



Manual

# OPC UA

Version 1 (2023-04-21)  
EN



Service hotline tel. +49 (0) 74 61/92 86-8 63

Service hotline fax +49 (0) 74 61/92 86-8 69

Service hotline e-mail: [service@leibinger-group.com](mailto:service@leibinger-group.com)

Translation of the original instructions

All rights, particularly those relating to translation, publication and reproduction, are the sole property of Paul Leibinger GmbH & Co. KG. This document may not be processed or saved in part or in whole in any form, e.g. transcribed, photocopied, microfilmed or reproduced by any other method, without the written consent of Paul Leibinger GmbH & Co. KG, nor may it be made available to third parties.

Paul Leibinger GmbH & Co. KG reserves the right to any changes of the scope of delivery with regards to the technology, hardware and software as well as the associated materials, e.g. manuals and operating instructions.

# Contents

<b>1</b>	<b>Description .....</b>	<b>5</b>
<b>2</b>	<b>Making the connection between server and client.....</b>	<b>6</b>
2.1	Creating data sources: OPC UA .....	6
2.2	OPC UA certificates .....	8
<b>3</b>	<b>Address field .....</b>	<b>9</b>
3.1	Start up/shut down printer .....	9
3.1.1	WakeUp .....	9
3.1.2	ShutDown .....	9
3.2	Start/stop printing .....	9
3.2.1	StartPrinting.....	9
3.2.2	PausePrinting.....	9
3.2.3	PrintingState .....	10
3.3	Nozzle sealing system (nozzle object).....	10
3.3.1	OpenNozzle.....	10
3.3.2	CloseNozzle.....	10
3.3.3	NozzleState .....	10
3.3.4	OpenProgress .....	11
3.4	Fill levels.....	11
3.4.1	InkLevel.....	11
3.4.2	SolventLevel.....	11
3.5	Product speed (ProductionSpeed object) .....	11
3.5.1	MicroMeterPerSecond .....	11
3.5.2	ProductsPerMinute.....	11
3.6	Product counter (counters - object) .....	12
3.6.1	Counter1Value.....	12
3.6.2	Counter2Value .....	12
3.6.3	Counter1EndValue.....	12
3.7	Jobs, job lists .....	13
3.7.1	DriveList.....	13
3.7.2	GetJobs .....	13
3.7.3	GetJobLists.....	14
3.7.4	GetExternSelects.....	14
3.7.5	ExternOrJobListName .....	15
3.7.6	JobIndex.....	15
3.8	Messages (messages object).....	16
3.8.1	GetMessages .....	16
3.8.2	AcknowledgeAllMessages.....	16
3.8.3	ActiveError .....	17
3.9	The compatibility object (compatibility).....	18
3.9.1	Messages (ActiveError object).....	18
3.9.2	Counter1EndValueDouble .....	18
3.9.3	Counter1ValueDouble .....	19
3.9.4	Counter2ValueDouble .....	19
3.9.5	TriggerAcknowledgeAllMessages .....	19
3.9.6	TriggerCloseNozzle .....	19

---

3.9.7 TriggerOpenNozzle .....	19
3.9.8 TriggerPausePrinting .....	19
3.9.9 TriggerShutdown .....	20
3.9.10 TriggerStartPrinting .....	20
3.9.11 TriggerWakeup .....	20

# 1 Description

The **OPC UA** communication interface enables platform-independent transmission of any device data and parameters. Device data can be measurement values, sensor data, control variables or control instructions.

## 2 Making the connection between server and client

- ▶ Make sure that the printer (server) and the affected computer (client) are connected to the network.
  - ▶ *Create an OPC UA data source on the printer [▶ 6].*
  - ▶ Enter the printer connection settings on the client.
  - ▶ Create a connection to the printer on the client.
  - ▶ Accept the printer's certificate on the client.
  - ▶ Check whether the client's certificate is shown on the printer.
  - ▶ *Accept the client's certificate on the printer [▶ 8].*
- ⇒ The connection is set up.

### 2.1 Creating data sources: OPC UA

Condition:

- ✓ The network connection to the OPC UA client is available.



