



**Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!**

***Rocky Enterprise Linux 9.2 Manual Pages on command 'json\_pp.1'***

**\$ man json\_pp.1**

JSON\_PP(1) Perl Programmers Reference Guide JSON\_PP(1)

NAME

json\_pp - JSON::PP command utility

SYNOPSIS

json\_pp [-v] [-f from\_format] [-t to\_format] [-json\_opt options\_to\_json1[,options\_to\_json2[,...]]]

DESCRIPTION

json\_pp converts between some input and output formats (one of them is JSON). This program was copied from json\_xs and modified.

The default input format is json and the default output format is json with pretty option.

OPTIONS

-f

-f from\_format

Reads a data in the given format from STDIN.

Format types:

json

as JSON

eval

as Perl code

-t

Writes a data in the given format to STDOUT.

null

no action.

json

as JSON

dumper

as Data::Dumper

-json\_opt

options to JSON::PP

Acceptable options are:

ascii latin1 utf8 pretty indent space\_before space\_after relaxed canonical allow\_nonref  
allow\_singlequote allow\_barekey allow\_bignum loose escape\_slash indent\_length

Multiple options must be separated by commas:

Right: -json\_opt pretty,canonical

Wrong: -json\_opt pretty -json\_opt canonical

-v

Verbose option, but currently no action in fact.

-V

Prints version and exits.

## EXAMPLES

```
$ perl -e'print q[{"foo":"??","bar":12345678900000000000000000}]' | \
```

```
json_pp -f json -t dumper -json_opt pretty,utf8,allow_bignum
```

```
$VAR1 = {
```

```
    'bar' => bless( {
```

```
        'value' => [
```

```
            '0000000',
```

```
            '0000000',
```

```
            '5678900',
```

```
            '1234'
```

```
        ],
```

```
        'sign' => '+'
```

```
    }, 'Math::BigInt' ),
```

```
    'foo' => "\x{3042}\x{3044}"
```

```
};
```

```
$ perl -e'print q[{"foo":"??","bar":12345678900000000000000000}]' | \
```

```
json_pp -f json -t dumper -json_opt pretty
```

```
$VAR1 = {
```

```
'bar' => '12345678900000000000000000',  
'foo' => "\x{e3}\x{81}\x{82}\x{e3}\x{81}\x{84}"  
};
```

#### SEE ALSO

JSON::PP, json\_xs

#### AUTHOR

Makamaka Hannyaharamitu, <makamaka[at]cpan.org>

#### COPYRIGHT AND LICENSE

Copyright 2010 by Makamaka Hannyaharamitu

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

perl v5.34.0

2023-11-23

JSON\_PP(1)