SAS® 9.4 JSON Procedure Tip Sheet

Exporting a JSON File Using Default Options

This example exports data from Sashelp.Class. All PROC JSON default options are enabled.

The output begins with an open JSON object container ({) as the top-level container. Following the top-level container, the SAS export version and exported SAS data set name is the default SAS metadata. A JSON array container is opened ([) to collect the observations as a list of values. Each observation is then exported as a nested JSON object ({ }). PROC JSON writes the data as a single line. Each observation consists of key-value pairs for the variable names and values. Trailing blanks are removed from the end of character data. The output file ends by closing the array container (]) and the top-level container.

```
proc json out='C:\DefaultOutput.json';
   export sashelp.class (where=(age=12));
run;
```

```
{"SASJSONExport":"1.0","SASTableData+CLASS":[{"Name":"James","Sex":"M","Age":12,"Height":57.3,"Weight":83},{"Name":"Jane","Sex":"F","Age":12,"Height":59.8,"Weight":84.5},{"Name":"John","Sex":"M","Age":12,"Height":59,"Weight":99.5},{"Name":"Louise","Sex":"F","Age":12,"Height":56.3,"Weight":77},{"Name":"Robert","Sex":"M","Age":12,"Height":64.8,"Weight":128}]}
```

Note: The actual JSON output file is a single line of text.

Controlling JSON Output With Options

This example exports data from Sashelp. Class and uses options to control the JSON output.

1 The PRETTY option creates a more readable format by using indention to illustrate the JSON container structure. **2** The WHERE= data set option specifies conditions to select observations from the SAS data set. **3** The NOKEYS option removes SAS variable names, and by including NOKEYS in the EXPORT statement, each selected observation is exported as a list of values in an array container. **4** The NOSASTAGS option removes SAS metadata, and by including NOSASTAGS in the EXPORT statement, the top-level container is a JSON array.

```
proc json out='C:\ControlOutput.json'
      pretty; 1
   export sashelp.class (where=(age=12)) 2
      / nokeys 3 nosastags; 4
run;
```

```
[
  [
    "James",
    "M",
    12,
    57.3,
    83
  ],
  [
    "Jane",
    "F",
    12,
    59.8,
    84.5
  ],
  [
    "John",
    "M",
    12,
    59,
    99.5
  ],
  [
    "Louise",
    "F",
    12,
    56.3,
    77
  ],
  [
    "Robert",
    "M",
    12,
    64.8,
    128
  ]
]
```

JSON Containers

JSON object container {…}
begins with a left brace ({) and ends with a right brace (}). An object container collects key-value pairs that are written as pairs of names and values. A value can be any of the supported JSON data types, an object, or an array. Each key (name) is followed by a colon and then the value. The key-value pairs are separated by a comma.

JSON array container […]
begins with a left bracket ([) and ends with a right bracket (]). An array container collects a list of values that are written as a list of values without names. A value can be any of the supported JSON data types, an object, or an array. Values are separated by a comma.

Container Tips

- The top-level container can include any number of containers. Each container can then nest containers to an arbitrary depth.
- Objects require a list of key-value pairs, where the value can be JSON data types, an object, or an array.
- Arrays are merely a list of values, objects, or arrays.
- WRITE VALUES statement(s) for an object must result in an even number of values, and the key portion of the key-value pair must be a string.

For complete information, refer to the SAS 9.4 documentation at support.sas.com/documentation.

SAS Institute Inc. World Headquarters
+1 919 677 8000 www.sas.com/offices

SAS® 9.4 JSON Procedure Tip Sheet

This tip sheet places frequently used information in one place, on one sheet of paper, so you don't have to search throughout the documentation. This tip sheet presents SAS 9.4 information for the JSON procedure.

- PROC JSON reads data from a SAS data set and writes it to an external file in JSON[1] representation.
- You can control the exported data with several options that remove content and affect the format.
- PROC JSON provides statements that enable you to write additional data to the external file and control JSON containers.

[1] JavaScript Object Notation (JSON) is a text-based, open standard data format that is designed for human-readable data interchange. JSON is based on a subset of the JavaScript programming language and uses JavaScript syntax for describing data objects.

SAS® 9.4 JSON Procedure Tip Sheet

PROC JSON Syntax

PROC JSON OUT=*fileref* | '*external-file*'
<option(s)>;

EXPORT *<libref.>SAS-data-set*
<(SAS-data-set-option(s))> </option(s)>;

WRITE VALUES *value(s) </option(s)>*;

WRITE OPEN *type*;

WRITE CLOSE;

Argument Description

OUT=*fileref* | '*external-file*'
identifies the JSON output file with either a SAS fileref that is assigned with the FILENAME statement or the physical location of the JSON output file.

