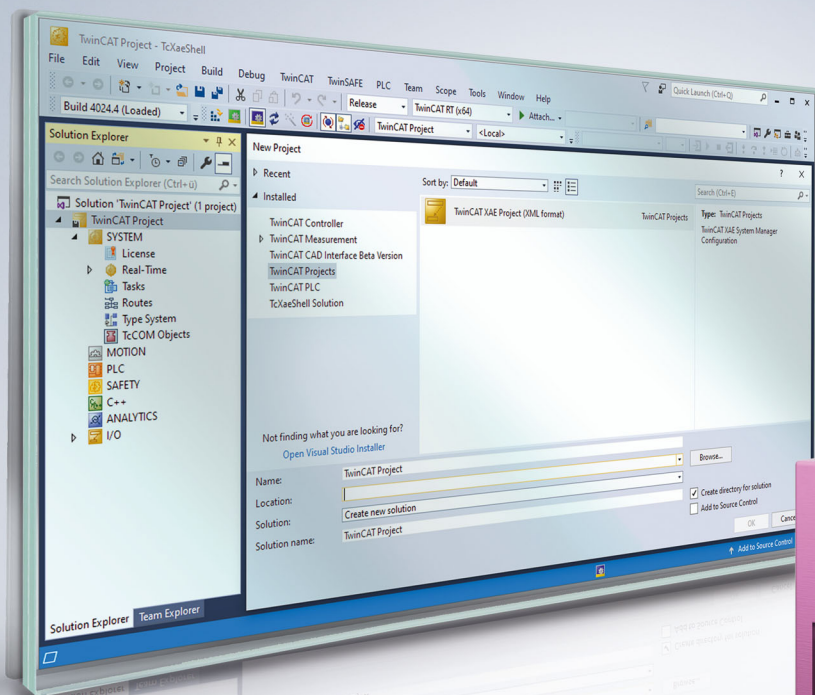


# BECKHOFF New Automation Technology

Manual | EN

# TF6020

## TwinCAT 3 | JSON Data Interface





# Table of contents

<b>1 Foreword</b> .....	<b>5</b>
1.1 Notes on the documentation .....	5
1.2 Safety instructions .....	6
1.3 Notes on information security.....	7
<b>2 Overview</b> .....	<b>8</b>
<b>3 Installation</b> .....	<b>9</b>
3.1 System requirements .....	9
3.2 Installation .....	9
3.3 Licensing .....	9
<b>4 Technical introduction</b> .....	<b>12</b>
4.1 API description .....	12
4.2 Access via ADS.....	12
4.3 Access via MQTT .....	13
4.3.1 Topic structure .....	14
<b>5 Examples</b> .....	<b>15</b>
<b>6 Appendix</b> .....	<b>16</b>
6.1 Error diagnosis .....	16



# 1 Foreword

## 1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.

It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.

It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

### Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without prior announcement. No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

### Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.

Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

### Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702  
with corresponding applications or registrations in various other countries.



EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

### Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

## 1.2 Safety instructions

### Safety regulations

Please note the following safety instructions and explanations!  
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

### Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

### Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

### Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

#### **DANGER**

##### **Serious risk of injury!**

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

#### **WARNING**

##### **Risk of injury!**

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

#### **CAUTION**

##### **Personal injuries!**

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

#### **NOTE**

##### **Damage to the environment or devices**

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.



##### **Tip or pointer**

This symbol indicates information that contributes to better understanding.

## 1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

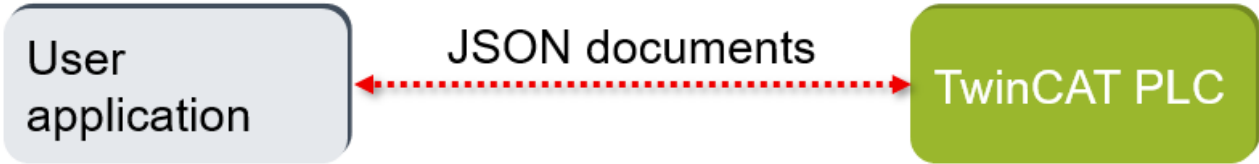
In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our <https://www.beckhoff.com/secguide>.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

