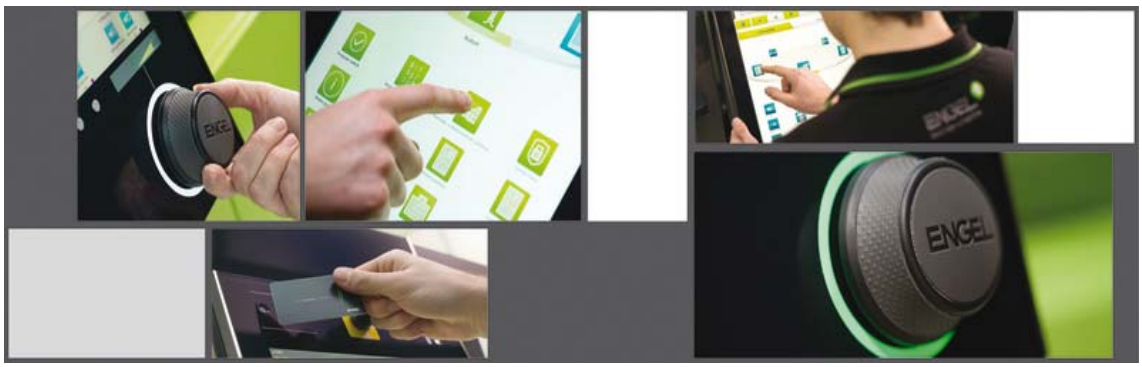
ENGEL



User information - EUROMAP 77

ENGEL

WARNING!

Improper use of the machine can lead to severe injury to persons and damage to equipment!
You must read this instruction manual carefully and familiarize yourself with the safety precautions before operating the system or carrying out any maintenance.
You must ensure that the persons carrying out the activities on the machine have read and understood the relevant sections and chapters of the instruction manual.

ENGEL AUSTRIA GmbH
A-4311 Schwertberg
Tel.: +43.50.620.0 | Fax: +43.50.620.3609
service@engel.at | http://www.engelglobal.com

Version: G/11/300/1/4

Printing date:**22.2.2022**
Language:Englisch

Translation of original instruction manual

Table of contents

ENGEL Data interface

Interface between injection molding machine and MES - Manufacturing Execution System.

Based on

- EUROMAP 77, Release 1.00a, 28. Januar 2019
- EUROMAP 83, Release 1.01, 28. Januar 2019

This document should provide a first overview about the availability of the specified functions in Eu-romap 77. For detailed information, please use the appropriate version of the EURO-MAP specifications.

They are available for download on the following listed pages:

- http://www.euromap.org/en/euromap77
- http://www.euromap.org/en/euromap83

The data interface is available for all machines with control CC300 from Software version V4.50.

Information

In order to use all the functions described in this document, a software update may be required.

1 EUROMAP 77 - OPC UA based communication

How is communication?
OPC UA was defined as the transmission technology for the EUROMAP 77 standard.
OPC UA is an industrial M2M communication protocol with a platform independent and service ori-ented architecture, which includes all individual OPC class specifications within one extendable framework. The first release happened 2008.
Further information is available online at www.opcfoundation.org.

What is available?
Within the scope of EUROMAP 77 and EUROMAP 83 information models were developed, which define the data, that is available and the provided functionalities.
EUROMAP 83 is herein the superior standard with general definitions of plastics and rubbery machines.

2 Functional overview

The following functions are available with the EUROMAP 77

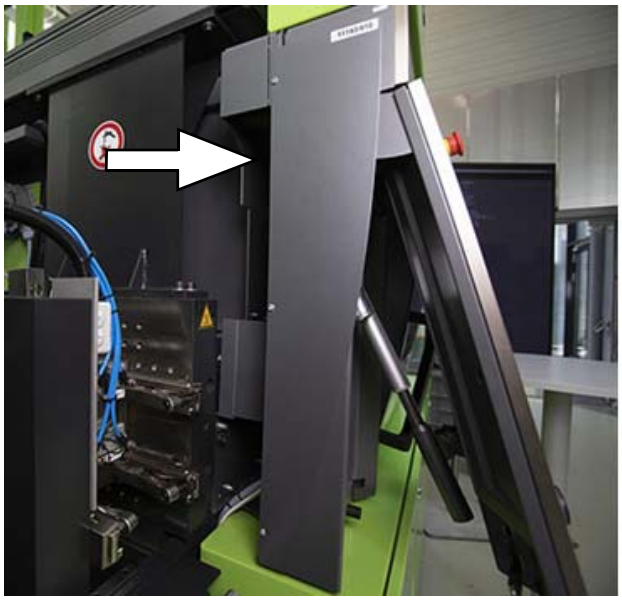
- General information about the machine (manufacturer, model, serial number…), current configuration and status of the machine including moulds, injection units and power units, and logbook of relevant changes on the machine.
- Order management:
  Production cycles and methods to send jobs from the MES to the machine and to release the production.
- Data set management:
  Storage of the settings in data sets. These also include information about process parameters (times, temperatures, pressures, etc.) of the machine, but also of integrated robot systems. EUROMAP 77 enables data set transfer between injection molding machines and MES for setting up a central memory for data sets.

Source: EUROMAP 77 Specification, Release 1.00a, 28 January 2019.

3 Network connection to the machine

A network connection is located at the rear of the monitor or as an option at the switch cabinet to connect the machine with the site network.
EUROMAP 77 requires a network connection to the machine (client-server communication via network).

Example: Network connection located at the rear of the screen of the injection molding machine.



Network configuration

The network configuration can be implemented from user level 11 in the service application. The following screen pages will take you to the network configuration:

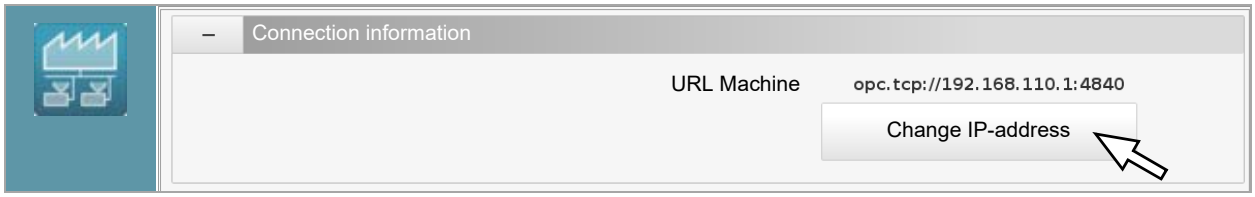
- **Production management system**
  Screen page is available from software version V4.82.
- **System administration**

Screen page

The screen page is below the following components and tasks:

Components	Tasks
Peripheral unit	Production

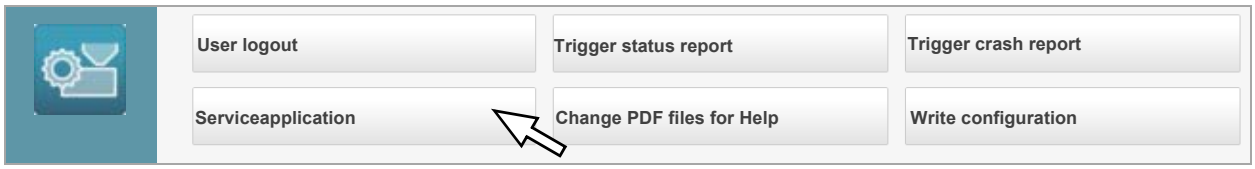
Configuration



Screen page

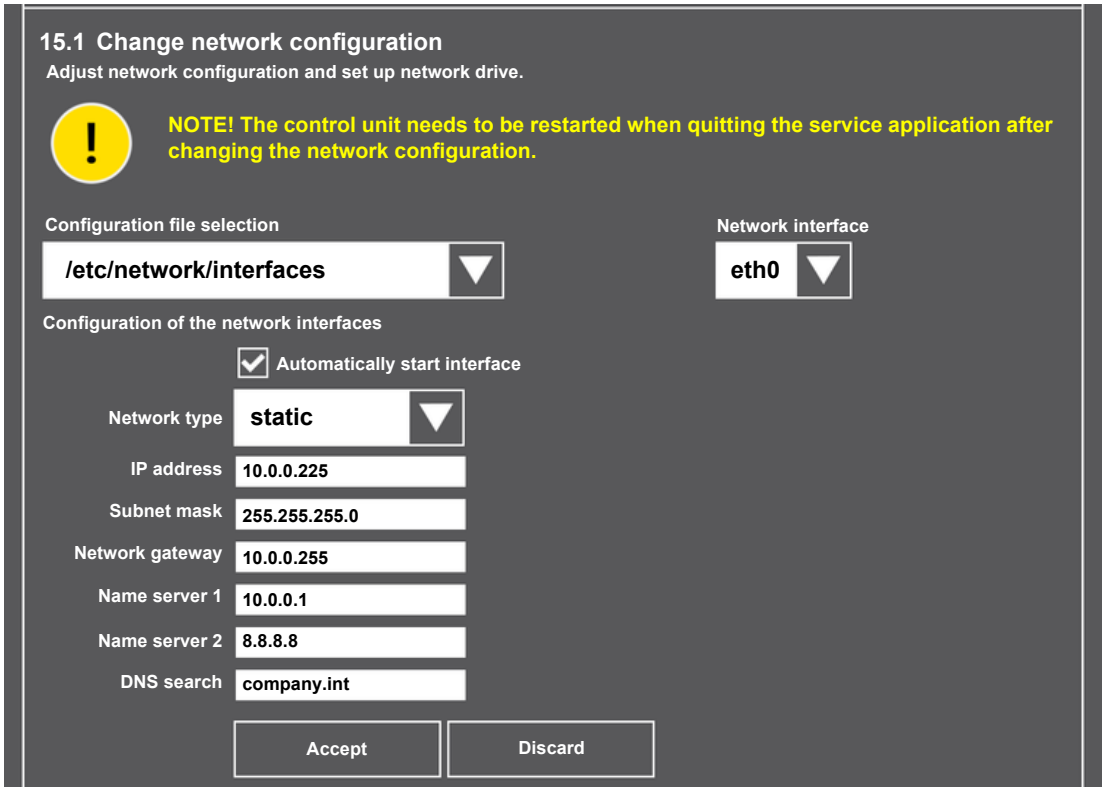
The screen page is below the following components and tasks:

Components	Tasks
	System



Serviceapplication - Change network configuration

Change



1. Select ’System Administration’ - ’Network’ - ’Change’.

   - **Configuration file selection**
     When selecting a configuration file, the selection option appears for network interfaces that were saved in the configuration file. These network interfaces can be edi-

ted. If you change the network settings, then the control unit creates a backup of the last settings. This backup can be used to restore data after an incorrect setting.