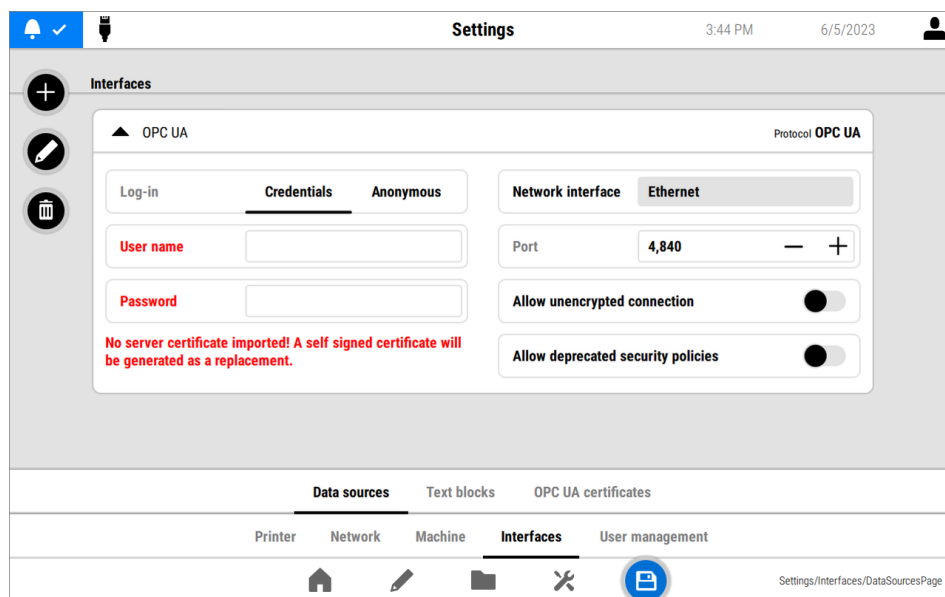
- ▶ Press the **Add** icon.

⇒ The **Name** dialog box appears.

- ▶ Enter a suitable name for the interface in the **Name** field. Under name, enter if possible the external source to prevent mix-ups with other data sources (e.g. ERP system, PLC, etc.).

- ▶ Press **OK**.

⇒ The area with the settings for the OPC-UA interface appears.



- ▶ Enter the desired settings. Observe the following detailed descriptions.



- ▶ Press the **Save** icon in the menu bar to save the settings.
- ▶ Press **OPC UA certificates**.

- ▶ Check whether the printer has generated a server certificate in the **Server certificate status** area.
- ⇒ The data source has been created. You can now use the data source within text blocks.

OPC UA interface settings:

---

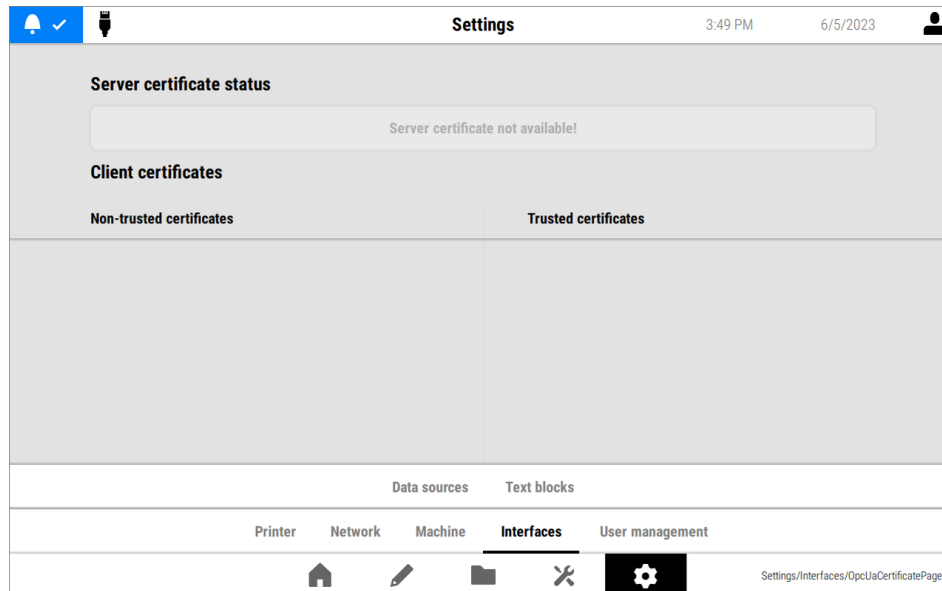
<b>Log-in</b>	Select login options for the server. Credentials: <ul style="list-style-type: none"><li>– The connection can only be established with specified access data.</li><li>– Enter <b>User name</b> and <b>Password</b>.</li></ul> Anonymous: <ul style="list-style-type: none"><li>– Anonymous access is allowed.</li><li>– A <b>User name</b> and <b>Password</b> are not required.</li></ul>
<b>User name</b>	Enter the username for logging in to the server.
<b>Password</b>	Password for server authentication.
<b>Network interface</b>	Select network interface for access to the server.
<b>Port</b>	Enter the port number for communication. Default port = 4840.
<b>Allow unencrypted connection</b>	Enable only if the client does not support encryption. Unencrypted connections do not require certificates, but the data can be read by third parties during transmission.
<b>Allow deprecated security policies</b>	The OPC-UA server offers the following security policies: <ul style="list-style-type: none"><li>– Basic256Sha256</li><li>– Basic128Rsa15 (outdated)</li><li>– Basic256 (outdated)</li></ul> Enable only if the client does not support Basic256Sha256 security policy. If active, Basic128Rsa15 and Basic256 security policies are allowed.