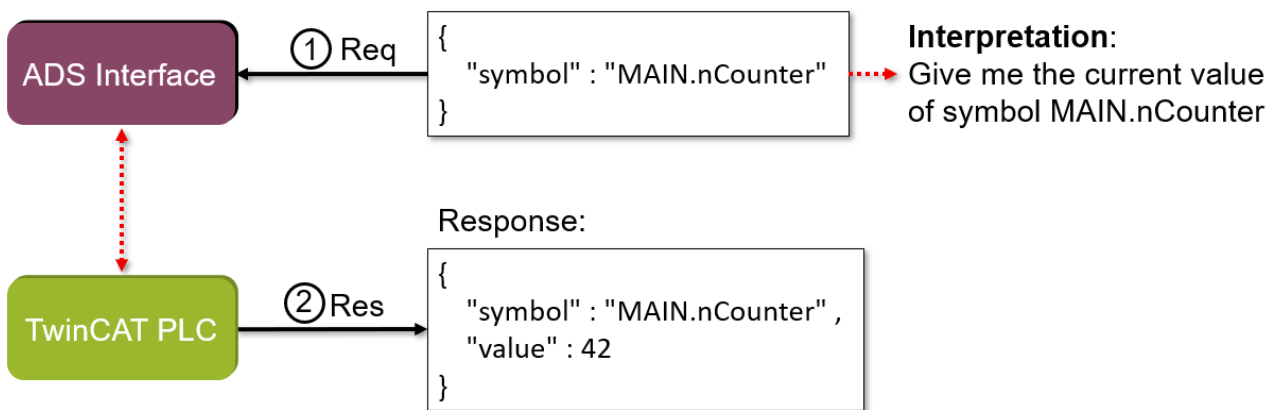
To stay informed about information security for Beckhoff products, subscribe to the RSS feed at <https://www.beckhoff.com/secinfo>.

## 2 Overview

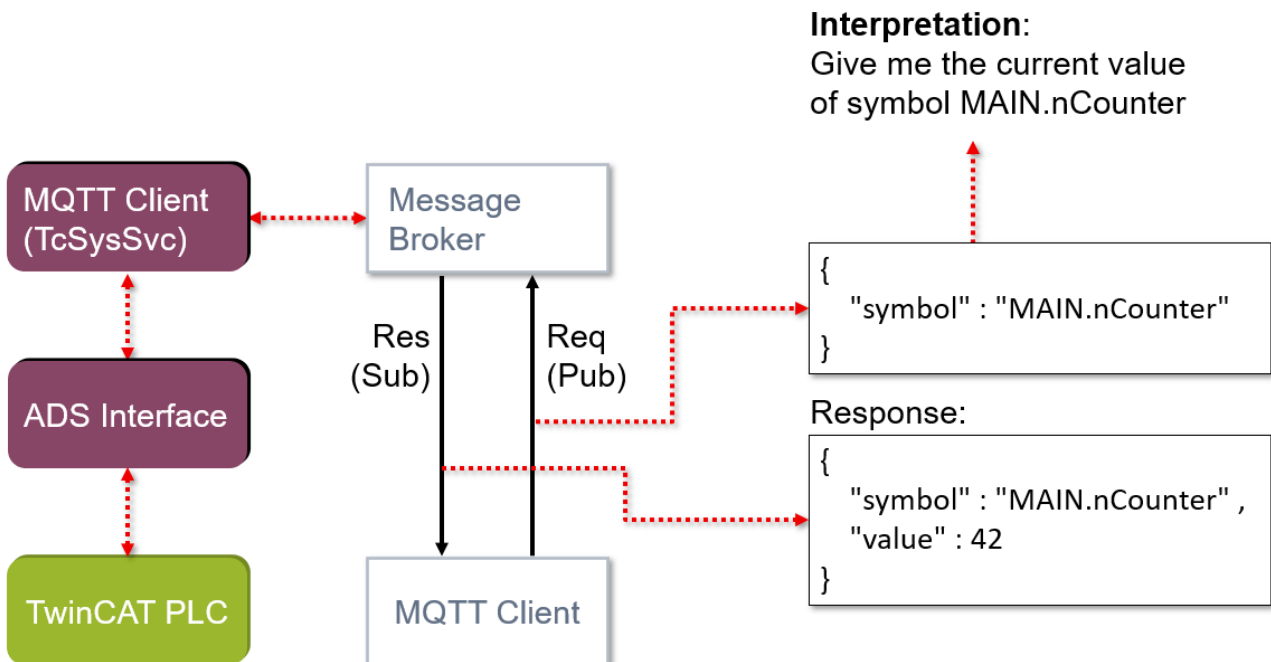
The TC3 JSON Data Interface is a communication interface for the flexibly adaptable exchange of data between TwinCAT systems and user-specific applications. The communication takes place in JSON format. The JSON format and the use of either ADS or MQTT ensures generic access from many different programming environments.