- **Network interface**
  Select 'eth0' as network interface. This is located on the rear of the screen or optionally on the control cabinet.
- **Automatically start the network interface**
  If the screen switch is ticked, the network interface is configured when the control unit starts. If the screen switch is not ticked, the network interface is not active.

2. Adjust network configuration.

   - **Network type**
     Select either 'static' (static IP address) or 'dhcp' (automatically assigned IP address). Also fill out the other fields in the 'static' setting.
   - **IP address**
     The IP address assigned to the control unit.
     Do not use the following internal control unit addresses:

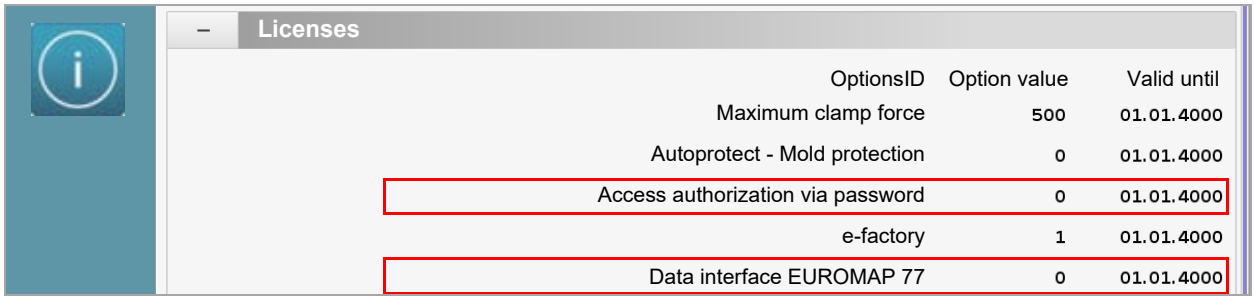
Machine network	System network
192.168.100.1	192.168.101.10
192.168.100.2	192.168.101.11
192.168.100.10	192.168.101.20
192.168.100.20	192.168.101.50 - 59
192.168.100.30	192.168.101.60
192.168.100.40	192.168.101.62
	192.168.101.64
192.168.110.x	192.168.101.70 - 85

   - **Subnet mask**
     Subnet mask
   - **Network gateway**
     IP-address of the gateway in the network
   - **Name server**
     The DNS server IP address can also be left blank.
   - **DNS search**
     Suffix for DNS requests.

3. Press **[Accept]** to confirm the change.
   Press **[Cancel]** to not save the change.
4. Restart the control unit so that the changes become effective.

4 EUROMAP 77 Authentication

Using the data interface EUROMAP 77 requires an OPC UA capable client software (the client) connecting to the OPC UA Server on the IMM (the server). To establish the connection, the client must authenticate to the server.
For authentication, the user name and password are required. As of software version V4.82, authentication is also possible with a certificate. An anonymous login is rejected by the server.
Together with the EUROMAP 77 data interface you will also receive the license 'Access authorization via password', which allows you to create users with a password locally on the machine. These users with a password can be used for authentication on the server.

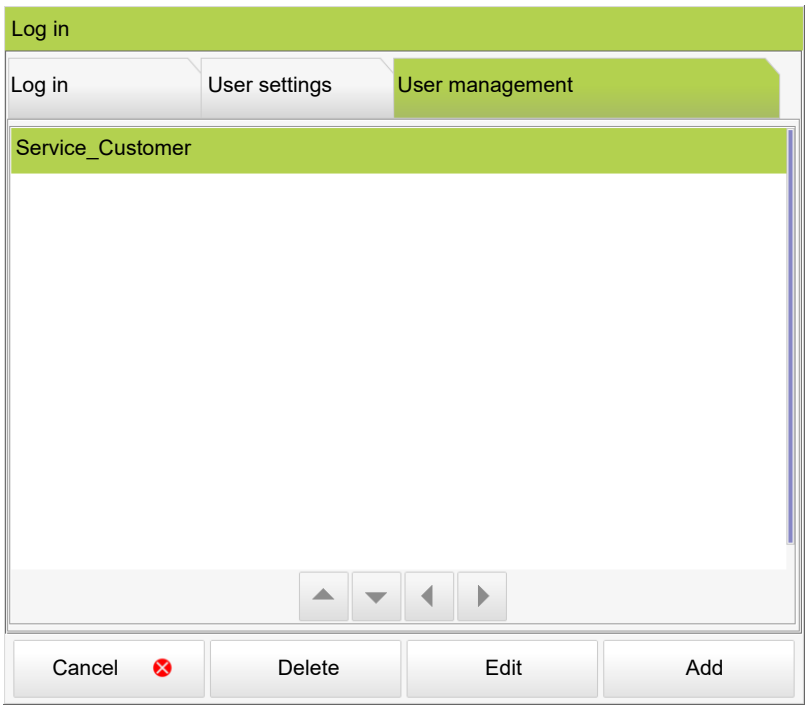


User creation for the EUROMAP 77 authentication:
Only required if authentication is not done with a certificate.

1. Log on to the machine using a user authorization card with the 'Administrator' role (e.g.: 'Service_Customer' user authorization card).
2. Tap on the user name in the header.

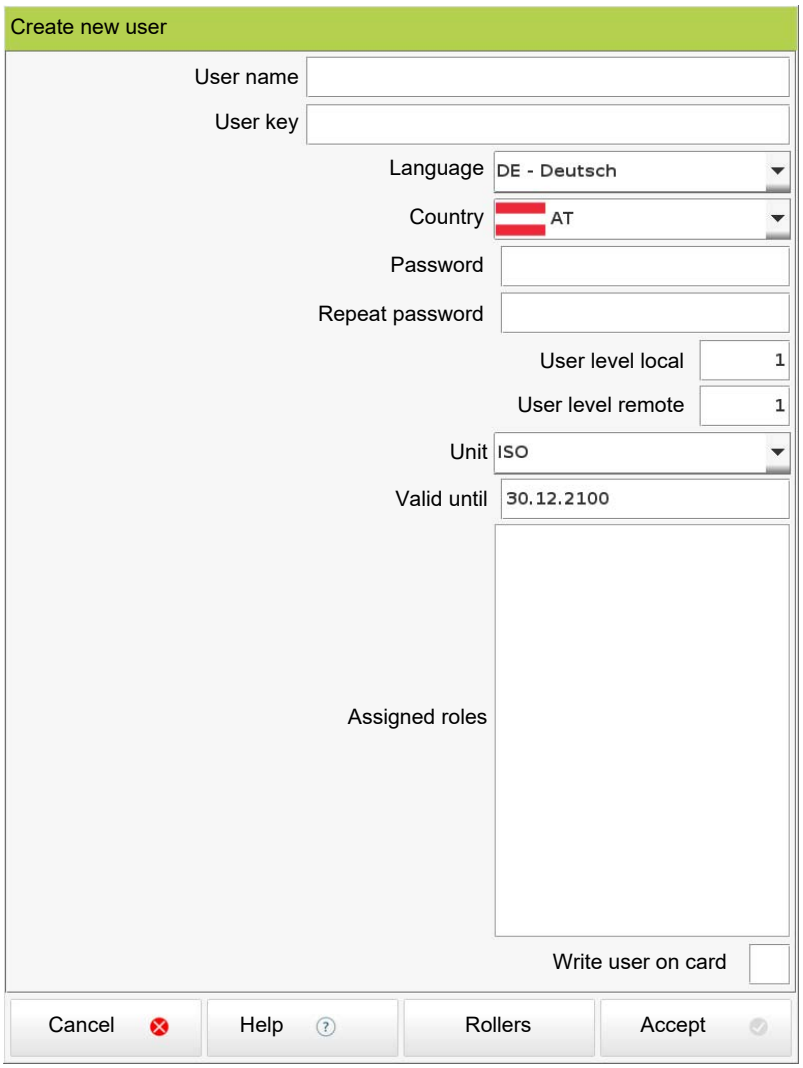
☐ 'Log-on' dialog box appears

3. Change to the 'user management' tab in the dialog window.



4. Press the **[add]** key.

☐ The dialog window 'create new user' appears.

