



KUKA System Technology

KUKA.DeviceConnector 2.1 and variants

For KUKA System Software 8.3, 8.5, 8.6 and 8.7

For VW System Software 8.3, 8.6 and 8.7



Issued: 10.09.2021

KUKA.DeviceConnector 2.1 V4

KUKA Deutschland GmbH

© Copyright 2021

KUKA Deutschland GmbH

Zugspitzstraße 140

D-86165 Augsburg

Germany

This documentation or excerpts therefrom may not be reproduced or disclosed to third parties without the express permission of KUKA Deutschland GmbH.

Other functions not described in this documentation may be operable in the controller. The user has no claims to these functions, however, in the case of a replacement or service work.

We have checked the content of this documentation for conformity with the hardware and software described. Nevertheless, discrepancies cannot be precluded, for which reason we are not able to guarantee total conformity. The information in this documentation is checked on a regular basis, however, and necessary corrections will be incorporated in the subsequent edition.

Subject to technical alterations without an effect on the function.

KIM-PS5-DOC

Translation of the original documentation

Publication: Pub KUKA.DeviceConnector 2.1 (PDF) en
PB13101

Book structure: KUKA.DeviceConnector 2.1 V4.1
BS11870

Version: KUKA.DeviceConnector 2.1 V4

Contents

1	Introduction.....	5
1.1	Target group.....	5
1.2	Industrial robot documentation.....	5
1.3	Representation of warnings and notes.....	5
1.4	Terms used.....	6
1.5	Trademarks.....	6
1.6	Licenses.....	7
2	Product description.....	9
2.1	Product description.....	9
2.2	Variants of the option package.....	9
2.3	Data processing with OPC UA client or MQTT broker.....	10
2.4	Intended use and misuse.....	11
3	Safety.....	13
3.1	Safety measures for single point of control.....	13
4	Installation.....	15
4.1	System requirements.....	15
4.2	Installation via smartHMI.....	15
4.2.1	Installing or updating the option package.....	15
4.2.2	Uninstalling the option package.....	16
4.3	Installation via WorkVisual.....	17
4.3.1	Installing or updating the option package.....	17
4.3.2	Uninstalling the option package.....	18
5	Operation.....	19
5.1	Connecting an OPC UA client to the server.....	19
5.2	OPC UA user groups.....	19
5.3	Data according to VDMA standard.....	20
5.3.1	Information about the robot controller.....	21
5.3.2	Information about the kinematic system.....	22
5.3.3	Information on safety states.....	23
5.4	File transfer.....	24
5.5	Messages and events.....	26
5.6	Process data.....	27
5.7	Customer-specific variables (CustomerData).....	28
5.8	Customer-specific information (CustomerInfo).....	29
5.9	Soft PLC data.....	31
5.10	PROFINET data.....	32
5.11	Diagnostic data.....	33
5.12	OPC UA node IDs.....	34
6	Configuration.....	35
6.1	Changing the OPC UA user password.....	35
6.2	Creating a self-signed certificate for authentication.....	35
6.3	Configuring OPC UA process data.....	36
6.3.1	DataSets.....	36

6.3.2	DataPoint.....	37
6.3.3	auto_fill_source.....	39
6.4	Message Queuing Telemetry Transport (MQTT).....	39
6.4.1	ConnectionParameters.....	40
6.4.2	DataSets.....	41
6.4.3	DataPoints.....	44
7	Troubleshooting.....	47
7.1	No connection to the OPC UA server.....	47
7.2	MQTT connection error.....	48
7.2.1	Incorrect target address.....	48
7.2.2	Invalid certificate.....	48
8	Messages.....	51
8.1	Information about the messages.....	51
8.2	System messages from module: DeviceConnector.....	51
8.2.1	DeviceConnector 00001.....	51
8.2.2	DeviceConnector 00003.....	53
8.2.3	DeviceConnector 00006.....	53
8.2.4	DeviceConnector 00007.....	54
8.2.5	DeviceConnector 00008.....	54
8.2.6	DeviceConnector 00009.....	55
8.2.7	DeviceConnector 00010.....	56
9	Appendix.....	59
9.1	Configuration file "ProcessData.opcj" for OPC UA.....	59
9.2	Example of a configuration file for MQTT.....	60
10	KUKA Service.....	63
10.1	Requesting support.....	63
10.2	KUKA Customer Support.....	63
	Index	65

1 Introduction

1.1 Target group

This documentation is aimed at users with the following knowledge and skills:

- Advanced knowledge of the robot controller system
- Advanced knowledge of network connections



For optimal use of KUKA products, we recommend the training courses offered by KUKA College. Information about the training program can be found at www.kuka.com or can be obtained directly from our subsidiaries.

1.2 Industrial robot documentation

The industrial robot documentation consists of the following parts:

- Documentation for the robot arm
- Documentation for the robot controller
- Documentation for the smartPAD-2 (if used)
- Operating and programming instructions for the System Software
- Instructions for options and accessories
- Spare parts overview in KUKA Xpert

Each of these sets of instructions is a separate document.

1.3 Representation of warnings and notes

Safety

These warnings are provided for safety purposes and **must** be observed.



DANGER

These warnings mean that it is certain or highly probable that death or severe injuries **will** occur, if no precautions are taken.



WARNING

These warnings mean that death or severe injuries **may** occur, if no precautions are taken.



CAUTION

These warnings mean that minor injuries **may** occur, if no precautions are taken.

NOTICE

These warnings mean that damage to property **may** occur, if no precautions are taken.



These warnings contain references to safety-relevant information or general safety measures.
These warnings do not refer to individual hazards or individual precautionary measures.

This warning draws attention to procedures which serve to prevent or remedy emergencies or malfunctions:

SAFETY INSTRUCTION

The following procedure must be followed exactly!

Procedures marked with this warning **must** be followed exactly.

Notices

These notices serve to make your work easier or contain references to further information.



Tip to make your work easier or reference to further information.

1.4 Terms used

Term	Description
IP	Internet Protocol The Internet Protocol is used to define subnetworks by means of physical MAC addresses.
KLI	KUKA Line Interface Ethernet interface of the robot controller for external communication.
KONI	KUKA.OptionalNetworkInterface Optional Ethernet interface of the robot controller for external communication.
MQTT	Message Queuing Telemetry Transport Open messaging protocol for transferring telemetry data between devices, which can be used even with high delays and limited networks
OPC UA	Open Platform Communications - Unified Architecture Open standard for data exchange between devices (vendor- and platform-independent)
SPOC	Single Point of Control
TCP	Transmission Control Protocol Protocol of the data exchange between devices of a network. TCP constitutes a virtual channel between 2 sockets in a network connection. Data can be transmitted on this channel in both directions.

1.5 Trademarks

mosquitto is a trademark of the Eclipse Foundation.

PROFINET® is a trademark of the PROFIBUS users' organization.

UaExpert® is a trademark of Unified Automation GmbH.

Windows is a trademark of Microsoft Corporation.

1.6 Licenses

The KUKA license conditions and the license conditions of the open-source software used can be found in the following folders:

- Under .\LICENSE on the data storage medium with the installation files of the KUKA software
- On the KUKA smartHMI, in the main menu under **Help > Info**, tab **Licenses**
- Under D:\KUKA_OPT*Option package name*\LICENSE on the robot controller
- In the license folder under the name of the option package in the **Options** catalog in WorkVisual



Further information about open-source licenses can be requested from the following address: opensource@kuka.com

2 Product description

2.1 Product description

Description

The KUKA.DeviceConnector 2.1 option package enables access to robot data via OPC UA and MQTT.

- OPC UA
Data can be transferred bidirectionally within a network.
- MQTT
Data can be transferred unidirectionally across network boundaries.

KUKA.DeviceConnector 2.1 enables customer-specific server peripherals to be integrated.

Functions

OPC UA:

- Robot controller as OPC UA server
- Connection of multiple OPC UA clients
- Reading of robot data
- Reading and writing of process variables

MQTT:

- Robot controller as MQTT client
- Connection of multiple MQTT brokers
- Reading of robot data
- Reading of process variables

2.2 Variants of the option package

The option package is available in a number of variants: These variants differ in the amount of data they provide:

- KUKA.DeviceConnector 2.1
Basic package for access to all data defined in VDMA standard 40010-1 for robotics
- KUKA.DeviceConnector FileTransfer 2.1
Package for access to the file system of the robot controller, including LOG files
- KUKA.DeviceConnector Messages 2.1
Package for access to all messages and Windows event logs of the robot controller
- KUKA.DeviceConnector SystemVars 2.1
Package for access to all system variables and global KRL variables including customer-specific variables (CustomerData)
- KUKA.DeviceConnector CustomerData 2.1
Package for access to all customer-specific variables
- KUKA.DeviceConnector SoftPLC 2.1
Package for access to all Soft PLC data
Precondition: KUKA.PLC ProConOS or KUKA.PLC ProConOS Ltd is installed.

- KUKA.DeviceConnector Advanced 2.1
Complete package with all available data and functions, including:
 - Soft PLC data
Precondition: KUKA.PLC ProConOS or KUKA.PLC ProConOS Ltd is installed.
 - PROFINET data
Precondition: KUKA.PROFINET® -/S or KUKA.PROFINET® M/S is installed.

Depending on the application scenario, required data points and file information, the option packages can be installed in combination.

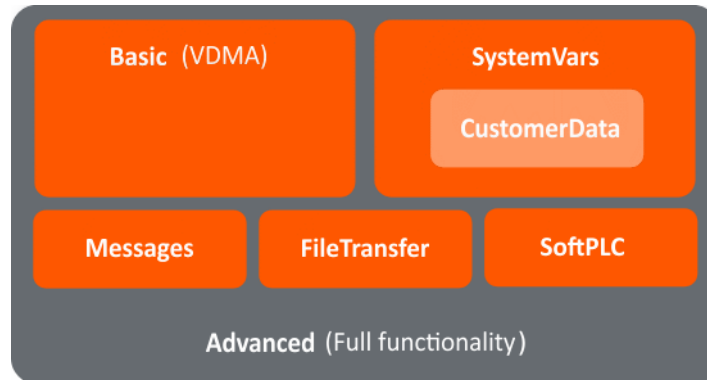


Fig. 2-1: Overview of variants

Exceptions:

- Advanced option cannot be combined (contains all available data)
- CustomerData and SystemVars options cannot be combined (CustomerData included in SystemVars)

2.3 Data processing with OPC UA client or MQTT broker

Description

In order to be able to process the data published with KUKA.DeviceConnector, an OPC UA client or MQTT broker is required.

- An OPC UA client can connect to KUKA.DeviceConnector via the KLI or KONI of the robot controller.
 - Port 4840 is reserved for OPC UA. The connection can only be established via this port.
- KUKA.DeviceConnector can connect to one or more MQTT brokers.
 - For connection configuration, see: (>>> [6.4 "Message Queuing Telemetry Transport \(MQTT\)" Page 39](#))

Demo clients

We recommend using the following demo clients for test purposes.