Basically, there are two different ways to access the JSON Data Interface. The first way is direct access to the ADS interface (cf. [Access via ADS \[► 12\]](#)).



In the second option, the TwinCAT System Service is included in the communication as an MQTT client and handles the requests to the ADS interface. A user application connects as an MQTT client to a message broker, to which the TwinCAT System Service is also connected (cf. [Access via MQTT \[► 13\]](#)).





## 3 Installation

### 3.1 System requirements

Technical data	Description
Operating system	Windows 7/10, Windows Embedded Standard 7, Windows CE 7, TwinCAT/BSD
Target platform	PC architecture (x86, x64 and ARM)
TwinCAT version	TwinCAT 3.1 build 4024.11 or higher
Required TwinCAT setup level	TwinCAT 3 XAE, XAR
Required TwinCAT license	TF6020 TC3 JSON Data Interface

### 3.2 Installation

No separate setup is required for the TF6020 JSON Data Interface function. All the components are supplied directly with the TwinCAT setup.

### 3.3 Licensing

The TwinCAT 3 function can be activated as a full version or as a 7-day test version. Both license types can be activated via the TwinCAT 3 development environment (XAE).

#### Licensing the full version of a TwinCAT 3 Function

A description of the procedure to license a full version can be found in the Beckhoff Information System in the documentation "[TwinCAT 3 Licensing](#)".

#### Licensing the 7-day test version of a TwinCAT 3 Function

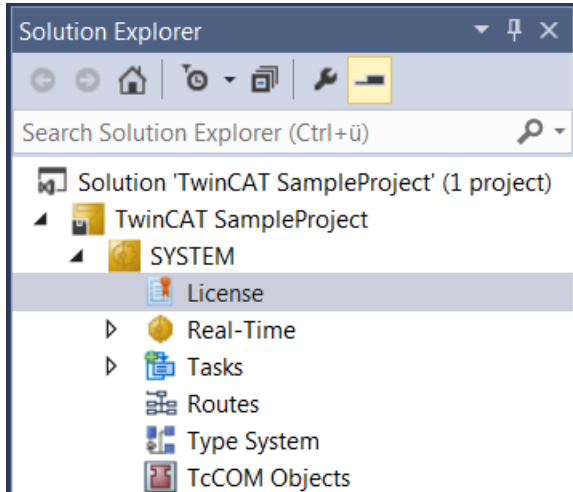


A 7-day test version cannot be enabled for a [TwinCAT 3 license dongle](#).

---

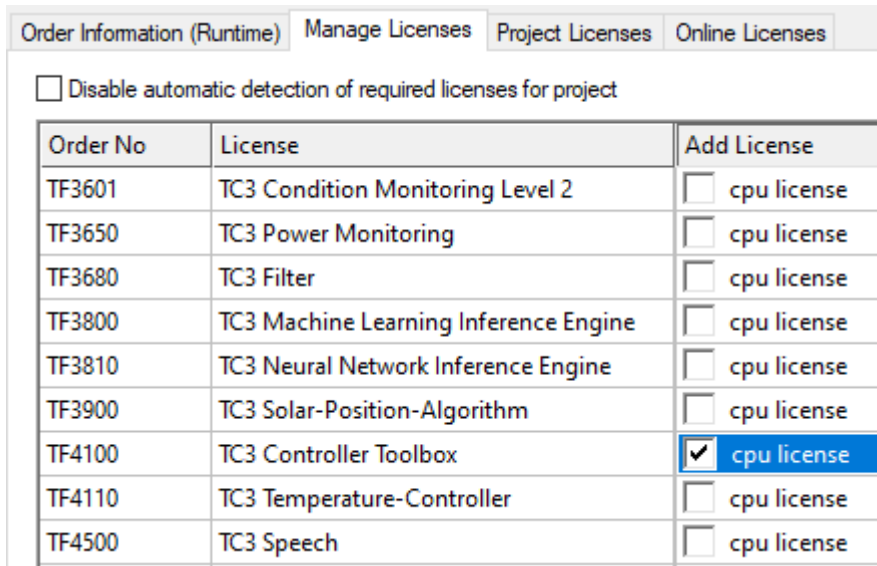
1. Start the TwinCAT 3 development environment (XAE).
2. Open an existing TwinCAT 3 project or create a new project.
3. If you want to activate the license for a remote device, set the desired target system. To do this, select the target system from the **Choose Target System** drop-down list in the toolbar.
  - ⇒ The licensing settings always refer to the selected target system. When the project is activated on the target system, the corresponding TwinCAT 3 licenses are automatically copied to this system.