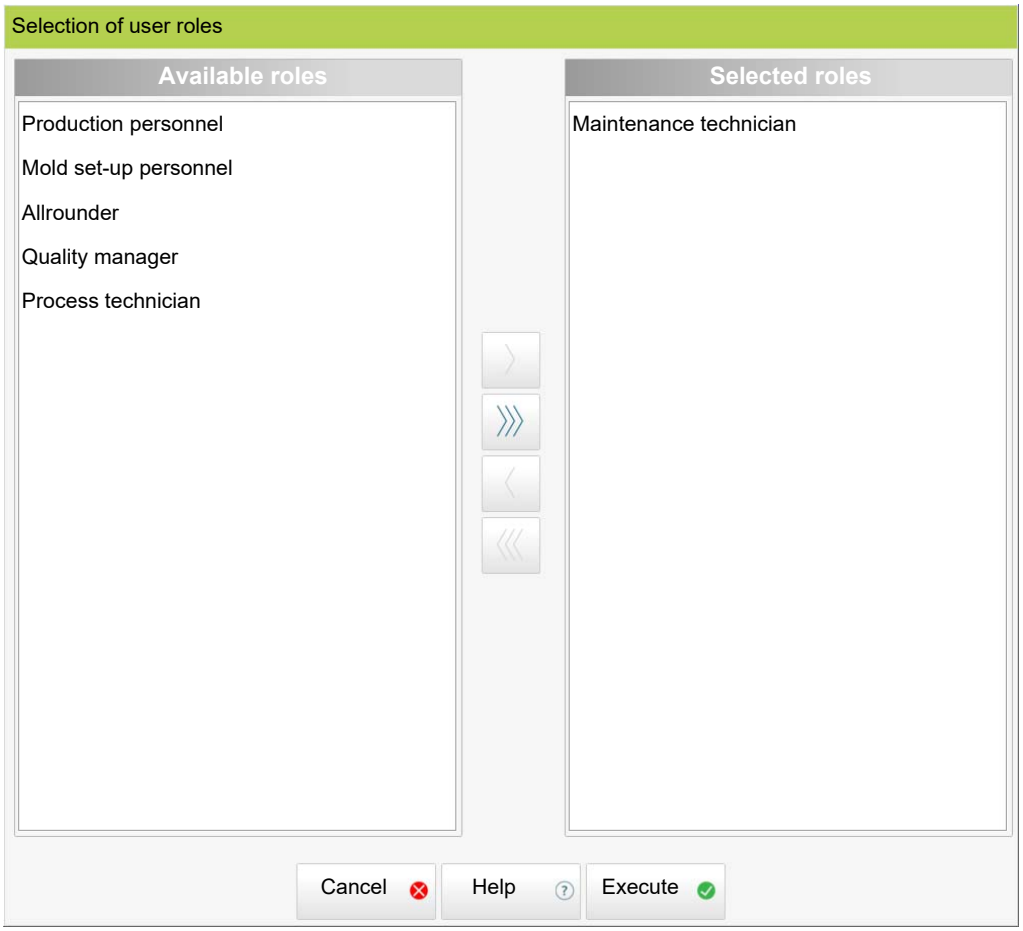


- **User name**
  Name of the new user
- **User key**
  The user key must contain only lowercase letters and numbers.
  For the authentication to the server, the user key is used as user name not the display user name. The text field User key will be prefilled with a generated unique iden-

tifier.
Change the user key to the desired user name for logging in at the server.

- **Language**
  Setting has no effect for the EUROMAP 77. English is always used in the OPC UA interface (at least from the machine).
- **Country**
  Setting has no effect for EUROMAP 77.
- **Password**
  The password must be at least 5 characters long. This password will be used later to authenticate to the server.
- **Repeat password**
- **User level local**
  Access level on the machine. User level is also used when writing via EUROMAP 77.
- **User level remote**
  Access level for the robot handheld operator panel, additional panel and access via the network to the machine. Setting has no effect for EUROMAP 77.
- **Unit**
  Setting has no effect for EUROMAP 77. ISO absolute is always used.
- **Valid until**
  Date up to which the user is valid.
- **Assigned roles**
  Setting has no effect for EUROMAP 77.

5. Activate ’Write user on card’ (optional).
6. Enter the relevant values from ’User name’ to ’Valid until’.
7. Click on the **[Roles]** button to select the requested roles.



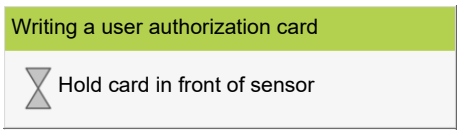
☐ By tapping (marking) an **Available roles** and pressing the arrow key**[>]**, this is

moved to the **Selected roles** and vice-versa.
The **[>>>]** and **[<<<]** keys move all roles.

- ☐ Press **[Execute]**, the roles are displayed under 'Assigned Roles'.

8. Press **[Accept]**.

- ☐ The following dialog appears. Hold the authorization card in front of the sensor until the dialog disappears.



4.1 Authentication with certificates

As of software version V4.82, authentication is possible with a certificate.

Term explanation

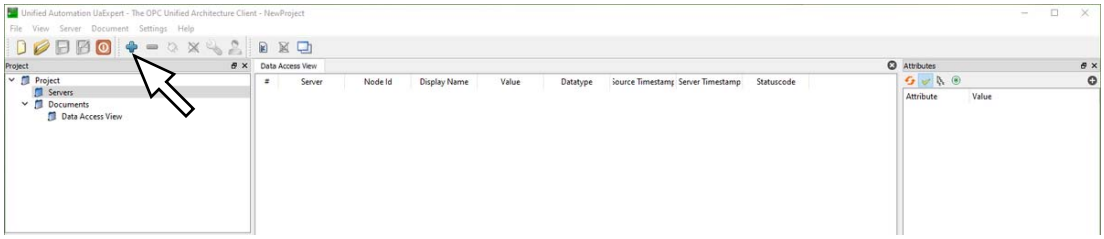
- **Application certificate**
  Uniquely identifies the application to the EUROMAP 77 server on the control unit.
- **User certificate**
  Uniquely identifies the user to the EUROMAP 77 server on the control unit.

Possible authentication methods on EUROMAP 77

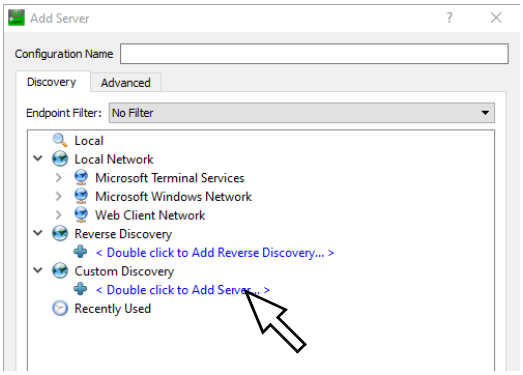
- **Authentication with user name and password**
  All users with the corresponding password, which are also available on the machine control unit, can be used.
- **Authentication with certificates**
  An application certificate and/or a user certificate is required for successful authentication with certificates. They are not issued by ENGEL. Self-signed certificates as well as official certificates from certification authorities (CA) are supported. Furthermore, certificates with the file extensions .crt, .cer, .der and .pem are possible.

4.1.1 Authentication with user certificates in UaExpert

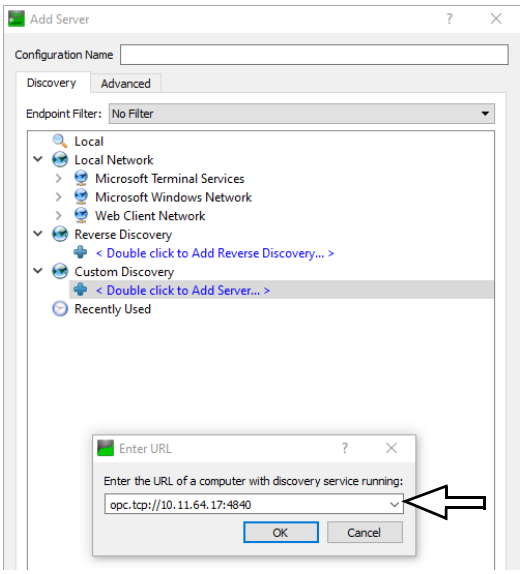
1. Create a new connection to an OPC UA server.



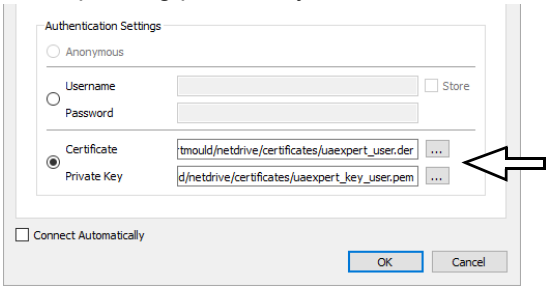
2. Double click on the following line.



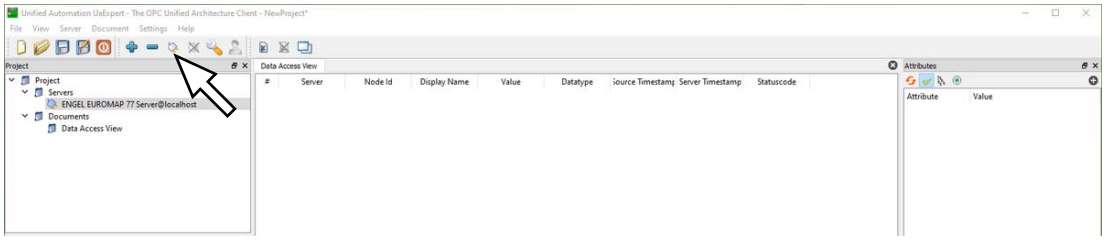
3. Select the required IP address and confirm with **[OK]**.



4. In the ’Certificate’ field, load the certificate file and, in the ’Private Key’ field, enter the corresponding private key.