- For OPC UA: **UaExpert v1.4.x and higher**
The UaExpert demo client can be downloaded free of charge from the website of Unified Automation GmbH:
www.unified-automation.com
- For MQTT: **mosquitto 1.6.2 or higher**

2.4 Intended use and misuse

Use

The KUKA.DeviceConnector 2.1 option package enables access to robot data via OPC UA and MQTT and is used to read and write variables. There are absolutely no safety-relevant functions. Safety-relevant data cannot be influenced/written to via the interface.

KUKA.DeviceConnector 2.1 may be incorrectly configured, with the effect that some variables can no longer be output. Operation in accordance with the intended use thus also involves compliance with the configuration instructions in the documentation of the option package.

Misuse

Any use or application deviating from the intended use is deemed to be misuse and is not allowed. It will result in the loss of warranty and liability claims. KUKA is not liable for any damage resulting from such misuse.

3 Safety

This documentation contains safety instructions which refer specifically to the option package described here.

The fundamental safety information for the industrial robot can be found in the “Safety” chapter of the Operating and Programming Instructions for System Integrators or the Operating and Programming Instructions for End Users.



Comply with safety-relevant information

The safe use of this product requires knowledge of and compliance with fundamental safety measures. Death, severe injuries or damage to property may otherwise result.

- The “Safety” chapter in the operating and programming instructions of the KUKA System Software (KSS) must be observed.
- The principle of “single point of control” (SPOC) must be completely adhered to.



WARNING

Danger to life and limb due to non-secure signal changes

Signal states can be changed by accessing an OPC UA client or by transferring a project from WorkVisual. Death, severe injuries and damage to property may result from non-secure signal changes.

- Only execute dangerous signals, e.g. the signal for opening/closing a gun, if at least one of the following conditions is met:
 - Safety gate is closed.
 - Automatic mode
 - Peripheral contactor (US2) is used.
 - The variables **bSPOC_UserSafetyActive** and **bSPOC_MotionEnabled** were taken into account when mapping the signals.



Keeping the number of OPC UA connections to a minimum

KUKA.DeviceConnector allows up to 60 individual OPC UA connections through which a manipulation of the robot controller is possible. This may result in unauthorized access to the robot controller.

- Keep the number of OPC UA clients and authorized OPC UA users to a minimum.

3.1 Safety measures for single point of control

Overview

If certain components are used on the industrial robot, safety measures must be taken to ensure complete implementation of the principle of “single point of control” (SPOC).

The relevant components are:

- Submit interpreter
- PLC
- OPC server
- Remote control tools
- Tools for configuration of bus systems with online functionality
- KUKA.RobotSensorInterface

- KUKA.DeviceConnector
(not KUKA.DeviceConnector pre-installed)

Since only the system integrator knows the safe states of actuators in the periphery of the robot controller, it is his task to set these actuators to a safe state, e.g. in the event of an EMERGENCY STOP.



Further safety measures for Single Point of Control

Depending on the specific application, further safety measures may be required to ensure complete implementation of the principle of “single point of control”. Failure to take this precaution into consideration may result in death, injuries or damage to property.

- Check whether further safety measures are required; if so, implement them.

T1, T2

In modes T1 and T2, the components referred to above may only access the industrial robot if the following signals have the following states:

Signal	State required for SPOC
\$USER_SAF	TRUE
\$SPOC_MOTION_ENABLE	TRUE

Submit interpreter, PLC

If motions, (e.g. drives or grippers) are controlled with the submit interpreter or the PLC via the I/O system, and if they are not safeguarded by other means, then this control will take effect even in T1 and T2 modes or while an EMERGENCY STOP is active.

If variables that affect the robot motion (e.g. override) are modified with the submit interpreter or the PLC, this takes effect even in T1 and T2 modes or while an EMERGENCY STOP is active.

- In T1 and T2, the system variable \$OV_PRO must not be written to by the submit interpreter or the PLC.

OPC server, KUKA.DeviceConnector, remote control tools

These components can be used with write access to modify programs, outputs or other parameters of the robot controller, without this being noticed by any persons located inside the system.

Safety measure:

If these components are used, outputs that could cause a hazard must be determined in a risk assessment. These outputs must be designed in such a way that they cannot be set without being enabled. This can be done using an external enabling device, for example.

Tools for configuration of bus systems

If these components have an online functionality, they can be used with write access to modify programs, outputs or other parameters of the robot controller, without this being noticed by any persons located inside the system.

- WorkVisual from KUKA
- Tools from other manufacturers

Safety measure:

In the test modes, programs, outputs or other parameters of the robot controller must not be modified using these components.

4 Installation

The option package can either be installed on the robot controller via the smarHMI or via WorkVisual.

4.1 System requirements

Hardware

- KR C4 or KR C5 robot controller
- or VKR C4 robot controller
- Laptop/PC

Software

Robot controller:

- KUKA System Software 8.3, 8.5, 8.6 or 8.7
- Or VW System Software 8.3, 8.6 or 8.7
- KUKA.DeviceConnector pre-installed 2.1

Laptop/PC:

- For KUKA and VW System Software 8.3:
 - WorkVisual 4.0 or higher
- For KUKA System Software 8.5 or higher:
 - WorkVisual 5.0 or higher
- For KUKA and VW System Software 8.6 and 8.7:
 - WorkVisual 6.0

Compatibility

KUKA.DeviceConnector 2.1 must not be installed on a robot controller together with the following option package:

- KUKA.OPC Server
- KUKA.OPC UA

4.2 Installation via smarHMI

4.2.1 Installing or updating the option package



It is advisable to archive all relevant data before updating a software package.

Precondition

- User rights:
 - KSS: function group **General configuration**
But at least the user group “Expert”
 - VSS: user group “User”
- T1 or T2 mode
- No program is selected.
- USB stick with the option package (KOP file)

NOTICE**Data loss due to USB sticks from third-party manufacturers**

Data may be lost if USB sticks from manufacturers other than KUKA are used for activities on the robot controller.

- For activities on the robot controller requiring a USB stick, use a KUKA stick.

The KUKA sticks are validated for use with the robot controller.

Procedure

1. Connect the USB stick to the robot controller or smartPAD.
2. In the main menu, select **Start-up > Additional software**.
3. Press **New software**: an entry for the option package (name and version) must be displayed in the **Name** column and drive **E:** or **K:** in the **Path** column.
If not, press **Refresh**.
4. If the entry for the option package is now displayed, continue with step 5.
Otherwise, the path from which the software is to be installed must be configured first:
 - a. Press the **Configure** button.
 - b. Select a line in the **Installation paths for options** area.
Note: If the line already contains a path, this path will be overwritten.
 - c. Press **Path selection**. The available drives are displayed.
 - d. If the stick is connected to the robot controller: Select **E:**.
If the stick is connected to the smartPAD: **K:** instead of **E:**
 - e. Press **Save**. The **Installation paths for options** area is displayed again. It now contains the new path.
 - f. Mark the line with the new path and press **Save** again.
5. Activate the check box next to the option package and press **Install**. Confirm the installation query with **OK**.
6. The request for confirmation *Do you want to activate the project [...]?* is displayed. The active project is overwritten during activation. If no relevant project will be overwritten: answer the query with **Yes**.
7. An overview with the changes and a request for confirmation are displayed. Answer this with **Yes**. The option package is installed and the robot controller carries out a reboot.
8. Remove the stick.

LOG file

A LOG file is created under C:\KRC\ROBOTER\LOG.

4.2.2 Uninstalling the option package

It is advisable to archive all relevant data before uninstalling a software package.

Precondition

- User rights:
 - KSS: function group **General configuration**
But at least the user group "Expert"

- VSS: user group “User”
- T1 or T2 mode
- No program is selected.

Procedure

1. In the main menu, select **Start-up > Additional software**.
2. Activate the check box next to the option package and press **Uninstall**. Answer the request for confirmation with **Yes**.
3. Answer the request for confirmation *Do you want to activate the project [...]?* with **Yes**.
4. An overview with the changes and a request for confirmation are displayed. Answer this with **Yes**. The option package is uninstalled and the robot controller carries out a reboot.

LOG file

A LOG file is created under C:\KRC\ROBOTER\LOG.

4.3 Installation via WorkVisual

4.3.1 Installing or updating the option package

Description

The option package is installed in WorkVisual and added to the project. During project deployment, the option package is automatically installed on the robot controller.



It is advisable to archive all relevant data before updating a software package.

Precondition

- User rights:
 - KSS: user group “Expert”
 - VSS: user group “User”
- T1 or T2 mode
- No program is selected.
- Network connection to the robot controller
- The option package is available as a KOP file.

Procedure

1. Install the option package in WorkVisual.
2. Load the active project from the robot controller.
3. Insert the option package into the project.
4. Deploy the project from WorkVisual to the robot controller and activate it.
5. The request for confirmation *Do you want to activate the project [...]?* is displayed on the smartHMI. The active project is overwritten during activation. If no relevant project will be overwritten: Answer the query with **Yes**.
6. An overview with the changes and a request for confirmation are displayed on the smartHMI. Answer this with **Yes**. The option package is installed and the robot controller carries out a reboot.



Information about procedures in WorkVisual is contained in the WorkVisual documentation.

LOG file

A LOG file is created under C:\KRC\ROBOTER\LOG.

4.3.2 Uninstalling the option package

Description

The option package can be uninstalled via WorkVisual.



It is advisable to archive all relevant data before uninstalling a software package.

Precondition

- User rights:
 - KSS: user group “Expert”
 - VSS: user group “User”
- T1 or T2 mode
- No program is selected.
- Network connection to the robot controller

Procedure

1. Load the project from the robot controller.
2. Remove the option package from the project. A window with modifications is displayed.
3. Deploy the project from WorkVisual to the robot controller and activate it.
4. Answer the request for confirmation *Do you want to activate the project [...]?* on the smartHMI with **Yes**.
5. An overview with the changes and a request for confirmation are displayed on the smartHMI. Answer this with **Yes**. The option package is uninstalled and the robot controller carries out a reboot.



Information about procedures in WorkVisual is contained in the WorkVisual documentation.

LOG file

A LOG file is created under C:\KRC\ROBOTER\LOG.

5 Operation



The following descriptions and procedures are based on the UaExpert demo client.

5.1 Connecting an OPC UA client to the server

Description

The robot controller with installed KUKA.DeviceConnector serves as the OPC UA server. This must be added in the OPC UA client to connect the client to the server.

If the client is connected to the server, the data are retrieved from the robot controller and displayed in the **Address Space** window.

Precondition

- IP address or name of the robot controller is known.

Procedure

- Enter the following URL when adding the server:
 - `opc.tcp://IP address:4840/`
 - Or: `opc.tcp://Controller name:4840/`

5.2 OPC UA user groups

Access to the data is controlled via user groups:

- Anonymous
May browse all OPC UA variables.
- OpcUaObserver
May browse and read all OPC UA variables.
- OpcUaOperator
May browse, read and write all OPC UA variables.
- OpcUaConfigAdmin
May configure the MQTT Publisher with a suitable OPC UA client.