4. In the **Solution Explorer**, double-click **License** in the **SYSTEM** subtree.



⇒ The TwinCAT 3 license manager opens.

5. Open the **Manage Licenses** tab. In the **Add License** column, check the check box for the license you want to add to your project (e.g. "TF4100 TC3 Controller Toolbox").



6. Open the **Order Information (Runtime)** tab.

⇒ In the tabular overview of licenses, the previously selected license is displayed with the status "missing".

7. Click **7-Day Trial License...** to activate the 7-day trial license.

⇒ A dialog box opens, prompting you to enter the security code displayed in the dialog.

8. Enter the code exactly as it is displayed and confirm the entry.

9. Confirm the subsequent dialog, which indicates the successful activation.

⇒ In the tabular overview of licenses, the license status now indicates the expiry date of the license.

10. Restart the TwinCAT system.

⇒ The 7-day trial version is enabled.

## 4 Technical introduction

### 4.1 API description

The following section describes the options provided by the JSON Data Interface with the help of short example requests and responses. The API is equally valid for access via MQTT or via ADS.

Type	Request/Response	Sample of the JSON payload
Reading a variable	Request	<code>{"symbol":"MAIN.nCounter"}</code>
	Response	<code>{"symbol":"MAIN.nCounter", "value":42}</code>
Reading a variable and its data type	Request	<code>{"symbol":"MAIN.nCounter", "datatype":null}</code>
	Response	<code>{"symbol":"MAIN.nCounter", "value":42, "datatype":INT}</code>
Simultaneous reading of several variables	Request	<code>[{"symbol": "MAIN.nCounter"}, {"symbol":"MAIN.nSecCounter"}]</code>
	Response	<code>[{"symbol":"MAIN.nCounter", "value":42 }, {"symbol":"MAIN.nSecCounter", "value":35}]</code>
Reading the TwinCAT symbol info	Request	<code>{"symbols":null}</code>
	Response	<i>Complete symbol info</i>
Writing a variable	Request	<code>{"symbol":"MAIN.nCounter", "value":42}</code>
	Response	<code>{"symbol":"MAIN.nCounter", "value":42}</code>
Writing several variables	Request	<code>[{"symbol":"MAIN.nCounter", "value":42}, {"symbol":"MAIN.nSecCounter", "value":35}]</code>
	Response	<code>[{"symbol":"MAIN.nCounter", "value":42}, {"symbol":"MAIN.nSecCounter", "value":35}]</code>
Method calls, optionally with input and output parameters	Request	<code>{"symbol": "MAIN.fbTester#M_Add", "parameter": {"intA":42, "intB":35}}</code>
	Response	<code>{"symbol":"MAIN.fbTester#M_Add", "value":77}</code>
	Response with additional outputs	<code>{"symbol":"MAIN.fbTester#M_Add", "value":77, "parameter":{"intC":32, "intD":64}}</code>

### 4.2 Access via ADS

One option for access is via MQTT and a message broker. In addition, access is also possible directly via ADS. In this case, the same interface is addressed that is addressed internally in the MQTT case.

The JSON documents corresponding to the API description (cf. [API description \[▶ 12\]](#)) are passed to the JSON Data Interface via ADS, and the response then also contains a JSON document. This is done by executing an ADS ReadWrite command with index group 0xf070 and index offset 0x0. The data to be written contains the request document, while the data to be read reserves the memory for the response document. The following code snippet shows how to access the JSON Data Interface using ADS .NET V5.

```

int adsPort = 851;
string responseString;

AdsClient adsClient = new AdsClient();
adsClient.Connect(adsPort);

string json = "{\"symbol\":\"MAIN.nCounter\"}";

byte[] writeData = new byte[json.Length+1];
MemoryStream writeStream = new MemoryStream(writeData);
BinaryWriter writer = new BinaryWriter(writeStream);
writer.Write(Encoding.ASCII.GetBytes(json));

byte[] readData = new byte[1000];

adsClient.ReadWrite(0xf070, 0, readData, writeData);

responseString = Encoding.ASCII.GetString(readData);

```

In principle, Ads-ReadWrite commands are always sent to the combination of Index Group and Index Offset described in the code. The response returned within the command then contains the JSON response of the JSON Data Interface.