5. Close the dialog window with **[OK]**.
6. Establish connection to the OPC UA server.

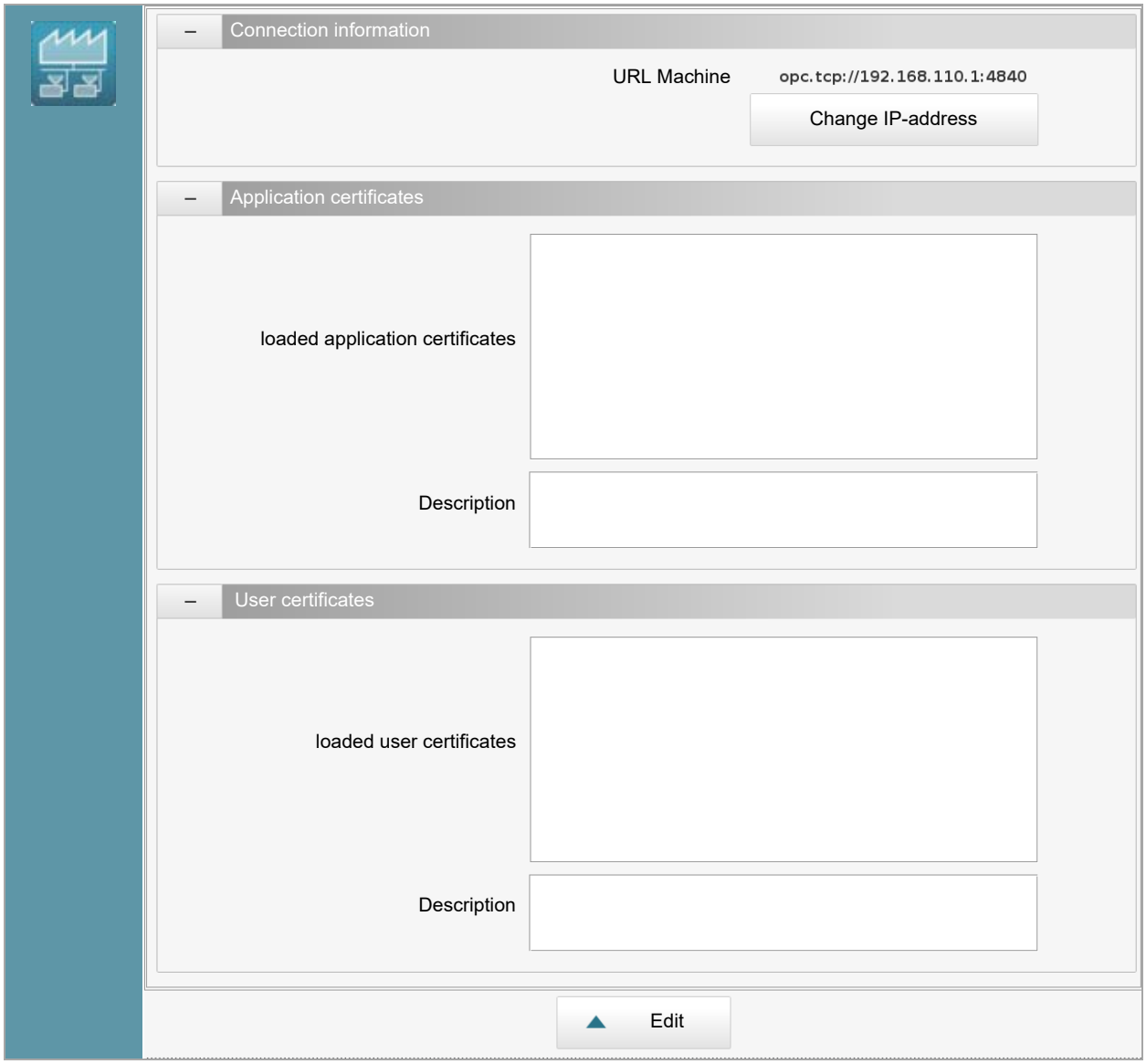


- The first connection fails because an unknown certificate is not trusted. To allow a connection to be established, the user certificate must be trusted on the machine control unit.
  [See Trust user certificates on the machine control unit on page 13.]

4.1.2 Trust user certificates on the machine control unit

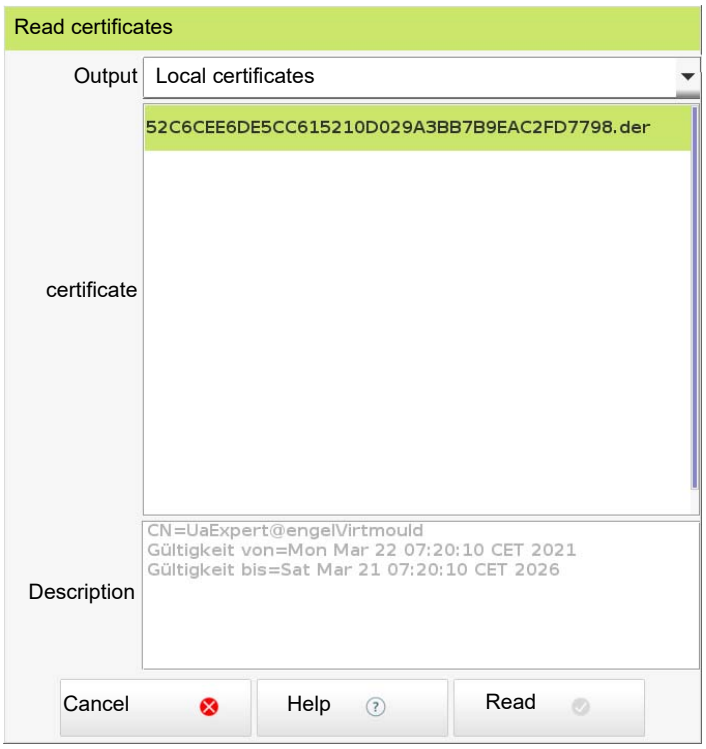
1. Log on to the machine control unit with user level 11.
2. Select screen page **Production management system**.

3. Select the **Configuration** tab.



4. Select menu key **[Edit]** -> **[Load user certificate]**.

   ☐ The following dialog box appears.



5. In the dialog box, select the certificate from a connected netdrive, USB stick or directly from the transferred certificates.
   The certificate transferred when a connection is established is given an automatically generated name to avoid any naming conflicts.
6. Press **[Read]**.

   ☐ After that, the connection to the OPC UA server is possible via UaExpert.

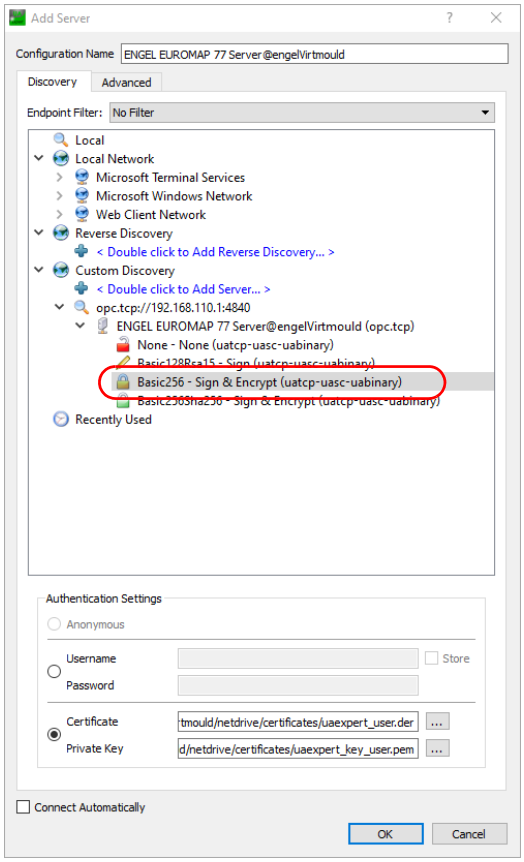
4.1.3 Trust application certificates on the machine control unit

If encrypted communication with the OPC server is required, the application certificate must also be trusted.

This requires creating a new server connection using one of the three supported encryption algorithms.

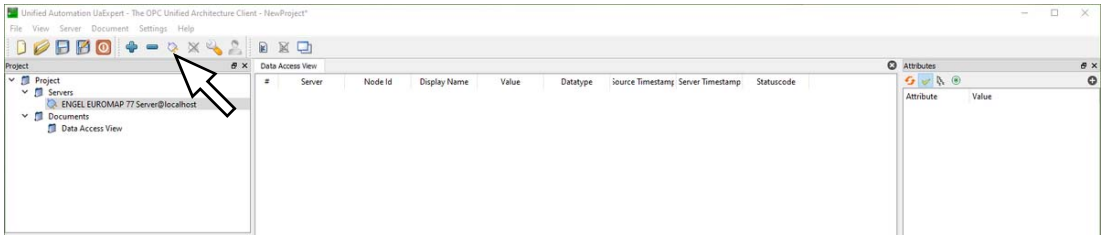
- Basic128
- Basic256
- Basic256Sha256

A certificate file does not need to be read in this case. The certificate is generated by UaExpert itself.



Procedure

1. Establish connection to the OPC UA server.



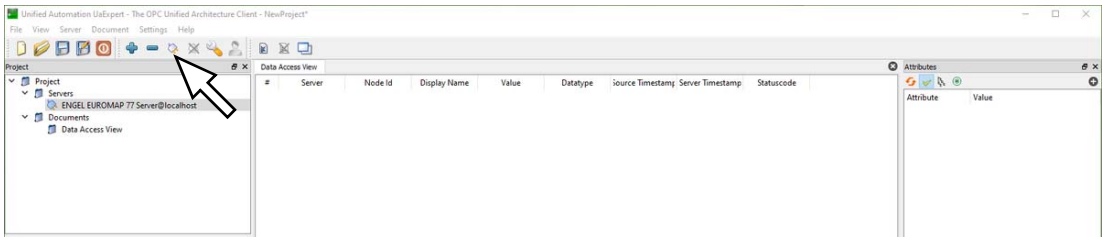
   ☐ This transfers application certificate to the machine control unit.
2. Log on to the machine control unit with user level 11.
3. Select screen page **Production management system**.

4. Select the **Configuration** tab.

5. Select menu key **[Edit]** -> **[Load application certificate]**.

   ☐ 'Read certificates' dialog box opens.
6. Select the transferred application certificate in the dialog box and press **[Read]**.

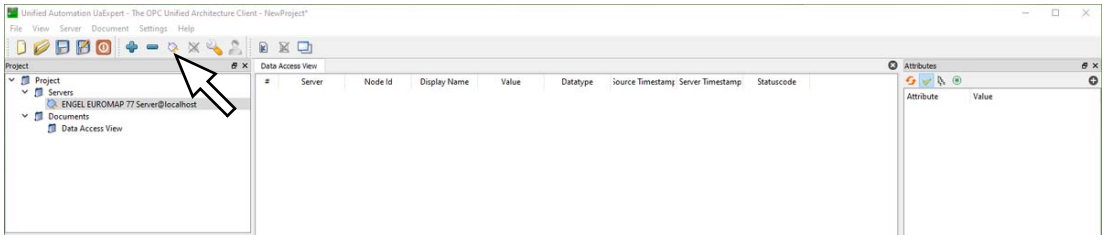
7. Establish connection to the OPC UA server.



- ☐ This transfers user certificate to the machine control unit.

8. Trust user certificate on the machine control unit.
[See Trust user certificates on the machine control unit on page 13.]

9. Establish connection to the OPC UA server.



4.1.4 Deleting the certificate

To delete a certificate again and thus no longer trust the certificate, the certificate to be deleted is selected on the **Manufacturing Execution System** screen page in the list of loaded certificates and deleted through the menu key **[Edit]** -> **[Delete application certificate]** or **[Delete user certificate]**.

