

# Diffable JSON Strings

If you need a JSON object/string to be consistently ordered regardless of how it was built or ordered in memory, the following script can help force it to be organized/ordered alphabetically so that it can be used with source control tools or use diff tools to compare different outputs to one another.

```
# Functions designed to be called via objectScript() in UI expressions.

from java.util import TreeMap
from com.inductiveautomation.ignition.common.util import Comparators
from com.inductiveautomation.ignition.common.gson import GsonBuilder

ciAlnumCmp = Comparators.alphaNumeric(False)

def __ordering(subject, listKey='name'):
    '''
    []Deep copy with conversion of maps to key-ordered maps and
    []conversion of lists-of-dicts that contain a 'name' key into
    []ordered lists.
    []Keys other than "name" may be supplied, or None to disable
    []re-ordering lists of dictionaries.
    []'''
    []if hasattr(subject, 'items'):
    []    []subst = TreeMap(ciAlnumCmp)
    []    []for k, v in subject.items():
    []        []subst[k] = __ordering(v)
    []    []return subst
    []if hasattr(subject, '__iter__'):
    []    []# Use a generator to exit quickly if any element of the
    []    []# list-like object is *not* a dictionary-like object.
    []    []if listKey and all(hasattr(inner, 'items') for inner in subject):
    []        []reordered = TreeMap(ciAlnumCmp)
    []        []reordered.update([(x.get(listKey, str(i)), __ordering(x)) for (i, x) in enumerate(subject)])
    []        []return reordered.values()
    []    []return [__ordering(x) for x in subject]
    []return subject
[]

def orderedJson(json, useGson = True):
```

```
'''
```

```
    Re-orders a JSON string alphabetically so it can be use with diff tools or other source control tools
```

```
Args:
```

```
    json (str): JSON Dictionary as a string to be re-ordered
```

```
    useGson (bool): Use Google's Gson library to pretty-print the JSON
```

```
Returns:
```

```
    str: Re-ordered JSON string
```

```
    ...
```

```
source = system.util.jsonDecode(json)
```

```
ordered = __ordering(source)
```

```
if useGson:
```

```
    gson = GsonBuilder().setPrettyPrinting().create()
```

```
    return gson.toJson(ordered)
```

```
else:
```

```
    return system.util.jsonEncode(dict(_=ordered), 2)[6:-1]
```

Source slightly tweaked from: <https://forum.inductiveautomation.com/t/json-diffing-discussion/96800/26>

---

Revision #4

Created 2024-10-03 21:53:20 UTC by Michael Flagler

Updated 2024-10-04 18:37:49 UTC by Michael Flagler