## 4.3 Access via MQTT

If the JSON Data Interface is to be accessed via MQTT, the connection must be configured by the TwinCAT System Service with an MQTT broker. The *StaticRoutes.xml* must be edited for this. This can be found on a TwinCAT system in the following locations:

Windows: *C:\TwinCAT\3.1\Target*.

TC/BSD: */usr/local/etc/TwinCAT/3.1/Target*

### Settings without TLS

```

<?xml version="1.0" encoding="UTF-8"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RemoteConnections>
    <Json>
      <Name>SampleConnection</Name>
      <Address>18.157.241.86</Address>
      <Topic>topic</Topic>
    </Json>
  </RemoteConnections>
</TcConfig>

```

For settings without TLS, a RemoteConnection of the type Json must be added to *StaticRoutes.xml*. A name for the connection, the IP address of the MQTT broker and the main topic are then specified within this RemoteConnection.

### Settings with certificates

```

<?xml version="1.0" encoding="UTF-8"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RemoteConnections>
    <Json>
      <Name>SampleConnection</Name>
      <Address>18.157.241.86</Address>
      <Topic>topic</Topic>
      <Tls>
        <Ca>C:\Users\Administrator\Certificate\rootCA.pem</Ca>
        <Cert>C:\Users\Administrator\Certificate\TF6020.pem</Cert>
        <Key>C:\Users\Administrator\Certificate\TF6020.key</Key>
        <Crl>C:\Users\Administrator\Certificate\SampleCRL.der</Crl>
      </Tls>
    </Json>
  </RemoteConnections>
</TcConfig>

```

If TLS is to be used, there is also the option of referencing certificates. The CA certificate, the client certificate and the client's private key can be specified either DER- or PEM-encoded or by means of a file path. A Certificate Revocation List (CRL) can also be specified in the same format.

### Set user name and password

```
<?xml version="1.0" encoding="UTF-8"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RemoteConnections>
    <Json>
      <Name>SampleConnection</Name>
      <Address>18.157.241.86</Address>
      <Topic>topic</Topic>
      <User>Beckhoff</User>
      <Pwd>8c3nfd3"!4c</Pwd>
      <Tls>
        <Ca>C:\Users\Administrator\Certificate\rootCA.pem</Ca>
        <Cert>C:\Users\Administrator\Certificate\TF6020.pem</Cert>
        <Key>C:\Users\Administrator\Certificate\TF6020.key</Key>
        <Crl>C:\Users\Administrator\Certificate\SampleCRL.der</Crl>
      </Tls>
    </Json>
  </RemoteConnections>
</TcConfig>
```

If access to a broker is used that does not allow anonymous access, the configuration of user name and password becomes necessary. For this purpose, the user name and password are stored in the appropriate places in the configuration file.

### 4.3.1 Topic structure

If MQTT is used for access to the TwinCAT JSON Data Interface, communication takes place via the broker with a defined topic structure. On one side there is a request-topic, on which a requesting client sends its JSON document and on the other side there is the response-topic, on which it receives the response of the JSON Data Interface.

- Topic for sending requests:      mainTopic/req/adsPort/invokeld
- Topic for receiving responses:    mainTopic/res/adsPort/invokeld

**mainTopic:** The main topic is defined in the configuration file (cf. [Access via MQTT \[▶ 13\]](#)).

**req/res:** set.

**adsPort:** The ADS port to be queried is entered here.

**invokeld:** The Invoke ID is arbitrarily chosen by the user and is used to identify different requests to the same JSON Data Interface.

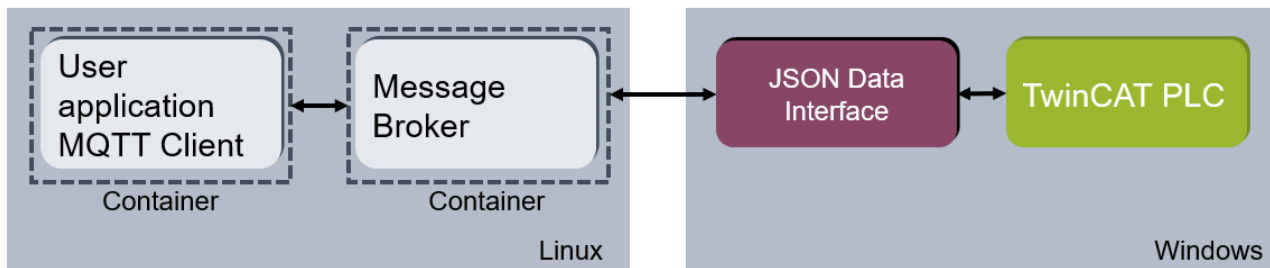
The JSON Data Interface mirrors the Invoke ID of a request topic and reuses it in the response topic. The requesting client must be subscribed to this response topic in order to receive the response.

