Handling JSON data
Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

WAGO is a registered trademark of WAGO Verwaltungsgesellschaft mbH.
Table of Contents

1  Notes about this Documentation .................................................................4
1.1 Copyright .................................................................................................4
1.1 Symbols .....................................................................................................4
1.2 Number Notation .......................................................................................6
1.3 Font Conventions ......................................................................................6
1.4 Legal Bases ................................................................................................7
1.4.1 Subject to Changes ................................................................................7
1.4.2 Personal Qualifications .........................................................................7
1.4.3 Limitation of Liability ..........................................................................7
1.5 Revision History ........................................................................................8

2  Description ..................................................................................................9

3  Material Used ..............................................................................................9
3.1 Required Libraries .....................................................................................9
3.2 Devices .....................................................................................................10
3.3 Tools ........................................................................................................10

4  Write a simple JSON String .........................................................................11

5  Access data within a JSON String ..............................................................12

6  Importing JSON string in IEC variable ......................................................14

List of Figures ................................................................................................17
List of Tables ..................................................................................................18
Notes about this Documentation

1.1 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

1.1 Symbols

DANGER

Personal Injury!
Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

DANGER

Personal Injury Caused by Electric Current!
Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

WARNING

Personal Injury!
Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

CAUTION

Personal Injury!
Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

NOTICE

Damage to Property!
Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.
**NOTICE**

Damage to Property Caused by Electrostatic Discharge (ESD)!
Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

---

**Note**

Important Note!
Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.

---

**Information**

Additional Information:
Refers to additional information which is not an integral part of this documentation (e.g., the Internet).
1.2 Number Notation

Table 1: Number Notation

<table>
<thead>
<tr>
<th>Number Code</th>
<th>Example</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal</td>
<td>100</td>
<td>Normal notation</td>
</tr>
<tr>
<td>Hexadecimal</td>
<td>0x64</td>
<td>C notation</td>
</tr>
<tr>
<td>Binary</td>
<td>'100'</td>
<td>In quotation marks, nibble separated with dots(.)</td>
</tr>
<tr>
<td></td>
<td>'0110.0100'</td>
<td></td>
</tr>
</tbody>
</table>

1.3 Font Conventions

Table 2: Font Conventions

<table>
<thead>
<tr>
<th>Font Type</th>
<th>Indicates</th>
</tr>
</thead>
<tbody>
<tr>
<td>italic</td>
<td>Names of paths and data files are marked in italic-type.</td>
</tr>
<tr>
<td></td>
<td>e.g.: \texttt{C:\Program Files\WAGO Software}</td>
</tr>
<tr>
<td>Menu</td>
<td>Menu items are marked in bold letters.</td>
</tr>
<tr>
<td></td>
<td>e.g.: \texttt{Save}</td>
</tr>
<tr>
<td>&gt;</td>
<td>A greater-than sign between two names means the selection of a menu item from a menu.</td>
</tr>
<tr>
<td></td>
<td>e.g.: \texttt{File &gt; New}</td>
</tr>
<tr>
<td>Input</td>
<td>Designation of input or optional fields are marked in bold letters.</td>
</tr>
<tr>
<td></td>
<td>e.g.: \texttt{Start of measurement range}</td>
</tr>
<tr>
<td>&quot;Value&quot;</td>
<td>Input or selective values are marked in inverted commas.</td>
</tr>
<tr>
<td></td>
<td>e.g.: Enter the value &quot;4 mA&quot; under \texttt{Start of measurement range}.</td>
</tr>
<tr>
<td>[Button]</td>
<td>Pushbuttons in dialog boxes are marked with bold letters in square brackets.</td>
</tr>
<tr>
<td></td>
<td>e.g.: [Input]</td>
</tr>
<tr>
<td>[Key]</td>
<td>Keys are marked with bold letters in square brackets.</td>
</tr>
<tr>
<td></td>
<td>e.g.: [F5]</td>
</tr>
</tbody>
</table>
1.4 **Legal Bases**

1.4.1 **Subject to Changes**

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

1.4.2 **Personal Qualifications**

The use of the product described in this document is exclusively geared to specialists having qualifications in PLC programming, electrical specialists or persons instructed by electrical specialists who are also familiar with the appropriate current standards.

Moreover, the persons cited here must also be familiar with all of the products cited in this document, along with the operating instructions. They must also be capable of correctly predicting any hazards which may not arise until the products are combined.

WAGO Kontakttechnik GmbH & Co. KG assumes no liability resulting from improper action and damage to WAGO products and third-party products due to non-observance of the information contained in this document.