5 ENGEL EUROMAP 77 Server

Establish a connection to the EUROMAP 77 server of the machine:

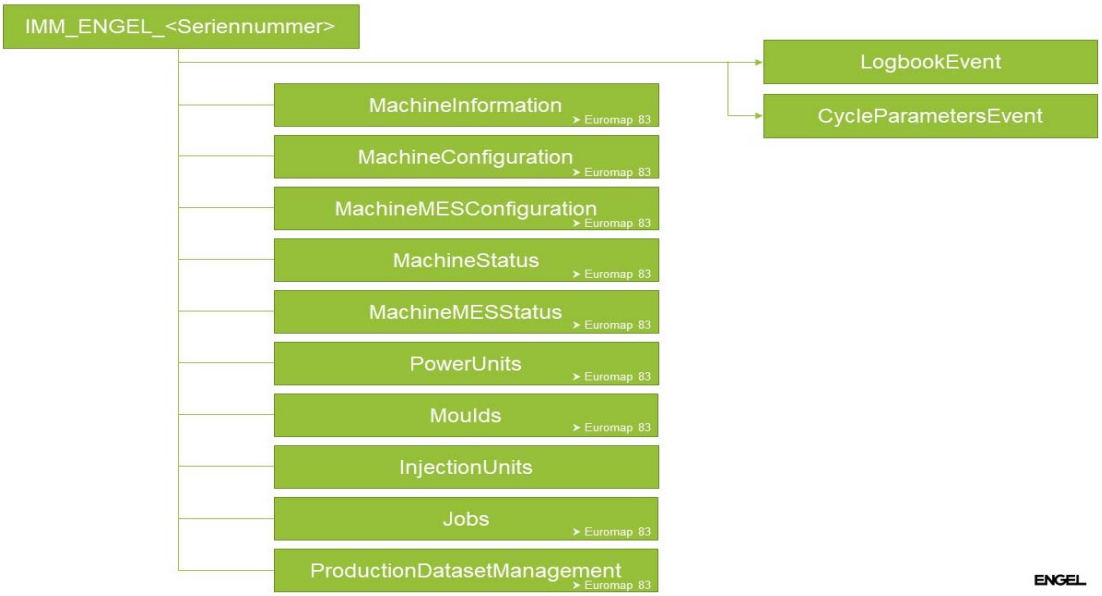
1. Enter the IP addresse of the machine with your client e. g. opc.tcp://<IP-adress>:4840 (Scheme = opc.tcp, Port = 4840). [See Network configuration on page 5.]
2. The login requires the user data of a machine user. This requires the creation of a user with user name and password. As of software version V4.82, also possible with certificate.
   [See EUROMAP 77 Authentication on page 8.]

Information

The user for logging on to the ENGEL OPC UA EUROMAP 77 server automatically has read and write permissions independent of the user level.

5.1 Representation of the injection moulding machine with its sub components

The moulding machine is grouped in the following sub components, in which the appropriate infor-mation and events are available. The sub components are accessible via BrowseName directly under the main object machine 'IMM_ENGEL_<Seriennummer>'.



Scheme from EUROMAP77-Specification
Figure 1 - IMM_MES_Interface Type Overview

5.1.1 LogbookEvent

EUROMAP 83 - Table 11

ParameterChangeLogType (Event)		
User User, who is responsible for the change that leads to the event	Mandatory	
EventOriginator Represents the originator of the event Default value: OPERATOR_2	Mandatory	Default value
JobCycleCounter Current value of JobCycleCounter	Optional	
ParameterId Id of the changed parameter	Mandatory	
OldValue Previous value	Mandatory	
OldValueUnit Unit of the old value	Optional	Only when unit can be provided of the control system
NewValue New value	Mandatory	
NewValueUnit Unit of the new value	Optional	Only when unit can be provided of the control system

UserLogType (Event)		
User User, who is responsible for the change that leads to the event	Mandatory	
EventOriginator Represents the originator of the event Default value: OPERATOR_2	Mandatory	Default value
JobCycleCounter Current value of JobCycleCounter	Optional	
UserChange Information of login (LOG_ON_0) or logoff (LOG_OFF_1)	Mandatory	

RemoteAccessLogType (Event)		
User User, who is responsible for the change that leads to the event	Mandatory	Not implemented
EventOriginator Represents the originator of the event Default value: OPERATOR_2	Mandatory	Not implemented
JobCycleCounter Current value of JobCycleCounter	Optional	Not implemented
RemoteUserName Name of the remote user	Mandatory	Not implemented
UserChange Information of login (LOG_ON_0) or logoff (LOG_OFF_1)	Mandatory	Not implemented
Orgin Information about the origin of the remote access	Optional	Not implemented

SequenceChangeLogType (Event)		
User User, who is responsible for the change that leads to the event	Mandatory	
EventOriginator Represents the originator of the event Default value: OPERATOR_2	Mandatory	Default value
JobCycleCounter Current value of JobCycleCounter	Optional	
SequenceChange Classification of the changes Current information on every change ’MODIFY_2’ **Supported values:** MODIFY_2 **Not available:** UPDATE_0 ADD_1 MOVE_3 DELETE_4	Mandatory	Default value

MachineModeChangeLogType (Event)		
User User, who is responsible for the change that leads to the event	Mandatory	
EventOriginator Represents the originator of the event Default value: OPERATOR_2	Mandatory	Default value
JobCycleCounter Value of shot counter	Optional	
OldMachineMode Old operation mode	Mandatory	
NewMachineMode New operation mode	Mandatory	

ProductionStatusChangeLogType (Event)		
User User, who is responsible for the change that leads to the event	Mandatory	
EventOriginator Represents the originator of the event Default value: OPERATOR_2	Mandatory	Default value
JobCycleCounter Value of shot counter	Optional	
OldProductionStatus Old production state	Mandatory	
NewProductionStatus New production state	Mandatory	

ProductionDatasetChangeLogType (Event)		
User User, who is responsible for the change that leads to the event	Mandatory	
EventOriginator Represents the originator of the event Default value: OPERATOR_2	Mandatory	Default value
JobCycleCounter Value of shot counter	Optional	
OldProductionDatasetName Name of previous activated data set	Mandatory	
NewProductionDatasetName Name of the new data set	Mandatory	

ProductionDatasetFrozenLogType (Event)		
User User, who is responsible for the change that leads to the event	Mandatory	Not implemented
EventOriginator Represents the originator of the event	Mandatory	Not implemented
JobCycleCounter Value of shot counter	Optional	Not implemented
OldValue Previous value	Mandatory	Not implemented
NewValue New value	Mandatory	Not implemented

StandstillReasonLogType (Event)		
User User, who is responsible for the change that leads to the event	Mandatory	
EventOriginator Represents the originator of the event Default value: OPERATOR_2	Mandatory	Default value
JobCycleCounter Value of shot counter	Optional	
StandstillReasonId Id of the current StandstillReasons	Mandatory	

MessageLogType (Event)		
User User, who is responsible for the change that leads to the event	Mandatory	
EventOriginator Represents the originator of the event Default value: MACHINE_1	Mandatory	Default value
JobCycleCounter Value of shot counter	Optional	
Id Id of alarm message	Mandatory	
IsStandstillMessage Indication if the message has led to a stand-still	Mandatory	Only based on alarm class (≤4)

MessageLogType (Event)		
Classification Classification of the alarm message	Mandatory	Not implemented
Active State Indicates whether error was set or reset.	Optional	Active/Inactive

UserFeedbackLogType (Event)		
User User, who is responsible for the change that leads to the event	Mandatory	Not implemented
EventOriginator Represents the originator of the event	Mandatory	Not implemented
JobCycleCounter Value of shot counter	Optional	Not implemented

5.1.2 CycleParametersEventType

EUROMAP 83 - Table 90

CycleParametersEventType		
JobName Name of the job	Mandatory	
JobStatus Current status of the job	Mandatory	
CurrentLotName Current production lot	Mandatory	
BoxId Id of the box	Optional	Not implemented
JobCycleCounter Number of finished cycles in the job	Mandatory	
BoxCycleCounter Number of finished cycles for the current box	Optional	Not implemented
MachineCycleCounter Number of finished cycles in the machine life time	Optional	
CycleTime Cycle time	Mandatory	
AverageCycleTime Average cycle time	Optional	Not implemented
JobPartsCounter Total number of produced parts	Mandatory	
JobGoodPartsCounter Number of good parts in the current job	Mandatory	

CycleParametersEventType		
JobBadPartsCounter Number of bad parts in the current job	Mandatory	
JobTestSamplesCounter Number of test sample parts in the current job	Mandatory	Not implemented
BoxPartsCounter Total number of produced parts in the current box	Optional	Not implemented
BoxGoodPartsCounter Total number of good parts in the current box	Optional	Not implemented
BoxBadPartsCounter Total number of bad parts in the current box	Optional	Not implemented
BoxTestSamplesCounter Total number of test sample parts in the current box	Optional	Not implemented
CycleQuality Information on the quality of the whole cycle Supported values: GOOD_CYCLE_0 in case of good part BAD_CYCLE_1 in case of bad part Not available: TEST_SAMPLE_CYCLE_2 FAILED_CYCLE_3	Mandatory	
CavityCycleQuality Information on the quality of the cycle for each cavity	Optional	Not implemented
PartId Ids of the parts which were produced in the cycle	Optional	Not implemented

5.1.3 EngelCycleParametersEventType

EUROMAP 77 - Extension of the CycleParametersEventType (Table 11)

- MouldCycleParameters_<Nr> (MouldCycleParametersType)
- InjectionUnitCycleParameters_<Nr> (InjectionUnitCycleParametersType)

MouldCycleParametersType		
Index Index of the mould Default value: 1	Mandatory	Default value
MouldTemperatureZones MouldTemperatureZone_<Nr>	Mandatory	

MouldCycleParametersType			
	Index Number of temperature zone within group	Mandatory	
	Name Name of temperature zone	Mandatory	
	Classification Type of the temperature zone	Optional	
	ActualTemperature Current temperature (real value) of the temperature zone	Mandatory	