<option(s)>
specifies statement options that control the resulting JSON output. (Statement options are listed in the next panel.)

<libref.>SAS-data-set
identifies the SAS data set to be exported with either a one- or two-level SAS name (library and member name).

<(SAS-data-set-option(s))>
specifies SAS data set options that apply to the input SAS data set.

value(s)
specifies one or more values to write to the JSON output file. Separate values with a blank space. A value can be a string, a number, the Boolean value TRUE or FALSE, or NULL.

type
specifies the type of JSON container to explicitly open as either ARRAY or OBJECT.

Statement Options

FMTCHARACTER | **NOFMTCHARACTER** [1 2]
determines whether to apply an associated character SAS format.

FMTDATETIME | NOFMTDATETIME [1 2]
determines whether to apply an associated date, time, or datetime SAS format.

FMTNUMERIC | NOFMTNUMERIC [1 2]
determines whether to apply an associated numeric SAS format (excluding date, time, and datetime SAS formats).

KEYS | NOKEYS [1 2]
determines whether to include or suppress SAS variable names as keys for values.

PRETTY | **NOPRETTY** [1]
determines how to format the JSON output. PRETTY uses indention. NOPRETTY writes output as a single line of text.

SASTAGS | NOSASTAGS [1 2]
determines whether to include or suppress top-level tags, the object top-level container, and the export metadata.

SCAN | NOSCAN [1 2 3]
determines whether PROC JSON scans and encodes input strings.

TABLENAME='*name*' [2]
specifies an alternate suffix for the SASTableData top-level tag.

TRIMBLANKS | NOTRIMBLANKS [1 2 3]
determines whether to remove or retain trailing blanks in character string values.

- Option supported in:
  [1] PROC JSON statement
  [2] EXPORT statement
  [3] WRITE VALUES statement
- Default option keywords appear in bold.

Creating and Controlling Containers

As the first statement after the PROC JSON statement:

- EXPORT with default options enabled implicitly opens an object top-level container, implicitly opens an array container to collect observations as a list of values, and then exports each observation as a nested object.
  {metadata [{obs1},{obs2}…]}
- EXPORT with NOKEYS enabled implicitly opens an object top- level container, implicitly opens an array container to collect observations as a list of values, and then exports each observation as a nested array.
  {metadata [[obs1],[obs2]…]}
- EXPORT with NOSASTAGS enabled implicitly opens an array top-level container, and then exports each observation as a nested object.
  [{obs1},{obs2}…]
- EXPORT with NOSASTAGS and NOKEYS enabled implicitly opens an array top-level container, and then exports each observation as a nested array.
  [[obs1],[obs2]…]
- WRITE VALUES implicitly opens an object top-level container.

WRITE OPEN OBJECT explicitly opens an object, which you must explicitly close with WRITE CLOSE.

WRITE OPEN ARRAY explicitly opens an array, which you must explicitly close with WRITE CLOSE.

Exporting Data, Writing Arbitrary JSON Output, and Creating Containers

This example exports a subset of Sashelp.Class, writes additional values to the JSON output file, and controls and nests JSON containers.

1 WRITE VALUES implicitly opens an object top-level container and writes a value to the top-level container. **2** WRITE OPEN ARRAY explicitly opens an array container. **3** WRITE VALUES writes a value to the open array container. **4** WRITE OPEN OBJECT explicitly opens an object container that is nested in the array container. **5** EXPORT exports a subset of the Sashelp.Class data set to the open object container, implicitly opens an array container to collect the observations as a list of values, and then nests the exported observation as an object. **6** WRITE CLOSE explicitly closes the open object container. **7** WRITE CLOSE explicitly closes the open array container. The object top-level container is implicitly closed.

```
proc json out='C:\Arbitrary.json' pretty;
   write values 'Age 11'; 1
   write open array; 2
   write values 'Girls'; 3
   write open object; 4
   export sashelp.class(where=((sex='F')
      and (age=11))); 5
   write close; 6
   write close; 7
run;
```

```
{
  "Age 11": [
    "Girls",
    {
      "SASJSONExport": "1.0 PRETTY",
      "SASTableData+CLASS": [
        {
          "Name": "Joyce",
          "Sex": "F",
          "Age": 11,
          "Height": 51.3,
          "Weight": 50.5
        }
      ]
    }
  ]
}
```