1.4.3 **Limitation of Liability**

This documentation describes the use of various hardware and software components in specific example applications. The components may represent products or parts of products from different manufacturers. The respective operating instructions from the manufacturers apply exclusively with regard to intended and safe use of the products. The manufacturers of the respective products are solely responsible for the contents of these instructions.

The sample applications described in this documentation represent concepts, that is, technically feasible application. Whether these concepts can actually be implemented depends on various boundary conditions. For example, different versions of the hardware or software components can require different handling than that described here. Therefore, the descriptions contained in this documentation do not form the basis for assertion of a certain product characteristic.

Responsibility for safe use of a specific software or hardware configuration lies with the party that produces or operates the configuration. This also applies when one of the concepts described in this document was used for implementation of the configuration.

WAGO Kontakttechnik GmbH & Co. KG is not liable for any actual implementation of the concepts.
1.5 Revision History

Table 3: Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Author</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0</td>
<td>u010663</td>
<td>First issue</td>
</tr>
</tbody>
</table>
2 Description

**NOTICE**

Installation of Sample Projects for e!COCKPIT

![Sample Projects](image)

Figure 1: e!COCKPIT Sample Projects

Sample programs can be called up from the e!COCKPIT Backstage view by clicking the Updates & Add-ons button in the navigation bar.

This application note describes how to handle data in JSON format.

This application note focuses on the included "WagoAppJSON_1.5.x.x_Example_01_Basic" project. Each function block is explained in an own "PLC_PRG_xxx" program.

3 Material Used

3.1 Required Libraries

<table>
<thead>
<tr>
<th>Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WagoAppJSON</td>
<td>Function blocks handling JSON data.</td>
</tr>
</tbody>
</table>
3.2 Devices

Table 5: Devices

<table>
<thead>
<tr>
<th>Supplier</th>
<th>Quantity</th>
<th>Description</th>
<th>Item No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAGO</td>
<td>1</td>
<td>PFC200</td>
<td>750-8202</td>
</tr>
</tbody>
</table>

3.3 Tools

Table 6: Tools

<table>
<thead>
<tr>
<th>Description</th>
<th>Item No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>e!COCKPIT workstation license</td>
<td>2759-101/1110-2002</td>
</tr>
</tbody>
</table>
4 Write a simple JSON String

In case of writing a more or less static JSON string, with only some values having different values throughout the time, using function block "Fb_JSON_Write_01" may be useful.

```json
{
  "Menu":
    {
      "id": "#Parameter",
      "topic": "#Parameter",
      "popup": {
        "menuitem": [
          {
            "value": "#Parameter",
            "description": "#Parameter"
          }
        ]
      }
    }
}
```

Figure 1: JSON-Data

The parameter must be provided by an array containing the actual values.

```
<table>
<thead>
<tr>
<th>MyJSON_Data</th>
<th>ARRAY [0..7] OF STRING</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0]</td>
<td>STRING</td>
</tr>
<tr>
<td>[1]</td>
<td>STRING</td>
</tr>
<tr>
<td>[2]</td>
<td>STRING</td>
</tr>
<tr>
<td>[3]</td>
<td>STRING</td>
</tr>
<tr>
<td>[4]</td>
<td>STRING</td>
</tr>
<tr>
<td>[5]</td>
<td>STRING</td>
</tr>
<tr>
<td>[6]</td>
<td>STRING</td>
</tr>
</tbody>
</table>
```

Figure 3: Parameter array of variable data for "MyJSON_Data"
5 Access data within a JSON String

In case data is formatted in JSON style, access to single values is done by function block “Fb_JSON_ParseAndModify”.

Figure 4: Block “Fb_JSON_ParseAndModify”

Step one is to read the JSON data by executing variable “xTrigger”.

```json
{
  "1": { "type": "digital", "objid": 5, "value": 43164 },
  "2": { "type": "digital", "objid": 3, "value": 115136 },
  "3": { "type": "analog", "objid": 8, "value": 34.8 },
  "4": { "type": "complex", "objid": 9, "value": 0.5 }
}
```

Figure 5: JSON-String

Once this action is successful finished output “diToken” will show the number of detected token.

(A token is only an internal organization counter and therefore only a control number with no additional user relevant information.)

Step 2 allows access to the JSON data by a JSON pointer using method “GetValueByPath”.

If you want to get the value from the element “4”, the JSON pointer would be: `/4/value`
Access data within a JSON String

Figure 6: Example with JSON pointer /4/value
6 Importing JSON string in IEC variable

In case of a large amount of JSON data it may be more easy using “FbWrite_ToIEC_ByRule”.