InjectionUnitCycleParametersType		
Index Index of the InjectionUnit	Mandatory	
CushionVolume Material volume remained in front of the screw after injec-tion and holding pressure	Mandatory	
CushionStroke Screw position	Optional	Not implemented
PlastificationVolume Volume dosed by the machine for the next injection shot	Mandatory	
DecompressionVolumeBeforePlastification Decompression before plastification is the movement of the screw in the opposite direction to injection	Optional	
DecompressionVolumeAfterPlastification Decompression after plastification is the movement of the screw in the opposite direction to injection	Optional	
HydraulicPressureMaximum Maximum pressure in the hydraulic cylinder	Optional	Not implemented
SpecificPressureMaximum Pressure in the screw antechamber	Mandatory	
PlastificationRotationalSpeedMaximum Maximum plastification speed of the injection unit (RPM)	Optional	Not implemented
PlastificationRotationalSpeedAverage Average plastification speed of the injection unit (RPM)	Optional	Not implemented
PlastificationCircumferentialSpeedMaximum Maximum screw speed for plasticizing (mm/s)	Optional	Not implemented

InjectionUnitCycleParametersType		
PlastificationCircumferentialSpeedAverage Average screw speed for plasticizing (mm/s)	Optional	Not implemented
InjectionSpeedMaximum Maximum injection speed (mm/s)	Optional	
InjectionSpeedAverage Average injection speed (mm/s)	Optional	Not implemented
TransferVolume Switch-over point to the holding pressure via volume	Optional	
TransferStroke Switch-over point to the holding pressure via stroke	Optional	Not implemented
HoldHydraulicPressureMaximum Maximum holding pressure in the hydraulic cylinder	Optional	Not implemented
HoldHydraulicPressureAverage Average holding pressure in the hydraulic cylinder	Optional	Not implemented
HoldSpecificPressureMaximum Maximum holding pressure in front of the screw	Optional	Not implemented
HoldSpecificPressureAverage Average holding pressure in front of the screw	Optional	Not implemented
CavityPressureMaximum Maximum pressure during the injection process in the cavity or mould	Optional	Not implemented
PlastificationHydraulicPressureMaximum Maximum plastification pressure in cylinder	Optional	Not implemented
PlastificationHydraulicPressureAverage Average plastification pressure in cylinder Average plastification pressure in cylinder	Optional	Not implemented
PlastificationSpecificPressureMaximum Average plastification pressure in front of the screw tip	Optional	Not implemented
PlastificationSpecificPressureAverage Average plastification pressure in front of the screw tip	Optional	Not implemented
TransferHydraulicPressure Hydraulic pressure in the cylinder during switch-over to the holding pressure	Optional	Not implemented
TransferSpecificPressure Pressure in front of the screw tip during switch-over to the holding pressure	Optional	Not implemented

InjectionUnitCycleParametersType		
TransferCavityPressure Cavity pressure in the mould during switch-over to the holding pressure	Optional	Not implemented
BackPressure Back pressure is the melt-pressure against the screw movement during dosage	Optional	Not always available
InjectionTime Time required to fill the cavity or mould	Mandatory	
DosingTime Time to melt-up the plastic granulates and feed the melt for the next injection shot to the front of the screw	Mandatory	
FlowIndex Flow number	Optional	Not implemented
InjectionStartPosition Start position of the injection	Optional	Not implemented
VPChangeOverPosition Screw position at switching between injection (V) and hold-ing pressure (P)	Optional	Not implemented

Information

In the PD record, it is possible to provide further process parameters for the ENGEL EUROMAP 77 Server. For this purpose, it is necessary to define a new protocol on the machine.
The name of the protocoll must start with 'EUROMAP' (case sensitive!).
The exact description of the PD record see **Operator Manual - Machine**.

5.1.4 MachineInformation

EUROMAP 83 - Table 9

MachineInformation		
Manufacturer Manufacturer of the machine Default value: ENGEL Austria GmbH	Mandatory	Default value
DeviceManual Address of the user manual for the Device or web address of manufacturer. Default value: engelglobal.com	Mandatory	Default value
Model Machine Type Example: DUO 2460/650	Mandatory	
SoftwareRevision Software version Example: 4.30.00 (always 7 characters)	Mandatory	

MachineInformation		
SerialNumber Serial number of the machine Example: 226778	Mandatory	
DeviceRevision Overall revision level of the Device Default value: [CC300 A01]	Mandatory	Default value
HardwareRevision Revision level of the hardware of the Device Default value: [CC300 A01]	Mandatory	Default value
RevisionCounter Amount of modifications of static data within the device Default value: [-1]	Mandatory	Default value
DeviceClass Purpose of the Device Constant: „Injection Moulding Machine“	Optional	Default value
ControllerName Default value: [CC300]	Mandatory	Default value
SupportedLogbookEvents Information which LogbookEvents are supported by the machine Default value: [0;1;2;3;4;5;6;8;9] **Supported Events:** PARAMETER_CHANGE_0 USER_1 REMOTE_ACCESS_2 SEQUENCE_CHANGE_3 MACHINE_MODE_CHANGE_4 PRODUCTION_STATUS_CHANGE_5 PRODUCTION_DATASET_CHANGE_6 STANDSTILL_REASON_8 MESSAGE_9 **Not available:** PRODUCTION_DATASET_FROZEN_7 USER_FEEDBACK_10	Mandatory	Default value
EuromapSizeIndication	Optional	Not implemented

5.1.5 MachineConfiguration

EUROMAP 83 - Table 26

MachineConfiguration		
UserMachineName Individual machine name Default value: serial number of the machine	Mandatory, RW	
LocationName Description of the location of the machine	Mandatory, RW	
TimeZoneOffset Difference of the local time to UTC Default value: [0;false] Due to NTP (Network time protocol) on CC300 this method is not used	Mandatory, RW	Not implemented
SetMachineTime (Method) Setting time and TimeZoneOffset to UTC in minutes The time on control CC300 can be updated through NTP (Network Time Protocol) For description, please see the **Service Manual - Machine**	Mandatory	Not implemented
PageDirectory List of the screen pages of the machine (Id+Titel) Default value: [;]	Optional	Not implemented
GetPage (Method) Retrieving the image of a page of the control system	Optional	Not implemented
GetCurrentPage (Method) Screenshot of the control system with the currently shown contents	Optional	Not implemented

5.1.6 MachineMESConfiguration

EUROMAP 83 - Table 34

MachineMESConfiguration			
StandstillReasons Setting of the text for the standstill reasons on the machine. In99 the field Code on the machine, it is possible to set reason id from 1 to 99. But only 12 text explanations are shown on the central computer page. Software version <V4.82, see 'Central computer' screen page. Software version ≥V4.82, see 'Manufacturing Execution System' screen page on 'Data transmission' tab. Default value in the field Code: 0 for no standstill According to the specification, it will be an empty String		Mandatory, RW	
	Id Identifier of standstill reason Default value 0 (= no standstill)		
	Text Text of the standstill reason		
	LockedByMES Can be set or modified by the MES and may not be changed by the machine.		Not implemented
StandstillReasonsLockedByMES Indicates, if it is possible to change reasons on the machine. Constant: false		Mandatory, RW	Not implemented
MESUrl URL to display a webpage, which is generated by the MES in a web browser integrated in the machine. Default value: [MESUrl]		Optional, RW	Not implemented

5.1.7 MachineStatus

EUROMAP 83 - Table 36

MachineStatus			
IsPresent Informs the client if the machine is physically present and connected. The Server is always available when the machine is switched on and connected. Default value: [true]		Mandatory	Default value
Users Currently logged on users		Mandatory	
	Id Id of the user	Mandatory	
	Name Name of the user	Mandatory	
	IsPresent True, when the user is currently logged in. Indicates always the current user, therefore Default value: true	Mandatory	Default value
	CardUid UserId of the user card	Optional	
	UserLevel User level	Optional	
	UserRole User role of the current user	Optional	
	Language Currently used language	Optional	
MachineMode Current operation mode of the machine. **Supported values:** OTHER_0: if no other choice is correct. AUTOMATIC_1: full automatic mode (does not mean, that machine is producing). SEMI_AUTOMATIC_2: half automatic (does not mean, that machine is producing). MANUAL_3: manual mode or program interruption key. SETUP_4: setup-, teach- or calibration mode. **Not available:** SLEEP_5		Mandatory	
ActivateSleepMode Activation of sleep mode of MachineMode = SLEEP_5		Optional	Not implemented
DeactivateSleepMode Deactivation of sleep mode of MachineMode = SLEEP_5		Optional	Not implemented

5.1.8 MachineMESStatus

EUROMAP 83 - Table 41

MachineMESStatus		
StandstillReasonId Id of the current standstill reason. In99 the field Code on the machine, it is possible to set reason id from 1 to 99. But only 12 text explanations are shown on the central computer page. Software version <V4.82, see 'Central computer' screen page. Software version ≥V4.82, see 'Manufacturing Execution System' screen page on 'Data transmission' tab. Default value in the field Code: 0 for no standstill. According to the specification, it will be an empty String	Mandatory	
StandstillMessage Represents the fault which causes standstill. - Id - Message - Severity - Classification	Mandatory	
MESMessage Represents a text message sent from the MES to be shown on the machine. - Id - Message - Severity	Mandatory	Not implemented
SetMESMessage (Method) Method for setting the MESMessage. - Id - Message - Severity	Mandatory	Not implemented
ClearMESMessage (Method) Method for clearing the MESMessage.	Mandatory	Not implemented
ProductionControl Allows the MES to control the production of the machine.	Mandatory	

MachineMESStatus			
	ProductionStatus Represents the production status when the machine is in automatic or semi-automatic mode. One of the following information will be set. **Supported values:** OTHER_0 NO_PRODUCTION_1 READY_FOR_PRODUCTION_3 PRODUCTION_4 **Not available:** DRY_RUN_5 START_UP_2	Mandatory	
	ProductionReleasedByMES Release of production by MES. Default value: True.	Mandatory, RW	Not implemented
	AutomaticRunEnabled Indicates if semiautomatic or automatic run is allowed by MES. Default value: True Available from software version≥V4.80.	Mandatory, R	
	EnableAutomaticRun (Method) AutomaticRunEnabled = True Available from software version≥V4.80.	Mandatory	
	DisableAutomaticRun (Method) AutomaticRunEnabled = False Available from software version≥V4.80.	Mandatory	
	ProductionOnlyWithMES Indicates if production with the machine is only allowed when the MES is active. Default value: false	Optional, RW	Not implemented
	SetWatchDogTime (Method) Setting the duration time for which the production is released (seconds).	Optional	Not implemented
	ResetWatchDog (Method) Set of the timer again to the last value of „SetWatchDogTime“.	Optional	Not implemented
	RequestTestSample (Method) Request a test sample for separation.	Optional	Not implemented

