key : [1, 2, 3] }	配列	[1, 2, 3]
8	{ key : true }	真偽値	true
9	{ key : null }	Null値	null

※1　読み取り対象のJSONです。
※2　JSONを読み取るときにWinActorが内部で自動判別する型です。判別結果は変数に保持されません。
※3　読み取り結果として指定した変数に格納される値です。

注意：
No3,No4 文字列は ”” 付きで読み取られます。
No9 null値は null として読み取られます。

「値参照」目的の読み取りでは、
「JSON形式読み取り」の結果と同じになります。

JSON変数読み取り　読み取り結果(3/4)

表．key の読み取り結果（転記目的選択時）

No	JSON（※1）	型（※2）	読み取り結果（※3）
1	{ key : 123 }	整数値	123
2	{ key : 12.3 }	小数値	12.3
3	{ key : “123” }	文字列	123
4	{ key : “null” }	文字列	null
5	{ key : “” }	文字列	
6	{ key : { sub : 123 } }	オブジェクト	{ sub : 123 }
7	{ key : [1, 2, 3] }	配列	[1, 2, 3]
8	{ key : true }	真偽値	true
9	{ key : null }	Null値	

※1　読み取り対象のJSONです。
※2　JSONを読み取るときにWinActorが内部で自動判別する型です。判別結果は変数に保持されません。
※3　読み取り結果として指定した変数に格納される値です。

注意：
No3,No4,No5 文字列は ”” を取り除いて読み取られます。
No9 null値は 空文字として読み取られます。

「転記」目的の読み取りでは、
そのまま転記できるように加工した情報を読み取ります。

JSON変数読み取り　読み取り結果(4/4)

表．key の読み取り結果（型判定目的選択時）

No	JSON（※1）	型（※2）	読み取り結果（※3）
1	{ key : 123 }	整数値	INTEGER
2	{ key : 12.3 }	小数値	FLOAT
3	{ key : “123” }	文字列	STRING
4	{ key : “null” }	文字列	STRING
5	{ key : “” }	文字列	STRING
6	{ key : { sub : 123 } }	オブジェクト	OBJECT
7	{ key : [1, 2, 3] }	配列	ARRAY
8	{ key : true }	真偽値	BOOLEAN
9	{ key : null }	Null値	NULL

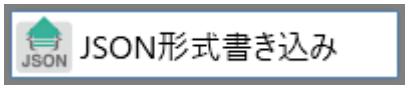
※1　読み取り対象のJSONです。
※2　JSONを読み取るときにWinActorが内部で自動判別する型です。
※3　読み取り結果として指定した変数に格納される値です。

「型判定」目的の読み取りでは、
型の自動判定結果を読み取ります。

JSON形式書き込み　書き込み結果（1/7）

JSONの各データ型が、「JSON形式書き込み」を使ってどのように書き込まれるかを記載します。

「JSON形式書き込み」プロパティでは「キー」と「型」と「値」を指定します。



JSON形式書き込み　書き込み結果(2/7)

表．key への書き込み結果(1/6)

No	型(※1)	値(※2)	書き込み結果(※3)

1	整数値	123	{
			"key" : 123
			}

2	整数値	12.3	エラー
3	整数値	null	エラー

4	整数値		{
			"key" : null
			}

5	小数値	123	{
			"key" : 123.0
			}

6	小数値	12.3	{
			"key" : 12.3
			}

7	小数値	null	エラー

8	小数値		{
			"key" : null
			}

※1　プロパティで指定する型です。
※2　プロパティで指定する値です。
※3　指定した変数もしくはファイルに書き込まれるJSONです。

注意：
No2 整数値指定で小数値を書き込もうとした場合はエラーとなります。
No5 小数値指定で整数値を書き込もうとした場合は小数値が書き込まれます。
No3,No4,No7,No8 null値を書き込む場合は、値を空にしてください。

JSON形式書き込み　書き込み結果(3/7)

表．key への書き込み結果(2/6)

No	型(※1)	値(※2)	書き込み結果(※3)