It is assumed, that an IEC variable is defined, which should contain all information from the JSON string.

IEC variable:

<table>
<thead>
<tr>
<th>MyJsonData[0]</th>
<th>typCustomData</th>
</tr>
</thead>
<tbody>
<tr>
<td>tmst</td>
<td>UINT</td>
</tr>
<tr>
<td>chan</td>
<td>BYTE</td>
</tr>
<tr>
<td>rfch</td>
<td>BYTE</td>
</tr>
<tr>
<td>freq</td>
<td>UINT</td>
</tr>
<tr>
<td>stat</td>
<td>BYTE</td>
</tr>
<tr>
<td>modu</td>
<td>STRING</td>
</tr>
<tr>
<td>datr</td>
<td>STRING</td>
</tr>
<tr>
<td>payload</td>
<td>STRING</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MyJsonData[1]</th>
<th>typCustomData</th>
</tr>
</thead>
<tbody>
<tr>
<td>tmst</td>
<td>UINT</td>
</tr>
<tr>
<td>chan</td>
<td>BYTE</td>
</tr>
<tr>
<td>rfch</td>
<td>BYTE</td>
</tr>
<tr>
<td>freq</td>
<td>UINT</td>
</tr>
<tr>
<td>stat</td>
<td>BYTE</td>
</tr>
<tr>
<td>modu</td>
<td>STRING</td>
</tr>
<tr>
<td>datr</td>
<td>STRING</td>
</tr>
<tr>
<td>payload</td>
<td>STRING</td>
</tr>
</tbody>
</table>

Figure 7: IEC variable containing the information from the JSON string
The JSON data:

```json
{  
  "Data_Objects":[
    {  
      "txmsg": 1519935596,  
      "chan": 5,  
      "fch": 1,  
      "Ireq": 869525,  
      "stat": 1,  
      "modu": "LORA",  
      "datz": "SF6BW12S",  
      "payload":"
    },
    {  
      "txmsg": 1519935597,  
      "chan": 6,  
      "fch": 2,  
      "Ireq": 869544,  
      "stat": 2,  
      "modu": "LORAIR",  
      "datz": "SF6BW1",  
      "payload": "110a000f00551001"
    }
  ]
}
```

Figure 8: Structure of the JSON file

---

**Note**

**Important Note!**

Attribute pack mode must be set in the data type definition

```
{attribute 'pack_mode' := '0'}
```

---

The name of the JSON object containing the array data is placed at input "sPathBase".

The structure of the data is defined in an array. This array contains information about the name and the data type.
Importing JSON string in IEC variable

```
bTestRule : ARRAY[0..MAX_ARRAY_ELEMENTS_CUSTOM] OF typRuleBase:=
  (sPath:='tmst', iConvertType:=eUInt),
  (sPath:='chan', iConvertType:=eByte),
  (sPath:='rfch', iConvertType:=eByte),
  (sPath:='frq', iConvertType:=eUInt),
  (sPath:='stat', iConvertType:=eByte),
  (sPath:='modu', iConvertType:=eString),
  (sPath:='datr', iConvertType:=eString),
  (sPath:='payload', iConvertType:=eString)
```

Figure 10: Example of a Rule

---

**Note**

**Important Note!**

The value “MAX_ARRAY_ELEMENTS_CUSTOM” must be set exactly one less than the number of elements.

```
MAX_ARRAY_ELEMENTS_CUSTOM:=7
```

Figure 11: ParameterList containing "MAX_ARRAY_ELEMENTS_CUSTOM"
List of Figures

Figure 1: JSON-Data.................................................................11
Figure 2: Function block "Fb_JSON_Writer_01".................................11
Figure 3: Parameter array of variable data for "MyJSON_Data"..............11
Figure 4: Block "Fb_JSON_ParseAndModify" ..................................12
Figure 5: JSON-String................................................................12
Figure 6: Example with JSON pointer /4/value ..................................13
Figure 7: IEC variable containing the information from the JSON string ....14
Figure 8: Structure of the JSON file................................................15
Figure 9: Function block "FbWrite_ToIEC_ByRule"..............................15
Figure 10: Example of a Rule.........................................................16
Figure 11: ParameterList containing "MAX_ARRAY_ELEMENTS_CUSTOM" ....16
List of Tables

Table 1: Number Notation ................................................................. 6
Table 2: Font Conventions ............................................................... 6
Table 3: Revision History ............................................................... 8
Table 4: Required Libraries ............................................................ 9
Table 5: Devices ............................................................................ 10
Table 6: Tools .............................................................................. 10