All user groups except Anonymous are password-protected. The default password for all groups is "kuka". The password can be changed for each group.

(>>> [6.1 "Changing the OPC UA user password" Page 35](#))



Changing default passwords

If the default passwords are not changed, this enables unauthorized persons to log on.

- Before start-up, change the passwords for the user groups.
- Only communicate the passwords to authorized personnel.

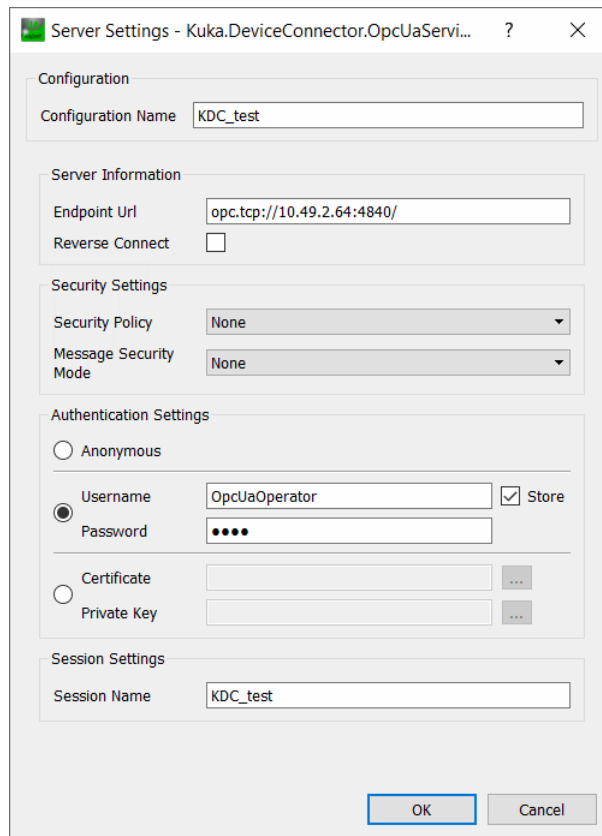


Fig. 5-1: Authentication via password

As an alternative to authentication via password, a self-signed certificate can be used.

(>>> [6.2 "Creating a self-signed certificate for authentication" Page 35](#))

5.3 Data according to VDMA standard



These data are available with the following option packages:

- KUKA.DeviceConnector
- KUKA.DeviceConnector Advanced

In accordance with VDMA standard 40010-1, the following data are available under the **MotionDeviceSystem** node:

- Controllers > KRC
Information about the robot controller
(>>> [5.3.1 "Information about the robot controller" Page 21](#))
- MotionDevices > Robot
Information about the kinematic system
(>>> [5.3.2 "Information about the kinematic system" Page 22](#))
- SafetyStates > SafetyState
Information about safety states of the robot controller
(>>> [5.3.3 "Information on safety states" Page 23](#))



Further information about the data can be found in the following document from the OPC Foundation:

- OPC UA Companion Specification Robotics - Part 1

5.3.1 Information about the robot controller

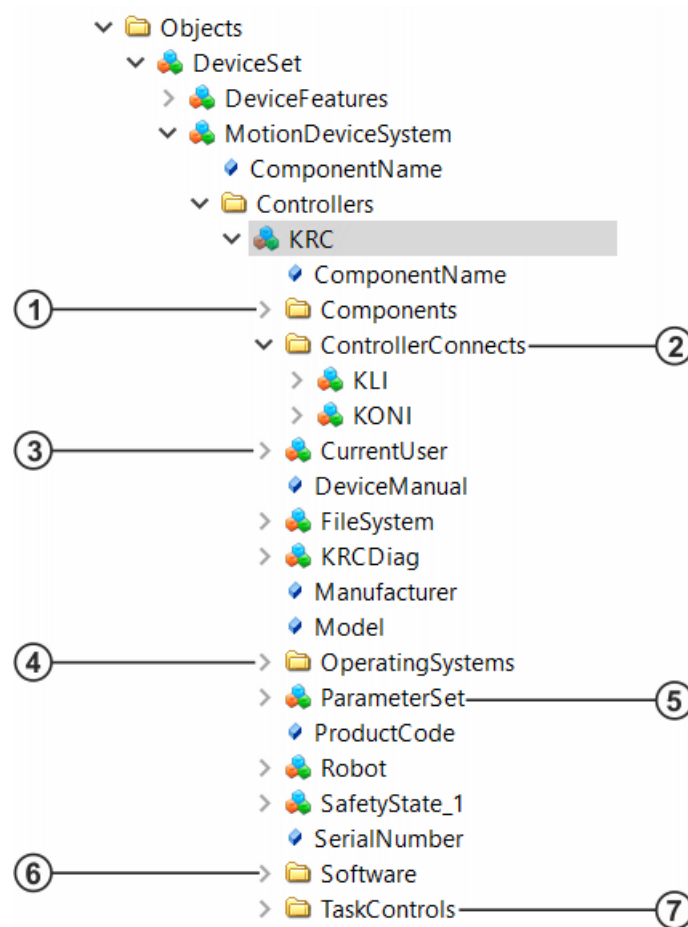


Fig. 5-2: Information about the robot controller

Item	Description
1	Electronic components of the robot controller with information on model name, serial number, firmware version, etc.
2	Network configuration
3	User currently logged on
4	Operating systems of the robot controller incl. Windows registry
5	Operating information, e.g.: • Robot controller runtime • Activated project • IP address of the robot controller
6	System Software and installed option packages with version, release date, installation date, etc.
7	Information about robot interpreter and submit interpreter, e.g.: • Program name • Program status • Call stack status

5.3.2 Information about the kinematic system

A wide range of information is available for each kinematic system that is part of the robot system. The kinematic system can be a robot, linear unit or positioner, for example.

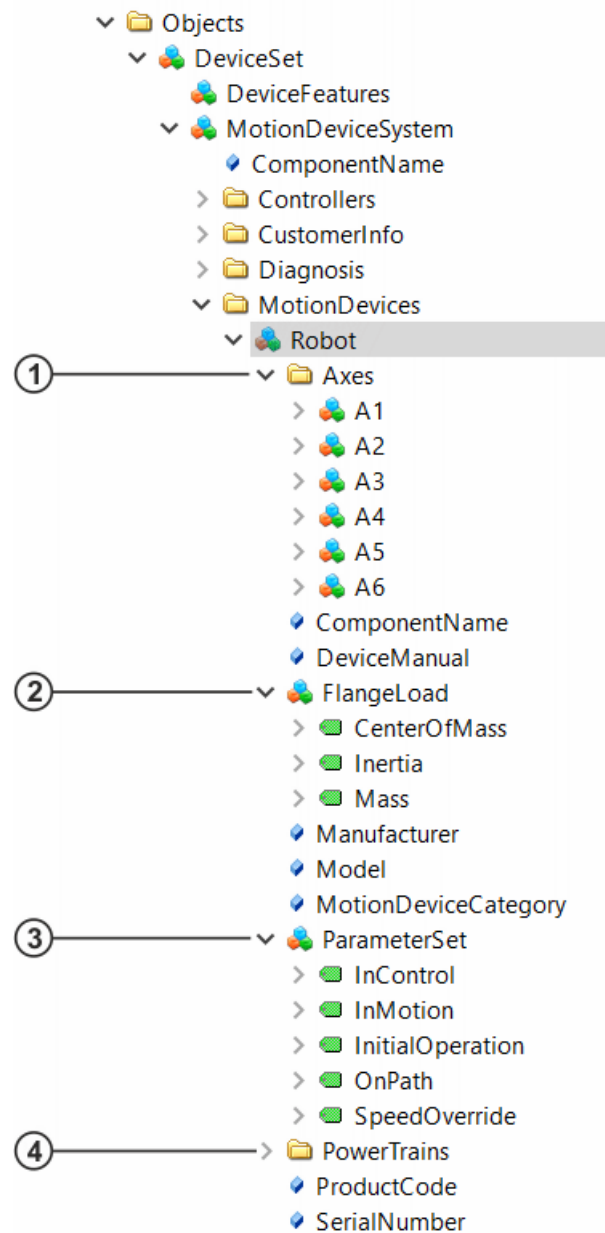


Fig. 5-3: Information about the kinematic system

Item	Description
1	Axes of the kinematic system
2	Payload data on the flange
3	Information about the robot motion, e.g.: • Drives under servo-control • Motion status • Program override

Item	Description
4	Information about the drives of the axes of a kinematic system, e.g.: • Brakes released/applied • Maximum torque deviation • Motor temperature

Further information is available for each axis of a kinematic system, e.g.:

- Supplementary load on axis
- Current axis position
- Current axis velocity

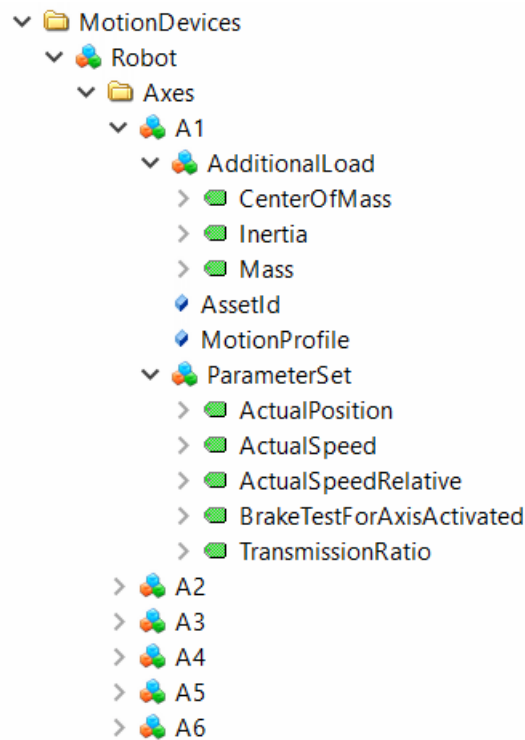


Fig. 5-4: Axis information

5.3.3 Information on safety states

The SafetyStates object provides information about safety states of the robot controller, e.g.:

- EMERGENCY STOP
- Enabling device
- Operating mode
- Safety stop

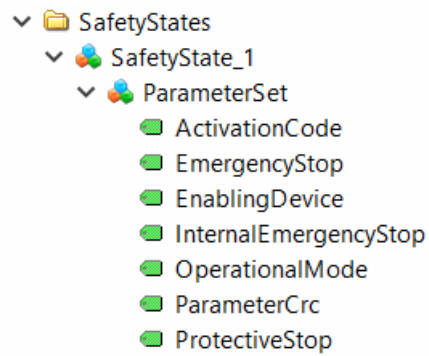


Fig. 5-5: Safety states

5.4 File transfer

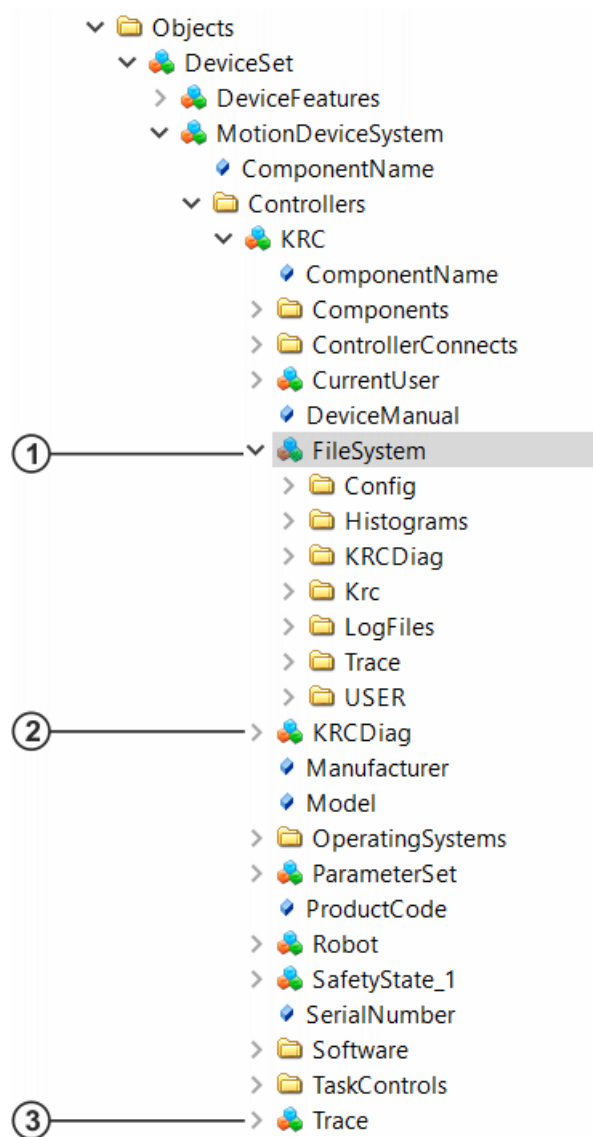


This functionality is available with the following option packages:

- KUKA.DeviceConnector FileTransfer
- KUKA.DeviceConnector Advanced

Description

The OPC UA client can access the file system of the robot controller as well as KRCDiags and traces.

**Fig. 5-6: File transfer**

Item	Description
1	File system of the robot controller with all configuration files, program files, LOG files, etc. • ...\KRC\Roboter\Config\User\ with the user-specific configuration files • ...\KRCDiag with the diagnostic packages • ...\KRC\Roboter\KRC\ with the robot programs, machine data, etc. • ...\KRC\Roboter\Log\ with the LOG files • ...\KRC\Roboter\Trace\ with the trace definitions and results • ...\KRC\Roboter\UserFiles\ with the user-specific files
2	KRCDiag for creating a diagnostic package
3	Trace function for recording and evaluating data (in accordance with trace configuration)

File operations

The OPC UA client provides various methods for file access:

- **Open**

Opens a file for further file operations.

The access mode must be specified when opening a file:

- **1**: File is opened with read access.
- **2**: File is opened with write access.
- **3**: File is opened with read and write access.

The return value is a handle number that is required for all further file operations.

- **SetPosition**

Defines the file position from which a subsequent file operation is to be carried out.

- **GetPosition**

Specifies the current file position. If no other file position is defined, a subsequent file operation will be performed from this position.

- **Read**

Reads the part of a file starting at the current file position. The file position jumps forwards by the number of bytes read.

- **Write**

Writes the part of a file starting at the current file position. The file position jumps forwards by the number of bytes written.

- **Close**

Closes an open file. On closing, the handle number becomes invalid.

Precondition

- Read: User group OpcUaObserver
- Write: User group OpcUaOperator

Procedure

1. Open the file to read or write.
 - a. Right-click on **Open** and call the method with **Call...**
 - b. Transfer the desired access mode to the method and click on **Call**.
The method provides the handle number for further file operations.
 - c. Close the method.
2. If necessary, define the file position from which a subsequent file operation is to be carried out with **SetPosition**.
3. Read and/or write the file with **Read** and/or **Write**.
4. Repeat steps 2 and 3 as often as required.
5. Close the file.

5.5 Messages and events



These data are available with the following option packages:

- KUKA.DeviceConnector Messages
- KUKA.DeviceConnector Advanced

Description

All messages and Windows event logs of the robot controller can be displayed in the **Event View** of the OPC UA client.

Events							
Events Alarms Event History							
A	C	Time	Severity	Server/Object	SourceName	Message	EventType
		09:21:11.452	100	KDCoffice / KRC	KRC	Block selection: BCO run required in T1/T2	NotificationConditionType
		09:21:11.452	100	KDCoffice / KRC	KRC	Automatic mode blocked.	NotificationConditionType
		09:22:48.309	800	KDCoffice / KRC	KRC	Drives off, intermediate circuit voltage still charged	NotificationConditionType
		09:21:11.549	100	KDCoffice / KRC	KRC	Drives not ready	NotificationConditionType
		09:22:48.309	100	KDCoffice / KRC	KRC	Drives off, intermediate circuit voltage still charged	NotificationConditionType
		09:37:59.075	500	KDCoffice / KRC	KRC	Object is not available	NotificationType
		09:38:13.263	500	KDCoffice / KRC	KRC	The logged-on user switched from Operator to A...	NotificationType

Details	
Name	Value
Identifier	1:51007155_KSS220
ConditionName	KSS220
EventId	len=16, 0xb61517255e0a66458d7d3821bbf40cb5
▼ EventType	ModelId
	NamespaceIndex 6
	IdentifierType Numeric
	Identifier 1001
	Message "en", "Drives off, intermediate circuit voltage still charged"
	Retain True
	Severity 800
	SourceName KRC

Fig. 5-7: Messages and events

The messages and events in the **Event View** are updated continuously. The message text is always English. Additionally, specific details are displayed for each message/event. For the error messages of the robot controller, for example, the error number or the Retain value:

- Retain = True when the event occurs
- Retain = False if the event is canceled, e.g. after error acknowledgement

Procedure

- To display the messages and events in the client, drag the KRC node to the **Configuration** area of an **Event View**.

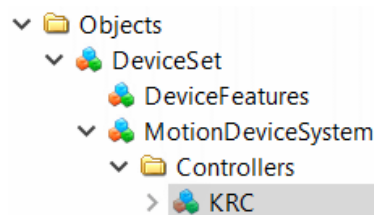


Fig. 5-8: KRC node

- To open an **Event View**, select **Document > Add...** in the Client menu and add the **document type Event View** with **Add**.

5.6 Process data



These data are available with the following option packages:

- KUKA.DeviceConnector SystemVars
The SystemVars option also contains the customer-specific variables (CustomerData).
- KUKA.DeviceConnector Advanced

Description

The OPC UA client can access all system variables and global KRL variables. The variables can be write-protected and only readable or both readable and writable.

All relevant directories are available under the **ProcessData** node.

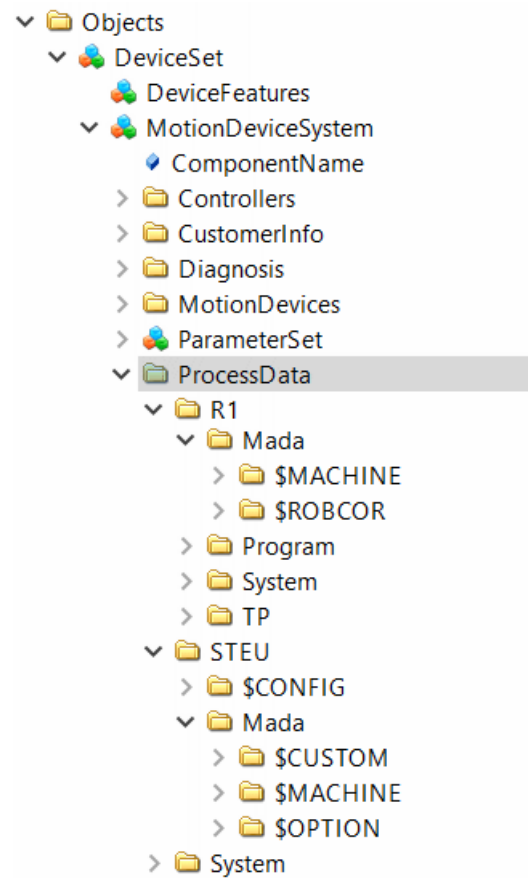


Fig. 5-9: Process data

Precondition

- Read: User group OpcUaObserver
- Write: User group OpcUaOperator

Procedure

Modifying a variable value:

- Select the variable node in the data tree and enter the new value directly under **Value** in the **Attributes** window.
- Or: Drag the variable node into a **Data Access View** and enter the new value under **Value**.

5.7 Customer-specific variables (CustomerData)



These data are available with the following option packages:

- KUKA.DeviceConnector CustomerData
- KUKA.DeviceConnector SystemVars
- KUKA.DeviceConnector Advanced

The OPC UA client can access all customer-specific variables. In the SystemVars and Advanced options, these variables are automatically enabled. If the CustomerData option is used, the files in which the customer-specific variables are defined must be stored in the following directory on the robot controller:

- ...\\KRC\\Roboter\\KRC\\R1\\CUSTOMER\\

The user must create the CUSTOMER folder on the robot controller.

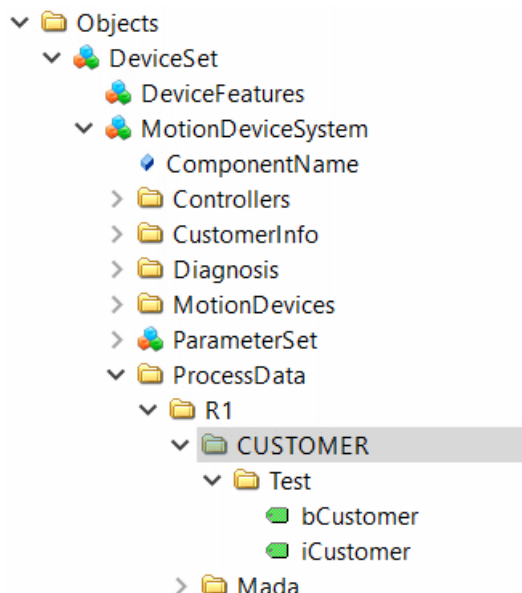


Fig. 5-10: CustomerData

5.8 Customer-specific information (CustomerInfo)



This functionality is available with the following option packages:

- KUKA.DeviceConnector Advanced

Description

The OPC UA client can save any amount of static information on the server. Using the OPC UA client, the user can create the variable nodes with the desired information and delete them again as required.