5.1.9 MachineMESStatus Events

MachineMESStatus Events			
MessageConditionType (Event) Alarm message of the machine			
	Id Id of the alarm message.	Mandatory	
	Classification Classification of the message.	Mandatory	Not implemented
	IsStandstillMessage Indication if the message has led to a standstill.	Mandatory	
	Time Time when the alarm occured.	Mandatory	
	Message Text string displayed with the alarm indication that provides additional information to the operator.	Mandatory	
	Severity Urgency of the event.	Mandatory	

5.1.10 PowerUnits

EUROMAP 83 - Table 56

PowerUnit_<Nr>		
Index Number of the power unit.	Mandatory	Not implemented
IsPresent Informs if the power unit is physically present.	Mandatory	Not implemented
Id Represents the ID of the power unit.	Mandatory	Not implemented
PowerOn Power Unit is switched on or off.	Mandatory	Not implemented
ActualTemperature Current temperature of the power unit.	Optional	Not implemented
ActualPressure Current pressure of the hydraulic unit (only if machine has a hydraulic unit).	Optional	Not implemented

5.1.11 Moulds

EUROMAP 83 - Table 56

Mould_<Nr>			
Index Number of the mould		Mandatory	
Id Mold number Default value: ’Description’		Mandatory	
IsPresent Mould is currently on the machine Default value: true		Mandatory	Default value
Description Current (physical) status of the mould Returns value from the ’Mold number’ field on the machine. 20 characters for software version <V4.80 50 characters for software version ≥V4.82		Mandatory	
MouldStatus Current (physical) status of the mould Current (physical) status of the mould. Is always related to current mould on the machine, therefore Default value: MOULD_INSTALLED_3		Mandatory	Default value
MouldTemperatureZones MouldTemperatureZone_<Nr>		Mandatory	
	Index Number of temperature zone within group	Mandatory	
	Name Default name of the temperature zone	Mandatory	
	IsPresent Temperature zone is present and connected Default value: true	Mandatory	Default value
	Classification Type of the temperature zone ’Supported values:’ HEATING_1 COOLING_2 **Not available:** OTHER_0 TEMPERATURE_CONTROL_3 HOT_RUNNER_4 MEASURING_5	Optional	

Mould_<Nr>			
	ControlMode Control mode of the temperature zone **Supported values:** OFF_1 AUTOMATIC_2 OPEN_LOOP_5 ONLY_MEASUREMENT_6 **Not available:** OTHER_0 TUNING_3 STANDBY_4	Mandatory	
	NominalTemperature Nominal value	Mandatory	
	HighDeviationTemperature1 Maximum value that is in the normal tolerance	Optional	
	HighDeviationTemperature2 Maximum tolerable value	Optional	Not implemented
	LowDeviationTemperature1 Minimum tolerable value	Optional	
	LowDeviationTemperature2 Minimum tolerable value	Optional	Not implemented
	ActualTemperature Current temperature (real value) of the temperature zone	Mandatory	
	StandbyTemperature Lowering temperature	Optional	

5.1.12 InjectionUnits

EUROMAP 77 - Table 9

InjectionUnit_<Nr>		
Index Number of the injection unit	Mandatory	
BarrelId Id of the barrel	Mandatory	Equals Index
IsPresent Information whether injection unit is currently physically connected.	Mandatory	
InProduction Information, if the InjectionUnit is used in the current running	Mandatory	

InjectionUnit_<Nr>			
ScrewId Id of the screw Default value: „"		Optional	Default value
ScrewDiameter Screw diameter		Optional	
ScrewVolume Shot volume		Optional	
MaxScrewStroke Maximum stroke of the screw		Optional	
TemperatureZones BarrelTemperatureZone_<Nr>		Mandatory	
	Index Number of the BarrelTemperatureZone	Mandatory	
	Name Name of the BarrelTemperatureZone	Mandatory	
	IsPresent Default value: true	Mandatory	Default value
	Classification Type of the temperature zone **Supported values:** HEATING_1 COOLING_2 **Not available:** OTHER_0 TEMPERATURE_CONTROL_3 HOT_RUNNER_4 MEASURING_5	Mandatory	
	ControlMode Control mode of the temperature zone **Supported values:** OFF_1 AUTOMATIC_2 OPEN_LOOP_5 ONLY_MEASUREMENT_6 **Not available:** OTHER_0 TUNING_3 STANDBY_4	Optional	
	NominalTemperature Nominal value	Mandatory	
	HighDeviationTemperature1 Maximum value that is in the normal tolerance	Optional	
	LowDeviationTemperature1 Minimum tolerable value	Optional	

InjectionUnit_<Nr>			
	HighDeviationTemperature2 Maximum tolerable value	Optional	Not implemented
	LowDeviationTemperature2 Minimum tolerable value	Optional	Not implemented
	ActualTemperature Current temperature (real value) of the temperature zone	Mandatory	
	StandbyTemperature Lowering temperature	Optional	
	Position Position of temperature zone. Starting with ’1’ from the material supply. The highest position is at the nozzle. CAUTION! In contrast, the sequence in the machine starts at the nozzle with ’1’.	Mandatory	

5.1.13 Jobs

EUROMAP 83 - Table 66

Jobs			
ActiveJob Current order		Mandatory	The attributes can be described via **SetCyclicJobData**.
	JobName Name of the job	Mandatory	
	JobDescription Description of the job	Mandatory	
	CustomerName Name of the customer for the current job	Mandatory	
	ProductionDatasetName Name of the production dataset	Mandatory	
	ProductionDatasetDescription Description of the data set	Mandatory	
	Material Material name used for the job. Limited to one entry per injection unit.	Mandatory	
	ProductName Name of the product(s) (multiple cavity). Limited to one entry.	Mandatory	

Jobs			
	ProductDescription Description of the product	Mandatory	
	ContinueAtJobEnd Indicates if the machine continues the production even if the nominal output has been reached. Default value: [false]	Mandatory	Default value
	NominalParts Total number of parts that shall be produced by the job (Sum of all cavities)	Mandatory	
	NominalBoxParts Number of parts that shall be put into one box	Optional	
	ExpectedCycleTime Calculated cycle time for the job	Optional	
	MouldId Id from the mould	Mandatory (RW)	
	NumCavities Number of cavities	Mandatory	
	SetCyclicJobData (Method) Method for setting the job data for cyclic jobs from MES to machine **Input-Argumente:** [in] String JobName [in] String JobDescription [in] String CustomerName [in] String ProductionDatasetName [in] String ProductionDatasetDescription [in] String[] Material [in] String[] ProductName [in] String[] ProductDescription [in] Boolean ContinueAtJobEnd [in] UInt64 NominalParts [in] UInt64 NominalBoxParts (NULL) [in] Duration ExpectedCycleTime (NULL) [in] String MouldId [in] UInt32 NumCavities)	Mandatory	Not implemented Not implemented Not implemented Not implemented
	RequestCyclicJobWriteEventType Event is used to initiate a call of the SetCyclicJob-Data method by the client		Not implemented
JobInPreparation Next planned order incl. all properties according to 'ActiveJob'.		Optional	Not implemented
SendJobList (Method) Send a list of jobs available on the client to the server		Optional	Not implemented

Jobs			
SendCyclicJobList (Method) Send a list of jobs for cyclic production (instead SendJobList)		Optional	Not implemented
ActiveJobValues Values of the active job		Mandatory	
	JobStatus Current status of the job Default value: [0] **Supported values:** JOB_FINISHED_8 JOB_IN_PRODUCTION_6 JOB_INTERRUPTED_7 OTHER_0 **Not available:** TRANSFERRED_ASSIGNED_1 SET_UP_ACTIVE_2 SET_UP_INTERRUPTED_3 SET_UP_FINISHED_4 START_UP_ACTIVE_5 TEAR_DOWN_ACTIVE_9 TEAR_DOWN_INTERRUPTED_10 TEAR_DOWN_FINISHED_11	Mandatory	
	StartJob (Method) Change the JobStatus to JOB_IN_PRODUCTION_6	Mandatory	Not implemented
	InterruptJob (Method) Change the JobStatus to JOB_INTERRUPTED_7	Mandatory	Not implemented
	FinishJob (Method) Change the JobStatus to JOB_FINISHED_8	Mandatory	Not implemented
	CurrentLotName Current production lot Default value: „“	Mandatory, RW	
	BoxId Id of the box in which the current production is put in	Optional	
	JobCycleCounter Number of finished cycles in the job	Mandatory	
	BoxCycleCounter Number of finished cycles for the current box	Optional	Not implemented
	MachineCycleCounter Number of finished cycles in the machine life time	Optional	Implemented

Jobs			
	LastCycleTime Cycle time of the recently finished cycle (millisec.)	Mandatory	
	AverageCycleTime Average cycle time	Optional	Not implemented
	JobPartsCounter Total number of produced parts	Mandatory	
	JobGoodPartsCounter Number of good parts in the current job	Mandatory	
	JobBadPartsCounter Number of bad parts in the current job	Mandatory	
	JobTestSamplesCounter Test sample parts are – according to the machine settings - rated as good or bad part. Therefore test sample parts counter is always 0. Default value: [0].	Mandatory, R	Default value and/or. Not implemented
	BoxPartsCounter Total number of produced parts in the current box	Optional	Not implemented
	BoxGoodPartsCounter Total number of good parts in the current box	Optional	Not implemented
	BoxBadPartsCounter Total number of bad parts in the current box	Optional	Not implemented
	BoxTestSamplesCounter Total number of test sample parts in the current box	Optional	Not implemented
	LastPartId Ids of the parts produced in the recently finished cycle	Optional	Not implemented
	StopAtCycleEnd (Method) MES directs the machine to stop at the end of the current cycle Available from software version≥V4.80.	Mandatory	
	ResetJobCounters (Method) Setting the cycle and parts counters for the job to 0	Mandatory	
	ResetBoxCounters (Method) Setting the cycle and parts counters for the current box to 0	Optional	Not implemented
	ResetAverageCycleTime (Method) Initiates a new calculation of the average cycle time for the job	Optional	Not implemented