## 5 Examples

This chapter deals with exemplary use cases for the TwinCAT JSON Data Interface. Example calls for all available commands can be found in the [API description](#) [▶ 12].

### Easy access from other operating systems

The JSON Data Interface represents a simple way of accessing the symbols of a TwinCAT system from other operating systems. Two communication channels are available for this purpose. On the one hand, as in the application idea below, both the client application and the JSON Data Interface can be connected to the same message broker.

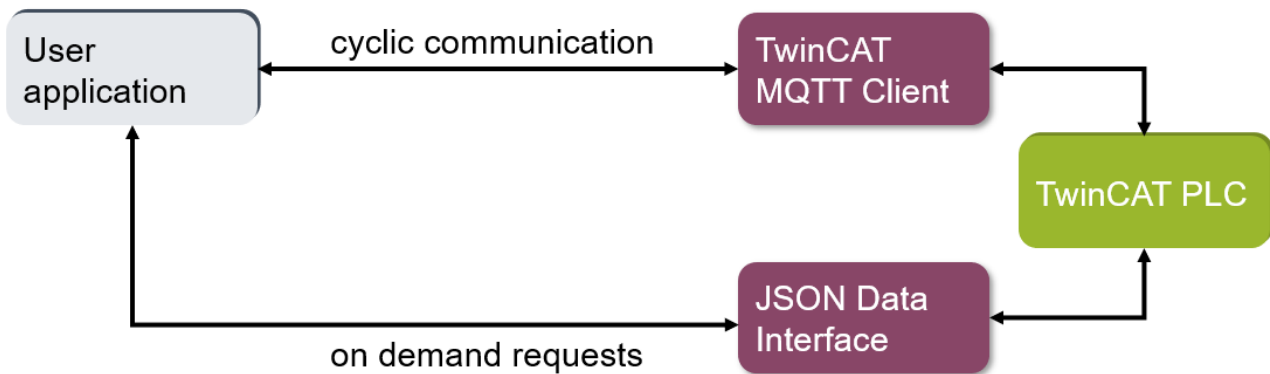


It is freely selectable on which of the two systems the message broker runs or whether it even runs on a third system. There is no technical difference here, there just has to be a connection from both sides to the same broker.

On the other hand, ADS communication can also be used on other operating systems. An ADS router is always required for ADS communication on a system. On other operating systems, however, this ADS router can also be replaced by a standalone router with reduced functionality, which is provided by TwinCAT ADS .NET V5.

### Two communication types in one application

The JSON Data Interface is also suitable for mapping two types of communication in an application. Another component such as the TwinCAT MQTT Client (TF6701) can be used for cyclic communication. Here, previously selected variables are then cyclically communicated to the user application via MQTT.



The user application then decides which scenarios to react to and how. If cyclic communication detects that more information is needed, the read commands of the JSON Data Interface can be used to query variables that are not communicated cyclically to the user application. On the other hand, if it is recognized that certain parameters need to be affected, the write commands of the JSON Data Interface can be used.

### Simple retrofitting of existing projects

The JSON Data Interface is available with TwinCAT version 3.1.4024.11 or higher. The following section only refers to TwinCAT systems with this or newer TwinCAT versions.

Direct access to the JSON Data Interface via ADS is recommended if no intervention in the TwinCAT system is possible or if this is not desired. Access via MQTT, on the other hand, offers more communication options, but requires adjustments on the target system. These adjustments are the editing of StaticRoutes.xml and a TwinCAT context change.

## 6 Appendix

### 6.1 Error diagnosis

Behavior	Description
The communicated JSON string is returned in the same form by the JSON Data Interface.	Check the sent JSON string for correct formatting. For example, a common mistake is to use "symbols" instead of "symbol".
The JSON Data Interface does not respond to my requests via MQTT.	Verify that both MQTT clients are connected to the message broker. Many message brokers offer monitoring functions for this purpose. If an error occurs, check why one of the MQTT clients is not connected.
The JSON Data Interface responds to a request with a message about a missing license.	Please activate the license for TF6020 via the known TwinCAT license mechanisms.





More Information:  
[www.beckhoff.com/tf6020](http://www.beckhoff.com/tf6020)

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Germany  
Phone: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