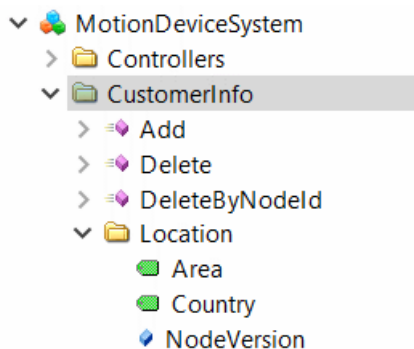


Fig. 5-11: Customer-specific information

The OPC UA client uses the following methods for this:

- **Add**
Method for adding a new node including possible hierarchy levels

- **Delete**
Method for deleting a node or hierarchy level using the browse name
- **DeleteByNodeId** (for advanced client users)
Method for deleting a node or hierarchy level using the OPC UA node ID

Variable values of existing nodes can be modified subsequently.

Precondition

- User group OpcUaOperator

Procedure

Adding a node:

1. Right-click on **Add** and call the method with **Call...**
2. Transfer the following input parameters to the method:
 - **Path:** Path to variable node
Using the path, any number of levels (separated by the "/" symbol) can be added under the CustomerInfo node.
If no path is specified, the variable node is created directly under the CustomerInfo node.
 - **Name:** Browse name of the variable
 - **Value:** Value of the variable
3. Select **Call**.
4. Close the method with **Close**.

Modifying a variable value:

- Select the variable node in the data tree and enter the new value directly under **Value** in the **Attributes** window.
- Or: Drag the variable node into a **Data Access View** and enter the new value under **Value**.

Deleting nodes:

1. Right-click on **Delete** and call the method with **Call...**
2. Transfer the input parameters **Path** and **Name** to the method:
 - In order to delete a specific variable node, enter the browse name of the variable and the corresponding path to the variable node (separated by the "/" symbol).
 - If the node to be deleted is located directly below the CustomerInfo node, it is not necessary to enter a path.
 - If only a path is entered, and no name, all variables and levels located below the level specified in **Path** are deleted.
3. Select **Call**.
4. Close the method with **Close**.

Deleting a node using the node ID:

1. Right-click on **DeleteByNodeId** and call the method with **Call...**
2. Transfer the node ID of the node that is to be deleted to the method as the input parameter:
 - "NamespaceIndex"
 - "IdentifierType"
 - "Identifier"
3. Select **Call**.
4. Close the method with **Close**.

5.9 Soft PLC data



These data are available with the following option packages:

- KUKA.DeviceConnector SoftPLC
- KUKA.DeviceConnector Advanced
- Precondition: KUKA.PLC ProConOS or KUKA.PLC ProConOS Ltd is installed.

Description

All Soft PLC data can be found under the PLC1 node. The corresponding global variables can be read and written.

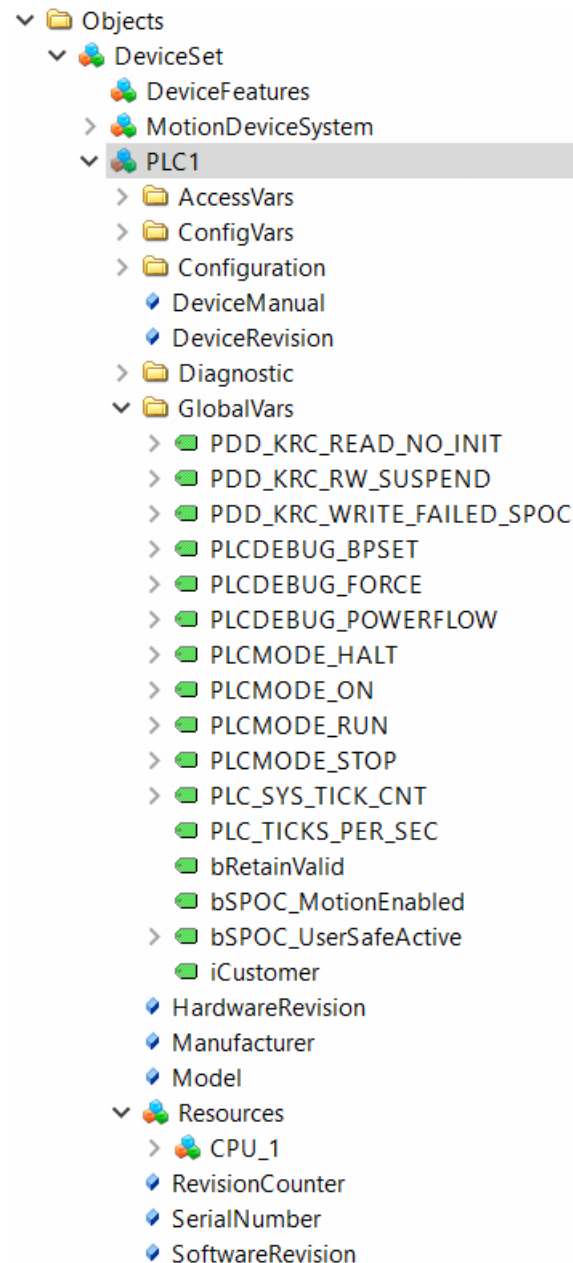


Fig. 5-12: Soft PLC data



Further information about the data can be found in the following document from the PLCopen and OPC Foundation:

- PLCopen OPC UA Information Model for IEC 61131-3

Precondition

- Read: User group OpcUaObserver
- Write: User group OpcUaOperator

Procedure**Modifying a variable value:**

- Select the variable node in the data tree and enter the new value directly under **Value** in the **Attributes** window.
- Or: Drag the variable node into a **Data Access View** and enter the new value under **Value**.

5.10 PROFINET data

These data are available with the following option packages:

- KUKA.DeviceConnector Advanced
- Precondition: KUKA.PROFINET® -/S or KUKA.PROFINET® M/S is installed.

The basic information about the PROFINET periphery can be found under the PROFINET® node, e.g.:

- Device name
- IP address
- I&M data
- Energy consumption

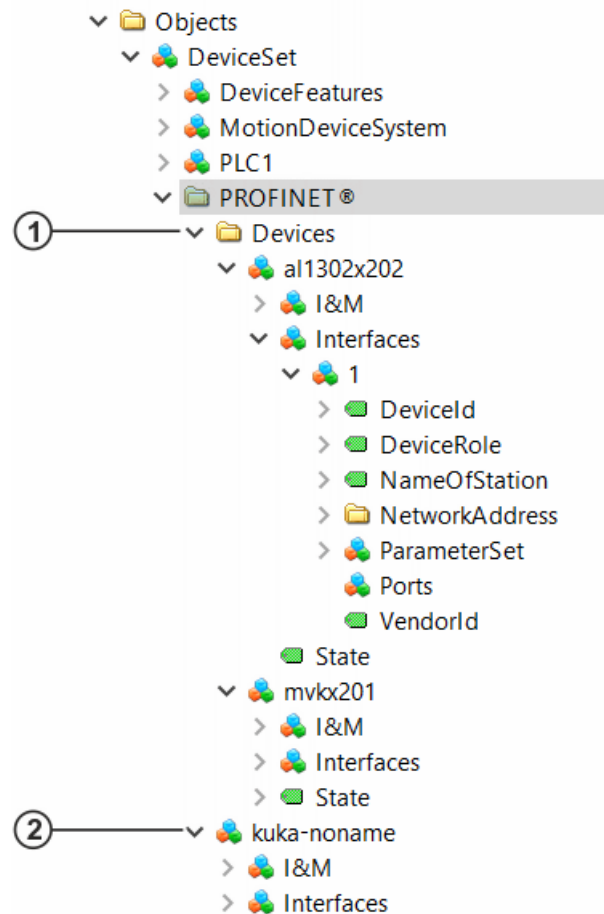


Fig. 5-13: PROFINET data

Item	Description
1	The connected PROFINET devices and their data are displayed under Devices .
2	This node is only available if the robot controller itself is configured as a PROFINET device.



Further information about the data can be found in the following documents from the OPC Foundation:

- OPC 30140: OPC UA for PROFINET Companion Specification
- OPC 30141: OPC UA for PROFINETenergy

5.11 Diagnostic data



These data are available with the following option packages:

- KUKA.DeviceConnector Advanced

The diagnostic data, i.e. all data of the diagnostic monitor and log events, can be found under the **Diagnosis** node. The diagnostic data can be read.

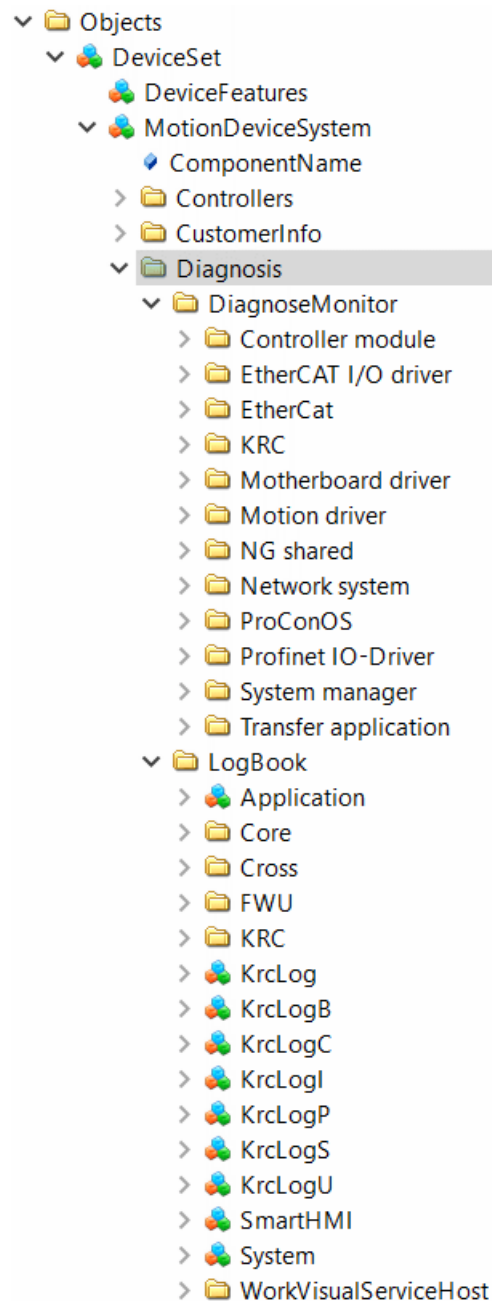


Fig. 5-14: Diagnostic data

5.12 OPC UA node IDs

The OPC UA nodes from the namespace <http://kuka.com/UA/Robotics/> have fixed node IDs that are linked to the browse path in the OPC UA tree, e.g.:

- 5100: Root/Objects/DeviceSet/MotionDeviceSystem/Controllers/KRC
- 5210: Root/Objects/DeviceSet/MotionDeviceSystem/Controllers/KRC/FileSystem/Config/User



A reference of all supported node IDs can be found on the data storage medium with the installation files of the DeviceConnector.

