

NAME

json – gawk extension to import from and export to JSON format

SYNOPSIS

```
@load "json"

status = json::from_json(string, array)

encoded = json::to_json(array [, use_real_array])

# compatibility functions in json_compat.awk:
@include "json_compat"

status = json_fromJSON(string, array)

encoded = json_toJSON(array [, use_real_array])
```

DESCRIPTION

JSON objects represent associations of names with values, what we will call *linear* arrays (arrays with numeric indices), and subgroupings (nested associative arrays). Such objects map well onto *gawk*'s true multidimensional associative arrays. This extension provides a mechanism to encode a *gawk* array as a JSON string and to decode a JSON object into a *gawk* array.

The *json* extension adds two functions as follows:

json::from_json(string, array)

This function takes a *string* representing a JSON object and decodes it into the *array* argument. The array is cleared first. The return status is one upon success or zero upon failure. In addition, *gawk*'s **ERRNO** variable is updated to (sort of) indicate what the problem was.

json::to_json(array[, use_real_array])

This function takes an *array* argument and encodes the array contents as a JSON object in the returned string. The returned string will be empty if an error of some kind occurred. In addition, *gawk*'s **ERRNO** variable is updated to (sort of) indicate what the problem was.

If the optional parameter *use_real_array* is not zero, **json::to_json()** encodes arrays indexed from 1 to *N* as JSON linear arrays, instead of as associative arrays. This gives a better rendition into JSON, at the expense of some additional CPU time to verify that the array is indeed indexed linearly.

BUGS

The mapping between *gawk* objects and JSON objects isn't perfect. In particular, JSON boolean objects map to 1 or 0, losing the type distinction. Typed regular expression constants in *gawk* are encoded as a string, with the regexp bracketed by @/ and /. For example, **pattern:"@[a-z]+[0-9]?/"**.

The **json_compat.awk** library file provides compatibility functions for the previous version of this extension, which did not take advantage of *gawk*'s namespace facility.

EXAMPLE

```
@load "filefuncs"
@load "json"
...
BEGIN {
    stat("/etc/passwd", data)
    statinfo = json::to_json(data)
    print statinfo
}
```

```
BEGIN {
    getline statinfo
    if (! json::from_json(statinfo, data))
        print "JSON import failed!" > "/dev/stderr"
        exit 1
    }

    # do something with the info
}
```

NOTES

This extension uses RapidJSON to parse a JSON document into a *gawk* array. This approximates the Document Object Model (DOM) paradigm. In the author's humble opinion, the SAX paradigm for parsing JSON does not match well with **awk**'s design. If you want SAX parsing of JSON, this isn't the place.

SEE ALSO

JSON home page: <https://www.json.org/>
Wikipedia: <https://en.wikipedia.org/wiki/JSON>
RapidJSON home page: <http://rapidjson.org/>

AUTHOR

Arnold D. Robbins

COPYING PERMISSIONS

Copyright © 2017, the Free Software Foundation, Inc.
Copyright © 2017, 2020, Arnold David Robbins.

Permission is granted to make and distribute verbatim copies of this manual page provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual page under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual page into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Foundation.