---

## 2.2 OPC UA certificates

The server checks whether or not the client certificate is trusted at the first connection attempt by the client:

- If the certificate is not trusted, the server rejects the client and writes the client certificate to the **Non-trusted certificates** area.
- If the certificate is trusted (CA), the server grants direct access to the client and writes the client certificate to the **Trusted certificates** area.



### Server certificate status:

#### Server certificate will be created upon save!

The process of creating a new certificate has been activated. If you want to cancel the process, exit the page and discard the changes.

#### Create self-signed server certificate

The printer creates a certificate for itself. The printer creates the certificate only after saving. The creation of a certificate takes about 30 seconds.

### Client certificates:

#### Non-trusted certificates

List of the untrusted certificates. Check the certificates in this list. If you want to allow a client to connect to the server, drag the client's certificate to the list of the trusted certificates.

#### Trusted certificates

List of the trusted certificates. All clients in this list can connect to the server.



## 3 Address field

### 3.1 Start up/shut down printer

#### 3.1.1 WakeUp

Wakes the printer up from sleep mode.

<b>NodeId</b>	Identifier	WakeUp
<b>NodeClass</b>		Method

#### 3.1.2 ShutDown

Shuts down the printer. The printer stays in sleep mode.

<b>NodeId</b>	Identifier	ShutDown
<b>NodeClass</b>		Method

### 3.2 Start/stop printing

#### 3.2.1 StartPrinting

Starts the printer. The printer starts to print as soon as it receives a PrintGo signal.

<b>NodeId</b>	Identifier	StartPrinting
<b>NodeClass</b>		Method

#### 3.2.2 PausePrinting

Pauses the printing. The printer ignores incoming PrintGo signals. The nozzle sealing system remains open.

<b>NodeId</b>	Identifier	PausePrinting
<b>NodeClass</b>		Method

### 3.2.3 PrintingState

Contains the current print status.

<b>NodeId</b>	Identifier	PrintingState
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	7[UInt32]

The variable values mean the following:

0	Nozzle sealing system is closed
1	Nozzle sealing system closes
2	Nozzle sealing system is open; no printing active
3	Nozzle sealing system opens
4	Nozzle sealing system is open; printing in progress
5	Printer prepares to open the nozzle sealing system

## 3.3 Nozzle sealing system (nozzle object)

### 3.3.1 OpenNozzle

Opens the nozzle sealing system.

<b>NodeId</b>	Identifier	OpenNozzle
<b>NodeClass</b>		Method

### 3.3.2 CloseNozzle

Closes the nozzle sealing system.

<b>NodeId</b>	Identifier	CloseNozzle
<b>NodeClass</b>		Method

### 3.3.3 NozzleState

Contains the current status of the nozzle sealing system.

<b>NodeId</b>	Identifier	NozzleState
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	7[UInt32]

### 3.3.4 OpenProgress

Contains the position of the nozzle sealing system in percent.

0 = Nozzle sealing system is closed

100 = Nozzle sealing system is completely open

<b>NodeId</b>	Identifier	OpenProgress
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	6[Int32]

## 3.4 Fill levels

### 3.4.1 InkLevel

Contains the current fill level of the ink reservoir tank in percent.

<b>NodeId</b>	Identifier	InkLevel
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	6[Int32]

### 3.4.2 SolventLevel

Contains the current fill level of the solvent reservoir tank in percent.

<b>NodeId</b>	Identifier	SolventLevel
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	6[Int32]

## 3.5 Product speed (ProductionSpeed object)

### 3.5.1 MicroMeterPerSecond

Contains the current product speed in  $\mu\text{m}$  per second

<b>NodeId</b>	Identifier	MicroMeterPerSecond
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	6[Int32]

### 3.5.2 ProductsPerMinute

Contains the current product speed in products per minute

<b>NodeId</b>	Identifier	ProductsPerMinute
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	6[Int32]

## 3.6 Product counter (counters - object)

### 3.6.1 Counter1Value

Contains the current value of product counter 1. You can adjust the counter reading to the printer by overwriting this value.