- Directory .\DOC\nodesets

6 Configuration

6.1 Changing the OPC UA user password

Description

The passwords of the OPC UA user groups can be changed via the Windows user management on the robot controller. There are no restrictions for the password, e.g. in terms of length or complexity.



If the changed password is lost, access to the Windows system is no longer possible – not even for KUKA.

Precondition

- Windows interface (**Minimize HMI**)
- Administrator privileges on the Windows user interface

Procedure

1. Press the Windows key, right-click on **Computer** and select **Manage**.
 2. In the tree structure, select **System Tools > Local Users and Groups > Users**.
 3. Select the desired user group, right-click on it and select **Set password**.
 4. Acknowledge the warning with **Proceed**.
 5. Define a new password and click on **OK**.
- The password is changed and can now be used.

6.2 Creating a self-signed certificate for authentication

Description

As an alternative to authentication with user name and password, the user can create a self-signed certificate. This enables the use of OPC UA clients that do not support logon with password.

Precondition

- An external program for creating certificates, e.g. XCA, is available.

Procedure

1. Open the program for creating certificates and start creation.
2. Enter data of the OPC UA user.
3. When defining the certificate as **OPC UA user authentication**, at least one of the two permissible role attributes must be specified:
 - Role: Opc:operator
 - Role: Opc:observer
4. Create certification key.

6.3 Configuring OPC UA process data

Description

In addition to the data predefined by KUKA, the user can publish application-specific process variables with the OPC UA server.

Publishable process variables are:

- System variables (\$ variables)
- Global KRL variables

A file template for configuration of the process variables is available following installation of the option package on the robot controller. The file follows a JSON syntax and contains a configuration example.

(>>> [9.1 "Configuration file "ProcessData.opcj" for OPC UA" Page 59](#))

File template

Directory	C:\KRC\ROBOTER\Config\User\Common\opcua
File	ProcessData.opcj

For configuration, the file template can be edited directly, or copied and edited and then saved in the aforementioned directory with a different user-defined name. Multiple opcj files can be saved in the directory.

The following is a description of the procedure for configuration using WorkVisual.



In the case of a software update, the process data configured in an opcj file are retained.

Precondition

- No project is open in WorkVisual.
- Network connection to the robot controller
- User group "Expert"
- T1 or T2 mode
- No program is selected.

Procedure

1. Load the active project from the robot controller.
2. Expand the file structure of the robot controller on the **Files** tab in the **Project structure** window.
3. Right-click on the desired file and open it in an editor via the context menu.
4. Edit the file as required and save it in the project.
5. Transfer the project back to the robot controller.



Information about procedures in WorkVisual is contained in the WorkVisual documentation.

6.3.1 DataSets

Description

A data set is a JSON object with the following properties:

Object	Use	Description
node	Required	Name of the root nodes of OPC UA Only "ProcessData" and "CustomerInfo" are allowed.
id	Required	Path of the folder structure (e.g. name1/ name2/name3) An empty ID is possible and corresponds to the root folder.
min_sample_time	Optional	The time that is required as standard to read all variables in the selected folder.
descr	Optional	Description of the data set that is also displayed in the information model.
data_points	Required	List of variables
auto_fill_source	Optional	(>>> 6.3.3 "auto_fill_source" Page 39)

Example

```
{
  "data_sets": [
    {
      "node": "ProcessData",
      "id": "Example",
      "min_sample_time": 1000,
      "descr": "A Commentary for the data set!",
      "data_points":
        [
        ]
    }
  ]
}
```

6.3.2 DataPoint

Description

A data point is a JSON object with the following properties:

Object	Use	Description
id	Required	Name of the OPC UA variable
min_sample_time	Optional	The time that is required as standard to read all variables in the selected folder.
descr	Optional	Description of the data set that is also displayed in the information model.
push	Optional	Indicates whether a KRL variable is TRUE or FALSE

Object	Use	Description
data_type	Optional	Data type If no value is specified, the system automatically attempts to determine the data type. If this fails, the data type STRING is used. Permissible data types are: • INT • DOUBLE • BOOL • STRING • POSITION • VECTOR • FRAME • POS • E6POS • AXISPOS • LOAD
writeable	Optional	Indicates whether a variable is writable. Default: FALSE
url	Optional	URL for the description of the data source; e.g. <code>krc:\$mode_opt</code>. The following schemas are supported: <code>krl</code>, <code>data</code>, <code>wmi</code>, <code>reg</code>, <code>xml</code>, <code>krc.diag</code> As an alternative to URL, <code>sys_name</code> or <code>value</code> can also be specified.
sys_name	Optional	KRL variable; e.g. <code>url=krl:<sys_name></code> As an alternative to <code>sys_name</code>, URL or <code>value</code> can also be specified.
value	Optional	Constant value; e.g. <code>url=data:<value></code> As an alternative to <code>value</code>, <code>sys_name</code> or URL can also be specified.

Example

```

"data_points": [
  {
    "id": "ControllerUptime",
    // Fully variable system name (required).
    "sys_name": "$ROB_TIMER",
    "min_sample_time": 100,
    "push": false
  },
  {
    "id": "ConstValue",
    "value": "const"
  },
  {
    "id": "ControllerUptimeUri",
    "url": "krl:$ROB_TIMER"
  }
]

```

6.3.3 auto_fill_source

Description

The DataSet can be configured so that it is automatically filled with variables of the specific KRL paths or system variables.

Syntax

The format of the *auto_fill_source* is a URN syntax:

`urn:rkl:<path>?[s[ystem]|r[ecurisve]]`

Code	Description
urn:rkl	Schema and namespace identifier (NID); always required
path	Path to the KRL module or to the directory of the KRL modules Example: <code>\r1\system\config</code> <code>\r1\system*</code>
recursive (r) system (s)	Two possible parameters are allowed: - recursive (r): Uses the path as the basis for a reverse (recursive) search. All matching objects present in the search path will be found. - system (s): Attaches all system variables of the controller to the DataSet. Parameters are separated by a semicolon. Example: <code>urn:rkl:*?s;r</code> The search looks for all variables. <code>urn:rkl:\r1\system\config</code> The search looks for <code>config.dat</code> variables only. <code>urn:rkl:\r1\?r</code> The search looks for all recursive r1 variables.

Example

```
{
  "node": "ProcessData",
  "id": "KRL",
  "min_sample_time": 1000,
  "descr": "DataSet automatically filled with KRL variables.",
  "auto_fill_source": "urn:rkl:*?s;r"
}
```

6.4 Message Queuing Telemetry Transport (MQTT)

Description

The Message Queuing Telemetry Transport (MQTT) protocol is an open messaging protocol that enables communication with a cloud. Telemetry

data are exchanged between the devices and the cloud in the form of messages. The protocol allows robots to be connected to an MQTT broker.

MQTT enables connections to be established between the robots and the broker. These connections are possible across network boundaries. Data transmission is unidirectional. Ports 1883 and 8883 are reserved for MQTT.

File template

Directory	C:\KRC\ROBOTER\Config\User\Common\opcua
File	*.pscj

A complete example can be found in the appendix.

(>>> [9.2 "Example of a configuration file for MQTT" Page 60](#))

6.4.1 ConnectionParameters

Description

The ConnectionParameters are a key that describes the settings for the connection to the MQTT broker.

Object	Use	Description
Address	Required	A ConnectionParameter that specifies the protocol (mqtt or mqttts), the IP and the port of the broker. As seen in the example, the user name and password can also be specified in the URL. As an alternative to the direct entry of the address, a KRL reference can be specified. This is specified in the following form: KRL:*File Path*/*Variable Name*
PublisherId	Required	ID from Host
ConnectionName	Required	Placeholder which is substituted by the network name of the controller
Enabled	Optional	Starts the connection
EnabledKrlVar	Optional	Indicates the name of a KRL variable that contains the value for Enabled. The connection can be switched on and off via this variable.

Example

The target address of the MQTT broker must be specified. In this example, the host is filled with the placeholder %HOSTNAME%. However, the ID can be renamed in the practical implementation.

```
{
  "ConnectionParameter":
  {
    "Address": "mqttts://username:password<broker-ipaddress>:8883",
    "PublisherId": "%HOSTNAME%"
    "ConnectionName": "KukaConnect"
    "Enabled": true
  }
}
```

6.4.2 DataSets

Description

The DataSets are **keys** that define a list of the data to be regularly transmitted:

Object	Use	Description
Name	Required	Name of the PublishedDataSet object in the AddressSpace.
Enabled	Optional	Defines whether the DataSet is activated. A deactivated DataSet will be ignored.
Topic	Required	A unique name for the MQTT topic Each topic of a KRC-based system should begin with "KUKA/KRC/". This is followed by a unique identifier for the system. This can be the host name or the serial number (robot/cabinet). Example: "KUKA/KRC/%HOSTNAME%/Operating data". This parameter defines the queue name in the associated WriterGroup.
NetworkMessageContentMask	Optional	The NetworkMessageContentMask defines the format of the NetworkMessage. The value is defined as a combination of the individual flag names. The individual values can be separated by means of ',' or ' '. Spaces are ignored. Alternatively, the corresponding numeric value can also be used. Example: "NetworkMessageHeader DataSet-MessageHeader PublisherId".= "11". This parameter defines the MessageSettings.NetworkMessageContentMask in the associated WriterGroup.

Object	Use	Description
Type	Required	The type of the DataSet. Data elements for a list of variables. Events for events, Files for file contents. For Files, the following parameters are available: • FileName: name of the file to be transmitted The special characters ? and * are possible. • Directory: name of the OPC UA node in which the file is located. • HashAlgorithm: method used to generate a checksum via the file. Valid values are: MD5, SHA1, SHA256, SHA384, SHA512 • CompressStrategy: indicates whether the file contents are compressed for transfer. Permissible values: – Auto – Never – Always • ChunkSize: maximum number of bytes that can be transferred together. If the specified ChunkSize value is exceeded, the file is distributed over several MQTT blocks. • TransmissionInterval: minimum time between consecutive data transfers in milliseconds • TransmitOnlyChanges: when this value is set to TRUE, the file is then only transferred if the contents have changed since the expiration of the last TransmissionInterval. • TransmitAtLeastOnce: when this value is set to TRUE, this file is transferred at least once – even if it was already present during the installation of DeviceConnector.
PublishingInterval	Required	The PublishingInterval defines the interval for publication of NetworkMessages and the embedded DataSetMessages. The value is entered in milliseconds. This parameter defines the PublishingInterval in the associated WriterGroup.
DefaultSamplingInterval	Optional	The DefaultSamplingInterval sets the default value for the sampling interval of all the DataPoints in this DataSet. This value is only valid for DataItems.