9	文字列	123	{
			"key" : "123"
			}

10	文字列	12.3	{
			"key" : "12.3"
			}

11	文字列	"123"	{
			"key" : "¥"123¥""
			}

12	文字列	null	{
			"key" : "null"
			}

13	文字列		{
			"key" : ""
			}

※1　プロパティで指定する型です。
※2　プロパティで指定する値です。
※3　指定した変数もしくはファイルに書き込まれるJSONです。

注意：
No11 WinActor側で外側の“”を付与します。値に””が含まれる場合は二重に””で括られる結果となります。
No12,No13 文字列型では null値を書き込むことができません。値にnullを指定した場合は文字列の”null”として扱われます。値を空にした場合は文字列の””として扱われます。
null値を書き込む場合は「Null値」型を利用してください。

JSON形式書き込み　書き込み結果(4/7)

表．key への書き込み結果(3/6)

No	型(※1)	値(※2)	書き込み結果(※3)

14	オブジェクト	{ }	{
			"key" : { }
			}

15	オブジェクト	{ sub : 123 }	{
			"key" : {
			"sub" : 123
			}
			}

16	オブジェクト	123	エラー

17	オブジェクト	null	{
			"key" : null
			}

18	オブジェクト		{
			"key" : null
			}

※1　プロパティで指定する型です。
※2　プロパティで指定する値です。
※3　指定した変数もしくはファイルに書き込まれるJSONです。

注意：
No16 オブジェクトを書き込む場合は{ } で括られた値を指定してください。
No17,No18 null値を書き込む場合は、値を空にしてください。No17でもnull値を書き込むことができますが、他の型での使い方と合わせてNo18の使い方を推奨いたします。

JSON形式書き込み　書き込み結果(5/7)

表．key への書き込み結果(4/6)

No	型(※1)	値(※2)	書き込み結果(※3)

19	配列	[ ]	{
			"key" : [ ]
			}

20	配列	[1,2,3]	{
			"key" : [1, 2, 3]
			}

21	配列	123	エラー

22	配列	null	{
			"key" : null
			}

23	配列		{
			"key" : null
			}

※1　プロパティで指定する型です。
※2　プロパティで指定する値です。
※3　指定した変数もしくはファイルに書き込まれるJSONです。

注意：
No21 配列を書き込む場合は[ ] で括られた値を指定してください。
No22,No23 null値を書き込む場合は、値を空にしてください。No22でもnull値を書き込むことができますが、他の型での使い方と合わせてNo23の使い方を推奨いたします。

JSON形式書き込み　書き込み結果(6/7)

表．key への書き込み結果(5/6)

No	型(※1)	値(※2)	書き込み結果(※3)
24	真偽値	true	{ "key" : true }
25	真偽値	false	{ "key" : false }
26	真偽値	123	{ "key" : false }
27	真偽値	0	{ "key" : false }
28	真偽値	-1	{ "key" : false }
29	真偽値	abc	{ "key" : false }
30	真偽値	null	{ "key" : false }
31	真偽値		{ "key" : null }

※1　プロパティで指定する型です。
※2　プロパティで指定する値です。
※3　指定した変数もしくはファイルに書き込まれるJSONです。

注意：
No26,No27,No28,No29 エラーにはならずfalseが格納されます。意図しないJSONの出力となる可能性があるため、真偽値を書き込む場合は true もしくは false の値を指定してください。
No30,No31 null値を書き込む場合は、値を空にしてください。

JSON形式書き込み　書き込み結果（7/7）

表．key への書き込み結果（6/6）

No	型（※1）	値（※2）	書き込み結果（※3）

32	Null値	123	{
			"key" : "123"
			}

33	Null値	12.3	{
			"key" : "12.3"
			}

34	Null値	"123"	{
			"key" : "¥"123¥""
			}

35	Null値	null	{
			"key" : "null"
			}

36	Null値		{
			"key" : null
			}

※1　プロパティで指定する型です。
※2　プロパティで指定する値です。
※3　指定した変数もしくはファイルに書き込まれるJSONです。

注意：
No32,No33,No34「Null値」の型は、基本的には「文字列」の型を指定した時と同じ書き込み結果となります。
No35,No36 空の値を指定した時の動作が「文字列」の型と異なります。
null値を書き込む場合は、値を空にしてください。値にnullを指定した場合は文字列の”null”として扱われます。

WinActor® JSON機能補足

NTTアドバンステクノロジ株式会社

WA7-C-20250603