<b>NodeId</b>	Identifier	Counter1Value
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	8[Int64]

### 3.6.2 Counter2Value

Contains the current value of product counter 2. You can adjust the counter reading to the printer by overwriting this value.

<b>NodeId</b>	Identifier	Counter2Value
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	8[Int64]

### 3.6.3 Counter1EndValue

Contains the stop value of product counter 1. You can adjust the stop value to the printer by overwriting this value.

<b>NodeId</b>	Identifier	Counter1EndValue
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	8[Int64]

## 3.7 Jobs, job lists

### 3.7.1 DriveList

Contains the list of drives that are available on the printer. The list is output in the form of an array. The values can be used to access the methods GetJobs, GetJobLists and GetExternSelects.

<b>NodeId</b>	Identifier	DriveList
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	12[String]

### 3.7.2 GetJobs

Provides the names of the jobs within the input directory path. With the **InputArguments** variable, you can transfer the directory path to the method. With the **OutputArguments** variable, the method outputs the list of the file names and their paths in the form of an array.

<b>NodeId</b>	Identifier	GetJobs
<b>NodeClass</b>		Method

#### InputArguments

Directory path in which the GetJobs method is to search for the files. If you do not input a directory path, the method will automatically search the internal printer storage.

<b>NodeId</b>	Identifier	50510
<b>NodeClass</b>		Variable

#### OutputArguments

List of files that have been found by the GetJobs method.

<b>NodeId</b>	Identifier	50246
<b>NodeClass</b>		Variable

### 3.7.3 GetJobLists

Provides the names of the job lists within the input directory path. With the **InputArguments** variable, you can transfer the directory path to the method. With the **OutputArguments** variable, the method outputs the list of the file names and their paths in the form of an array.

<b>NodeId</b>	Identifier	GetJobLists
<b>NodeClass</b>		Method

#### InputArguments

Directory path in which the GetJobs method is to search for the files. If you do not input a directory path, the method will automatically search the internal printer storage.

<b>NodeId</b>	Identifier	50248
<b>NodeClass</b>		Variable

#### OutputArguments

List of files that have been found by the GetJobs method.

<b>NodeId</b>	Identifier	50249
<b>NodeClass</b>		Variable

### 3.7.4 GetExternSelects

Provides the names of all jobs and job lists within the input directory path. With the **InputArguments** variable, you can transfer the directory path to the method. With the **OutputArguments** variable, the method outputs the list of the file names and their paths in the form of an array.

<b>NodeId</b>	Identifier	GetJobLists
<b>NodeClass</b>		Method

#### InputArguments

Directory path in which the GetJobs method is to search for the files. If you do not input a directory path, the method will automatically search the internal printer storage.

<b>NodeId</b>	Identifier	50248
<b>NodeClass</b>		Variable

#### OutputArguments

List of files that have been found by the GetJobs method.

<b>NodeId</b>	Identifier	50249
<b>NodeClass</b>		Variable

### 3.7.5 ExternOrJobListName

Contains the path and the name of the currently loaded job, job list or ExternSelect job list. You can load another job, job list or ExternSelect job list on the printer by overwriting the value.

<b>NodeId</b>	Identifier	ExternOrJobListName
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	12[String]

### 3.7.6 JobIndex

Contains the index of the currently loaded job within the job list. The index begins with 0. The first job has the index 0, and the second job the index 1.

You can load another job from the job list by overwriting the value.

<b>NodeId</b>	Identifier	ExternOrJobListName
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	6[Int32]

## 3.8 Messages (messages object)

### 3.8.1 GetMessages

Provides a list with the currently pending errors in the **JetFutureMessage** format. With the **InputArguments** variable, you can transfer the first error number to the method. If you do not know the first error number, you can input 0 as the error number. With the **OutputArguments** variable, the method outputs a list with a maximum of ten errors in the form of an array. If there are more than ten errors, the variable will provide the number of the next error for a new method access.

<b>NodeId</b>	Identifier	GetMessages
<b>NodeClass</b>		Method

#### InputArguments

Error number from which the method should start searching for more errors.

<b>NodeId</b>	Identifier	50257
<b>NodeClass</b>		Variable
<b>Value [0]</b>	Name	startId

#### OutputArguments

List of error messages that have been found by the method. In addition, the

<b>NodeId</b>	Identifier	50258
<b>NodeClass</b>		Variable
<b>Value [0]</b>	Name	messages
<b>Value [1]</b>	Name	nextId

### 3.8.2 AcknowledgeAllMessages

Acknowledges the currently pending messages.