Object	Use	Description
FieldContentMask	Optional	The FieldContentMask defines the format of an individual field in a DataSetMessage. The value is defined as a combination of the individual flag names. The individual values can be separated by means of ',' or ' '; spaces are ignored. Example: "StatusCode SourceTimestamp" or "RawData". Alternatively, the corresponding numeric value can also be used. "3" or "32". This parameter defines the DataSetFieldContentMask in the associated Writer.
KeyFrameCount	Required	The KeyFrameCount is the multiplier of the PublishingInterval. It defines the maximum number of PublishingIntervals that are to elapse before a KeyFrame message is sent. In the case of files, the KeyFrameCount determines after how many trigger intervals the file will nevertheless be transmitted even if the contents of the file have not changed. This parameter defines the KeyFrameCount in the associated Writer.
MessageContentMask	Optional	The MessageContentMask defines the format of the DataSetMessage header. The value is defined as a combination of the individual flag names. The individual values can be separated by means of ',' or ' '. Spaces are ignored. Alternatively, the corresponding numeric value can also be used. Example: "DataSetWriterId Sequence number"="5". This parameter defines the MessageSettings.DataSetMessageContentMask in the associated Writer.
DataPoints	Optional	A DataSet of the type DataItems defines a sequence of DataPoints. Each DataPoint describes a variable that is to be monitored.

Example

```

"DataSets":
[
// Definitions of the MQTT topics
{
  "Name": "device-metadata",
  "Enabled": true,
  "Topic": "KUKA/KRC/%HOSTNAME%/device-metadata",
  "NetworkMessageContentMask": "11",
  // Possible values for Type: DataItems, Events, Files
  "Type": "DataItems",
  "PublishingInterval": 5000,

```

```
"DefaultSamplingInterval": 1000,  
"FieldContentMask": "3",  
"KeyFrameCount": 12,  
"MessageContentMask": "12",  
"DataPoints":  
[  
]  
}
```

6.4.3 DataPoints

Description

DataPoints are components of the DataSets and the OPC UA nodes to be transmitted:

Object	Use	Description
PublishedName	Required	Name of the fields in the DataSetMessage. The name must be unique within the DataSet. The placeholders in the “Source” definition can lead to multiple variables. Placeholders may also be used in the PublishedName. “#”: will be replaced by a sequential number. May be used once as a placeholder. “{n}”: will be replaced in the “Source” definition by the nth placeholder. Use once for each “+” placeholder. The placeholders are counted upward from 1.
Source	Required	Source defines the variable to be included in the DataSet. The value is a BrowsePath. MotionDeviceSystem is defined as a root object. Only variables of objects below this root object can be addressed. The path to the variable can now be specified in full. Example: <i>MotionDevices/Robot/Axes/A1/ParameterSet/ActualPosition</i>. Within this path, individual directories can also be replaced by “+” as a placeholder. During evaluation, the placeholder is replaced by all names for which the path can be continued to the end. A MotionDevice system has an “Axes” folder. The folder contains definitions for all axes of the machine. If the current position of all axes is to be adopted in the DataSet, the following BrowsePath is sufficient: <i>MotionDevices/Robot/Axes/+/ParameterSet/ActualPosition</i>. A variable is added for each axis, regardless of how many axes the robot has. If all the axes are to be selected for all the machines, you can also use a placeholder for the machine name: <i>MotionDevices/+/Axes/+/ParameterSet/ActualPosition</i>. If placeholders are used in the source, placeholders must also be inserted in the PublishedName in order to obtain unique designations.

Example

```

"DataPoints":
[
{
  "PublishedName": "RobotName_{1}",
  "Source": "MotionDevices/+/Name",
  "SamplingInterval": 1500
}
]

```

```
} ,  
]
```

7 Troubleshooting

7.1 No connection to the OPC UA server

Error

The OPC UA client can no longer establish the connection to the OPC UA server on the robot controller.

Cause

Port 4840 of the KLI or KONI for OPC UA communication has been deactivated, e.g. by activating an earlier version of a WorkVisual project on the robot controller after installation of KUKA.DeviceConnector.

Solution

In order to be able to use the currently active project for OPC UA communication, the port must be enabled. The project must then be activated on the robot controller again via WorkVisual.

Precondition

- User group "Expert"
- Operating mode T1 or T2.
- No program is selected.

Procedure

1. In the KLI or KONI network configuration, check whether port 4840 is present in the NAT list; if not, enable the port:
 - a. In the main menu, select **Start-up > Network configuration**. The **Network configuration** window opens.
 - b. Press **Advanced....** The window for advanced network configuration opens.
 - c. Select the **NAT** tab. A list of all the shared ports of the Windows interface is displayed in the **Available ports:** area.
 - d. If the port is to be enabled:
 - i. Press **Add port**. A new port with the number "0" is added to the list.
 - ii. Enter 4840 in the **Port number:** box.
 - iii. Select **tcp** in the **Permitted protocols:** box.
 - iv. Press **Save**.

A maximum total of 40 ports can be enabled.
 - e. Close the **Network configuration** window using the Close icon.
 - f. Only if modifications have been made:

Reboot the robot controller so that the changes take effect. To do so, if PROFINET is used, select **Shutdown** in the main menu and select the option **Reload files**.
2. Load the active project from the robot controller in WorkVisual.
3. Transfer the project back from WorkVisual to the robot controller and activate it.

7.2 MQTT connection error

7.2.1 Incorrect target address

Error

No connection can be established to the MQTT broker.

Cause

A typical cause of an error in establishing the connection is an incorrect address.

Solution

The address under `ConnectionParameter` must match the target address of the MQTT broker.



The address entered must match the target address. This means, for example, that the encryption parameters must be correct, e.g. `mqtt://<IP address>` or `mqtts://<IP address>`.

Precondition

- User group "Expert"
- Operating mode T1 or T2.
- No program is selected.

Procedure

1. Check in the code whether the address under `ConnectionParameter` matches the target address of the MQTT broker.

```
{
  "ConnectionParameter":
  {
    "Address": "mqtts://192.168.46.158:8883",
    "PublisherId": "%HOSTNAME%"
  }
}
```

2. Correct the target address.
3. Load the active project from the robot controller in WorkVisual.
4. Transfer the project back from WorkVisual to the robot controller and activate it.

7.2.2 Invalid certificate

Error

No connection can be established to the MQTT broker.

Cause

One possible cause is the use of an invalid SSL certificate (e.g. because it has expired) by the MQTT broker. The problem also occurs if the communication protocol is not correct, e.g. `mqtt://<IP address>` instead of `mqtts://<IP address>`.

Solution



A valid SSL certificate must be issued for the MQTT broker.

The address entered must match the target address. This means, for example, that the encryption parameters must be correct, e.g. mqtt://<IP address> or mqtts://<IP address>.

Precondition

- User group "Expert"
- Operating mode T1 or T2.
- No program is selected.



Information on issuing a valid SSL certificate can be found in the documentation of the MQTT broker.

8 Messages

8.1 Information about the messages

The “Messages” chapter contains selected messages. It does not cover all the messages displayed in the message window.

8.2 System messages from module: DeviceConnector

8.2.1 DeviceConnector 00001

Message code	DeviceConnector 00001
Message text	Unable to establish connection {0} to {1}.
Message type	Status message
	
Possible cause(s)	Cause: URL address of the MQTT broker is incorrect (>>> Page 51) Solution: Adjusting the address of the MQTT broker via the variable display (>>> Page 51) Cause: URL address of the MQTT broker is incorrect (>>> Page 52) Solution: Adjusting the address of the MQTT broker via the configuration file (>>> Page 52)

Cause: URL address of the MQTT broker is incorrect

Description

If the URL address of the MQTT broker is specified incorrectly, it is not possible to establish a connection between the robot controller and the cloud.

Solution: Adjusting the address of the MQTT broker via the variable display

Description

The connection to the cloud is established via an MQTT broker. The following variables must be configured for this purpose (via variable display on KUKA smartHMI):

Variable	Description
KUKACONNECT_BROKERIP	Enter IP address of the MQTT broker.
KUKACONNECT_ENABLED	Set to TRUE to activate MQTT.

If the address to the MQTT cannot be established, it is essential to check whether the set variable values are correct.

Precondition

The following user rights are required for modifying a variable:

- KSS 8.5 or higher: Function group **General configuration**
- KSS 8.3: User group "Expert"
- VSS: User group "User"

Procedure

1. In the main menu, select **Display > Variable > Single**.
The **Variable display – Single** window opens.
2. Enter the name of the variable in the **Name** box and confirm with the Enter key.
3. The current value of the variable is displayed in the **Current value** box. If nothing is displayed, no value has yet been assigned to the variable.
4. To modify the variable: Enter the desired value in the **New value** box.
5. Press the **Set value** button. The new value is displayed in the **Current value** box.

Cause: URL address of the MQTT broker is incorrect

Description

If the URL address of the MQTT broker is specified incorrectly, it is not possible to establish a connection between the robot controller and the cloud.

Solution: Adjusting the address of the MQTT broker via the configuration file

Description

The connection of the robot controller to the MQTT broker is configured via the configuration file. The connection settings can be defined in the configuration file via the following parameters:

Object	Use	Description
Address	Required	A ConnectionParameter that specifies the protocol (mqtt or mqttts), the IP and the port of the broker.
PublisherId	Required	ID of the host.
ConnectionName	Required	Placeholder which is substituted by the network name of the controller
Enabled	Optional	

If the address to the MQTT cannot be established, check whether the set connection settings are correct.

Configuration file

Directory	C:\KRC\Roboter\User\Common
File	*.pscj

Procedure

1. Open the configuration file.

2. Check whether the connection settings are correctly configured.

8.2.2 DeviceConnector 00003

Message code	DeviceConnector 00003
Message text	Unable to find KRL variable "{0}" ({1})
Message type	Status message
	
Possible cause(s)	Cause: Configuration file contains errors (>>> Page 53) Solution: Check configuration file (>>> Page 53)

Cause: Configuration file contains errors

The KRL variable cannot be found if the configuration file does not meet the following three conditions:

- The configuration file must be of the file type *.pscj.
- The file path of the KRL variable must be correctly specified in the configuration file.
- The URL for the file path of the KRL variables must be of the data type String.

Solution: Check configuration file

Procedure

1. Call the storage location of the configuration file: C:\KRC\Roboter\User\Common
2. Ensure that the file ends in *.pscj.
3. Open the configuration file.
4. Check the file path under **Address** and adjust if necessary.

8.2.3 DeviceConnector 00006

Message code	DeviceConnector 00006
Message text	Failed to read file "{0}".
Message type	Notification message
	
Possible cause(s)	Cause: Configuration file contains errors (>>> Page 53) Solution: Check configuration file (>>> Page 54)

Cause: Configuration file contains errors

The KRL variable cannot be found if the configuration file does not meet the following three conditions:

- The configuration file must be of the file type *.pscj.
- The file path of the KRL variable must be correctly specified in the configuration file.
- The URL for the file path of the KRL variables must be of the data type String.

Solution: Check configuration file

Procedure

1. Call the storage location of the configuration file: C:\KRC\Roboter\User\Common
2. Ensure that the file ends in *.pscj.
3. Open the configuration file.
4. Check the file path under **Address** and adjust if necessary.