<b>NodeId</b>	Identifier	AcknowledgeAllMessages
<b>NodeClass</b>		Method



### 3.8.3 ActiveError

Contains information on the most important of the currently pending error messages in the **JetFutureMessage** data type. If no error is currently pending, then there will be a zero in the variable.

<b>NodeId</b>	Identifier	ActiveError
<b>NodeClass</b>		Variable
<b>DataType</b>		JetFutureMessage
	Identifier	3002

The JetFutureMessage format:

Field	Type	Use
Timestamp	DateTime	Time at which the message was generated
ErrorNumber	UInt32	Error number
NumberExtension	Int32	Number as information on the error
StringExtension	String	String as information on the error

## 3.9 The compatibility object (compatibility)

For clients with a limited functional scope, the server will provide the compatibility object. Use the compatibility object if your client does not support any methods, any data types it has defined itself and no 64-bit integer numbers.

### 3.9.1 Messages (ActiveError object)

#### ActiveErrorErrorNumber

Contains the number of the most important of the currently pending errors.

<b>NodeId</b>	Identifier	ActiveErrorErrorNumber
<b>NodeClass</b>		Variable
<b>Data Type</b>	IdentifierType	Numeric
	Identifier	7[UInt32]

#### ActiveErrorNumberExtension

Contains the extended number of the most important of the currently pending errors.

<b>NodeId</b>	Identifier	ActiveErrorNumberExtension
<b>NodeClass</b>		Variable
<b>Data Type</b>	Identifier	7[UInt32]

#### ActiveErrorStringExtension

Contains the extended string of the most important of the currently pending errors.

<b>NodeId</b>	Identifier	ActiveErrorStringExtension
<b>NodeClass</b>		Variable
<b>Data Type</b>	Identifier	12[String]

#### ActiveErrorTimestamp

Contains the time stamp of the most important of the currently pending errors.

<b>NodeId</b>	Identifier	ActiveErrorTimestamp
<b>NodeClass</b>		Variable
<b>Data Type</b>	Identifier	13[DateTime]

### 3.9.2 Counter1EndValueDouble

Contains the start value of SignalProductCounter 1 in the **Double** data type.

<b>NodeId</b>	Identifier	Counter1EndValueDouble
<b>NodeClass</b>		Variable
<b>Data Type</b>	Identifier	11[Double]

### 3.9.3 Counter1ValueDouble

Contains the current value of product counter 1 in the **Double** data type.

<b>NodeId</b>	Identifier	Counter1ValueDouble
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	11[Double]

### 3.9.4 Counter2ValueDouble

Contains the current value of product counter 2 in the **Double** data type.

<b>NodeId</b>	Identifier	Counter2ValueDouble
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	11[Double]

### 3.9.5 TriggerAcknowledgeAllMessages

Signal change from 0 to 1 or from 1 to 0 acknowledges all pending error messages.

<b>NodeId</b>	Identifier	TriggerAcknowledgeAllMessages
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	1[Boolean]

### 3.9.6 TriggerCloseNozzle

Signal change from 0 to 1 or from 1 to 0 closes the nozzle sealing system.

<b>NodeId</b>	Identifier	TriggerCloseNozzle
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	1[Boolean]

### 3.9.7 TriggerOpenNozzle

Signal change from 0 to 1 or from 1 to 0 opens the nozzle sealing system.

<b>NodeId</b>	Identifier	TriggerOpenNozzle
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	1[Boolean]

### 3.9.8 TriggerPausePrinting

Signal change from 0 to 1 or from 1 to 0 pauses the printing.

<b>NodeId</b>	Identifier	TriggerPausePrinting
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	1[Boolean]

### 3.9.9 TriggerShutdown

Signal change from 0 to 1 or from 1 to 0 shuts down the printer.

<b>NodeId</b>	Identifier	TriggerShutdown
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	1[Boolean]

### 3.9.10 TriggerStartPrinting

Signal change from 0 to 1 or from 1 to 0 starts the printing.

<b>NodeId</b>	Identifier	TriggerStartPrinting
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	1[Boolean]

### 3.9.11 TriggerWakeup

Signal change from 0 to 1 or from 1 to 0 wakes up the printer.

<b>NodeId</b>	Identifier	TriggerWakeup
<b>NodeClass</b>		Variable
<b>DataType</b>	Identifier	1[Boolean]









Paul Leibinger GmbH & Co. KG  
Daimlerstr. 14  
78532 Tuttlingen  
Germany  
Phone: +49 74 61 / 92 86-0  
Fax: +49 74 61 / 92 86-199  
Email: [info@leibinger-group.com](mailto:info@leibinger-group.com)