8.2.4 DeviceConnector 00007

Message code	DeviceConnector 00007
Message text	Unexpected error in "{0}"
Message type	Notification message
	
Possible cause(s)	Cause: Unexpected error (>>> Page 54) Solution: Contact KUKA Service (>>> Page 54)

Cause: Unexpected error

If the OPC UA or MQTT communication breaks off during the restart of the controller, an unexpected error message is displayed.

Solution: Contact KUKA Service

8.2.5 DeviceConnector 00008

Message code	DeviceConnector 00008
Message text	The license file {0} is no longer valid.
Message type	Notification message
	
Possible cause(s)	Cause: License file invalid (>>> Page 55) Solution: Installing new license file (>>> Page 55)

Cause: License file invalid**Description**

If a license file is not valid, the required data cannot be called up by the device used. This occurs when an incorrect license file is used, when the duration of validity for a license file has expired or when the license file and the serial number of the device do not match.

Solution: Installing new license file**Description**

License files offer the possibility of adding functions and additional data points to the scope of KUKA.DeviceConnector.

License files are uniquely assigned to devices via the device serial number.

To allow for the license file to be used, it must be stored in the correct directory after it is acquired and the system must be reconfigured.

Precondition

- User rights: Function group **General configuration**
- T1 or T2 mode
- No program is selected.

Procedure

1. Apply for the license file at DigitalServices.Robotics.De@kuka.com.
2. Copy the license file to C:/KRC/ROBOTER/config/user/common/opcuaccessconfig/.
3. Reconfigure the I/O driver
Further information can be found in the corresponding chapter in the system software documentation.

8.2.6 DeviceConnector 00009

Message code	DeviceConnector 00009
Message text	The license file {0} is invalid.
Message type	Notification message
	
Possible cause(s)	Cause: License file invalid (>>> Page 55) Solution: Installing new license file (>>> Page 56)

Cause: License file invalid**Description**

If a license file is not valid, the required data cannot be called up by the device used. This occurs when an incorrect license file is used, when the duration of validity for a license file has expired or when the license file and the serial number of the device do not match.

Solution: Installing new license file

Description

License files offer the possibility of adding functions and additional data points to the scope of KUKA.DeviceConnector.

License files are uniquely assigned to devices via the device serial number.

To allow for the license file to be used, it must be stored in the correct directory after it is acquired and the system must be reconfigured.

Precondition

- User rights: Function group **General configuration**
- T1 or T2 mode
- No program is selected.

Procedure

1. Apply for the license file at DigitalServices.Robotics.De@kuka.com.
2. Copy the license file to C:/KRC/ROBOTER/config/user/common/opcua/accessconfig/.
3. Reconfigure the I/O driver
Further information can be found in the corresponding chapter in the system software documentation.

8.2.7 DeviceConnector 00010

Message code	DeviceConnector 00010
Message text	The license file {0} is invalid. Incorrect serial number.
Message type	Notification message
	
Possible cause(s)	Cause: License file invalid (>>> Page 56) Solution: Installing new license file (>>> Page 56)

Cause: License file invalid

Description

If a license file is not valid, the required data cannot be called up by the device used. This occurs when an incorrect license file is used, when the duration of validity for a license file has expired or when the license file and the serial number of the device do not match.

Solution: Installing new license file

Description

License files offer the possibility of adding functions and additional data points to the scope of KUKA.DeviceConnector.

License files are uniquely assigned to devices via the device serial number.

To allow for the license file to be used, it must be stored in the correct directory after it is acquired and the system must be reconfigured.

Precondition

- User rights: Function group **General configuration**
- T1 or T2 mode
- No program is selected.

Procedure

1. Apply for the license file at DigitalServices.Robotics.De@kuka.com.
2. Copy the license file to C:/KRC/ROBOTER/config/user/common/opcua/accessconfig/.
3. Reconfigure the I/O driver
Further information can be found in the corresponding chapter in the system software documentation.

9 Appendix

9.1 Configuration file "ProcessData.opcj" for OPC UA

```
{
  "data_sets": [
    /* // Uncomment here.
    {
      // Name of the parent node in the OPC-UA structure.
      "node": "ProcessData",

      // Hierarchical path for the child node to be generated.
      "id": "Example",

      // Desired sampling rate applied to all variables in the data set
      // (supported values >= 100)
      "min_sample_time": 1000,

      "descr": "A Commentary for the data set!",

      // List of the system variables.
      "data_points": [
        {
          // Alias (mandatory).
          "id": "ControllerUptime",
          // Complete variable system name (mandatory).
          "sys_name": "$ROB_TIMER",

          // Desired sampling rate applied to a single variable
          // (optional, supported values >= 100).
          "min_sample_time": 100,

          // Mode of value update between OPC UA server and KSS.
          // Push=false makes OPC UA server synchronously read variable values
          // from controller via cyclic ShowVar calls
          // (default and recommended setting).
          // Push=true makes OPC UA server subscribe to acyclic SetInfo events
          // (recommended if user is familiar with
          // behaviour of controller's Cross interface).
          "push": false,

          // Indicates whether the variable may be written by client
          "writeable": false
        }
      ]
    },
    {
      "node": "ProcessData",
      "id": "Special/SpecialData",
      "min_sample_time": 1000,
      "descr": "A commentary for the data set!",
      "data_points": [
        {
          "id": "ModeOP",
          "sys_name": "$MODE_OP",
          "min_sample_time": 100,
          "push": true,
          "writeable": false
        }
      ]
    }
  ]
}
```

```

    ]
  }*/ //Uncomment here.
]
}

```

9.2 Example of a configuration file for MQTT

```

{
  "ConnectionParameter":
  {
    "Address": "mqtt://<broker-ipaddress>:8883",
    "PublisherId": "%HOSTNAME%",
    "ConnectionName": "KUKAConnect",
    "Enabled": true
  },
  "DataSets":
  [
    // Definitions of the MQTT topics
    {
      "Name": "device-metadata",
      "Enabled": true,
      "Topic": "KUKA/KRC/%HOSTNAME%/device-metadata",
      "NetworkMessageContentMask": "11",
      // Possible values for Type: DataItems,
      Events, Files
      "Type": "DataItems",
      "PublishingInterval": 5000,
      "DefaultSamplingInterval": 1000,
      "FieldContentMask": "3",
      "KeyFrameCount": 12,
      "MessageContentMask": "12",
      "DataPoints":
      [
        {
          "PublishedName": "RobotName_{1}",
          "Source": "MotionDevices/+/Name",
          "SamplingInterval": 1500
        },
        {
          "PublishedName": "RobotName_{1}",
          "Source": "Controllers/+/Name",
          "SamplingInterval": 1500
        }
      ]
    },
    {
      "Name": "operational-data",
      "Enabled": true,
      "Topic": "KUKA/KRC/%HOSTNAME%/operational-data",
      "NetworkMessageContentMask": "11",
      "Type": "DataItems",
      "PublishingInterval": 5000,
      "DefaultSamplingInterval": 1000,
      "FieldContentMask": "3",
      "KeyFrameCount": 12,
      "MessageContentMask": "12",
      "DataPoints":
      [
        {

```

```
    "PublishedName": "ActualPosition_{1}_{2}",
    "Source": "MotionDevices/+/Axes/+/ParameterSet/
    ActualPosition"
  },
  {
    "PublishedName": "ModeOp",
    "Source": "SafetyStates/SafetyState_1/
    ParameterSet/OperationalMode",
    "SamplingInterval": 1500
  },
  {
    "PublishedName": "AccuState",
    "Source": "ProcessData/SYSTEM/$ACCU_STATE"
  }
]
},
{
  "Name": "system-messages",
  "Topic": "KUKA/KRC/%HOSTNAME%/system-messages",
  "Type": "Events",
  "PublishingInterval": 1000,
  "KeyFrameCount": 0
}
]
```


10 KUKA Service

10.1 Requesting support

Introduction

This documentation provides information on operation and operator control, and provides assistance with troubleshooting. For further assistance, please contact your local KUKA subsidiary.

Information

The following information is required for processing a support request:

- Description of the problem, including information about the duration and frequency of the fault
- The greatest possible amount of information about the hardware and software components of the overall system

The following list gives an indication of the information which is relevant in many cases:

- Model and serial number of the kinematic system, e.g. the manipulator
- Model and serial number of the controller
- Model and serial number of the energy supply system
- Designation and version of the system software
- Designations and versions of other software components or modifications
- System Software diagnosis package
Additionally for KUKA Sunrise: Existing projects including applications
For versions of KUKA System Software older than V8: Archive of the software (Diagnosis package is not yet available here.)
- Application used
- External axes used

10.2 KUKA Customer Support

The contact details of the local subsidiaries can be found at:
www.kuka.com/customer-service-contacts

Index

A

Appendix.....	59
auto_fill_source.....	39

C

Compatibility.....	15
Configuration.....	35
Connection error, MQTT.....	48
ConnectionParameters, MQTT.....	40
Customer-specific information.....	29
Customer-specific variables.....	28
CustomerData.....	28
CustomerInfo.....	29

D

DataPoint.....	37
DataPoints, MQTT.....	44
DataSets.....	36
DataSets, MQTT.....	41
Diagnosis package.....	63
Diagnostic data.....	33
Documentation, industrial robot.....	5

E

Events.....	26
-------------	----

F

File transfer.....	24
Functions.....	9

I

Incorrect target address, MQTT.....	48
Installation.....	15
via smartHMI.....	15
via WorkVisual.....	17
Intended use.....	11
Introduction.....	5
Invalid certificate, MQTT.....	48
IP.....	6

K

KLI.....	6
Knowledge, required.....	5
KONI.....	6
KUKA Customer Support.....	63
KUKA Service.....	63

L

Licenses.....	7
---------------	---

M

Message Queuing Telemetry Transport.....	39
Messages.....	26, 51
Misuse.....	11
MQTT.....	6, 39
MQTT, connection error.....	48
MQTT, Connection Parameters.....	40
MQTT, DataPoints.....	44
MQTT, DataSets.....	41

O

OPC UA.....	6
OPC UA client, connecting to server.....	19
OPC UA nodes, node IDs.....	34
OPC UA, changing password.....	35
OPC UA, user groups.....	19
Open source.....	7
Operation.....	19

P

Process data.....	27
Process data, configuring.....	36
Product description.....	9
PROFINET, data.....	32

S

Safety.....	13
Safety instructions.....	5
Self-signed certificate, creating.....	35
Single point of control.....	13
Soft PLC, data.....	31
Software.....	15
SPOC.....	6, 13
Support request.....	63
System requirements.....	15
Hardware.....	15

T

Target group.....	5
TCP.....	6
Terms used.....	6
Trademarks.....	6
Training.....	5
Troubleshooting.....	47

U

Uninstalling	
via smartHMI.....	16
via WorkVisual.....	18
Updating	
via smartHMI.....	15

via WorkVisual.....	17
User groups, OPC UA.....	19

V

Variants.....	9
VDMA standard.....	20

W

Warnings.....	5
---------------	---