5.1.14 Jobs Events

Jobs Events		
RequestJobListEventType (Event) Initiates a call of SendJobList method by the client.		Not implemented
RequestCyclicJobListEventType (Event) Initiates a call of SendCyclicJobList method by the client.		Not implemented

5.1.15 ProductionDatasetManagementType

EUROMAP 83 - Table 96

ProductionDatasetManagementType			
ActiveProductionDatasetStatus		Mandatory	
	Information Status of the production dataset which is active in the control system of the machine	Mandatory	
	Modified Informs if the production dataset has been changed	Optional	Not implemented
	Frozen If TRUE, no changes on the machine in the produc-tion dataset (change of process parameters) are al-lowed.	Optional	Not implemented
	Load (Method) Loads a production dataset from the file system of the machine to the con-trol of the machine	Optional	
	Save (Method) stores a production dataset from the control of the machine to the file sys-tem of the machine with the given name	Optional	
ProductionDatasetInPreparationStatus Status of the production dataset which is in preparation		Optional	Not implemented
ProductionDatasetLists Is used to exchange information on the avai-lable pro-duction datasets on client and ser-ver		Optional	
	GetProductionDatasetList For reading a list from the server which production datasets are available on the machine's file system		Not implemented

ProductionDatasetManagementType			
	SendProductionDatasetList For sending a list of production datasets available on the client to the server		
ProductionDatasetTransfer For the transfer of production datasets		Mandatory	
	ClientProcessingTimeout Maximum of accepted duration in milliseconds for the server between method call and finished file transfer Default value: [120]		Default value
	GenerateFileForRead (Method) Dataset transfer from the machine to the client **Input argument:** ProductionDatasetReadOptionsType: Storage / Name **Supported values for Storage:** PRODUCTION_1: Create part data set from current settings FILE_SYSTEM_4: Read part data set with name from input argument **Not available:** PREPARATION_2		
	GenerateFileForWrite (Method) Dataset transfer from the client to the machine Input argument: ProductionDatasetWriteOptionsType: Storage / Name / Components **Supported values for Storage:** PRODUCTION_1: Movement of a part data set to the machine control with immediate activation FILE_SYSTEM_4: Movement of a part data set to the local file system of the machine. Name is required. **Not available:** PREPARATION_2		
	CloseAndCommit (Method) Termination of GenerateFileForWrite		

Further supported OPC UA methods of the temporary FileType-Object according to OPC UA Part 5:

- Read (Method)
- Write (Method)
- Close (Method)

GetProductionDatasetInformation (Method) Allows reading the description of a production da-taset during the file transfer from the server to the cli-ent with ProductionDataset-Transfer.	Optional	
SendProductionDatasetInformation (Method) Allows sending of the description of a production da-taset during the file transfer from the client to the server with ProductionDataset-Transfer.	Optional	

5.1.16 ProductionDatasetLists Events

ProductionDatasetLists Events		
RequestProductionDatasetListEventType Request to MES to send possible data sets (info about the data sets) to the machine via SendProductionDatasetList.		

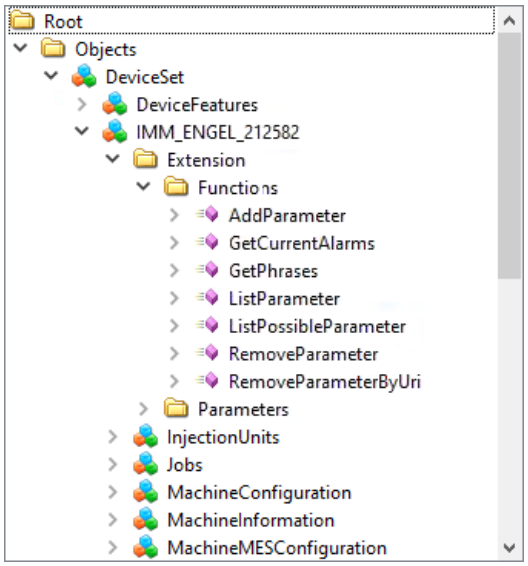
5.1.17 ProductionDatasetTransfer Events

ProductionDatasetTransfer Events		
RequestProductionDatasetReadEvent-Type Event to trigger a file transfer from the machine/server to the client. The event can be initiated e.g. by the operator.		Implemented
RequestProductionDatasetWriteEvent-Type Event to trigger a file transfer from the client to the machine/server. The event can be initiated e.g. by the operator.		Implemented

6 ENGEL Extensions

The following functions are an extension of ENGEL and are not defined in EUROMAP 77/83.

Integration of the Extention in the IMM_MES_InterfaceType



Behavior of the node IDs

For ENGEL machines, it must be taken into account that the node ID can change after the control unit is switched off/on. When a client makes a new connection, it should be expected that the node IDs may have changed in the meantime.
Based on this, a good approach would be to traverse the OPC UA information model at the beginning of a new client session and thus implicitly after switching off/on the machine, and to redetermine the node IDs/browse paths. It is not advisable to use the node ID fixed coded.

AddParameter

Create a new data end point in the extended information model. Data end point is added to the 'Parameters' folder and remains even over a restart. However, this may change the NodeId. Further information, see **ListParameter**. [See ENGEL Parameters URI on page 46.]

```
AddParameter (
[In] String ParameterURI
[Out] NodeId NodeId);
```

GetCurrentAlarms

Returns all alarms pending at the system.

```
GetCurrentAlarms (
[Out] AlarmDataType[] Alarms);
```

GetPhrases

Phrases for the PhraseKeys are returned in the specified language. For unknown languages, en (English) is used.
Language codes according to ISO 639-1 e.g. en, de.

```
GetPhrases (
[In] String[] PhrasesKeys,
[In] String Language,
[Out] String[] Phrases);
```

A PhraseKey can also be a parameter URI. This makes it possible to get the short and long text of a variable (result: 'short text|long text').

ListParameter

Listing of the currently included data endpoints in the extended information model
Function can be helpful after a restart, as this can change the NodeId´s of the data end points added via AddParameter.

```
ListParameter (
[Out] ParameterDataType[] ListOfAddedUris);
```

ListPossibleParameter

Listing of the parameters, which are available for extension of the information model
Existing variables can be searched for and then added via their URI using **AddParameter**.

```
ListPossibleParameter (
[In] String MachineryPart;
[In] String[] Components;
[Out] PossibleParameterDataType[] PossiblePa-
rameters);
```

Input arguments:
MachineryPart e.g. cc300://imm/
Components: LongText of the desired unit from 'PossibleParameterDataType'

The 'PossibleParameterDataType' is as 'SubType' from 'BaseDataType' defined as follows:

```
PossibleParameterDataType{
String VisUri;
String LangText;
};
```

Step-by-step approach possible

- Call without parameters:
  The MachineryParts are located in VisUri. Possible arguments of MachineryParts are returned.
- Call with MachineryPart / without components:
  Possible components are returned.
- Call with MachineryPart + component(s):
  The LongText arguments contain the components. Example for shot volume: VisUri = cc300://imm/cm#//c.InjectionUnit1/p.sv_rPlastStopVol/v; LongText = sv_rPlastStopVol

RemoveParameter

Deleting a data end point from the extended information model via its NodeId. Data end point is deleted from the 'Parameters' folder.

```
RemoveParameter (
[In] NodeId NodeId);
```

RemoveParameterByUri

Deleting a data end point from the extended information model via its URI. Data end point is deleted from the 'Parameters' folder.

```
RemoveParameterByUri (
[In] String Uri);
```

6.1 ENGEL Parameters URI

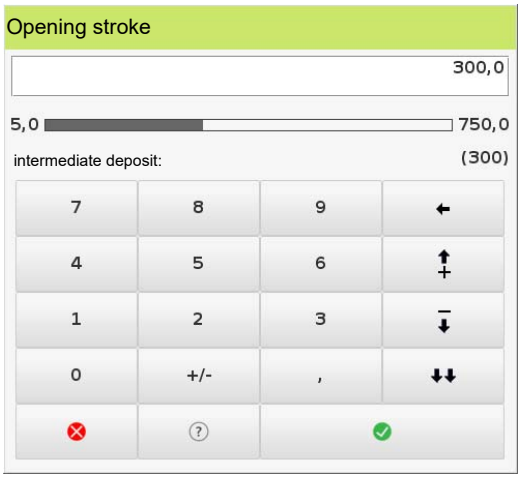
The ENGEL Extensions are also used to dynamically add parameters that are available in the machine software but are not covered by the standard EUROMAP 77 interface. To fully use this function, you need to know the parameter URI. The parameter URI (Unified Recource Identifier) is a unique designation of parameters that is used internally to identify parameters. Depending on the software version, various options for determining the parameter URI are available as described in the following chapters.

6.1.1 Determining parameter URI via function menu

Software version ≥V4.72

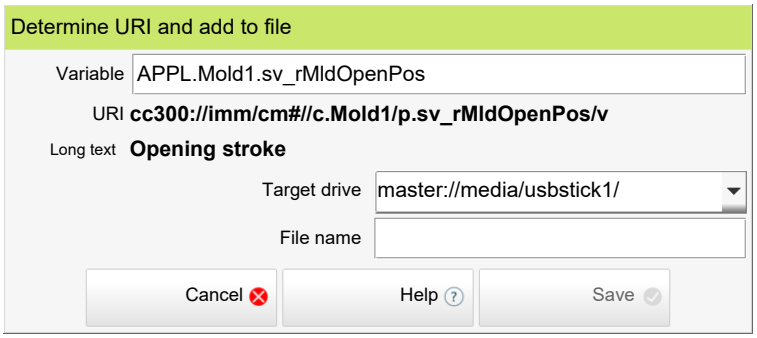
To determine the URI for a specific parameter proceed as follows:

1. Log on as a user with Level 11 rights.
2. Tap the desired parameter on the screen so that the input keyboard is displayed.



3. The dialog box closes again
4. Press **[Determine URI]** in the function menu.

   ☐ The following dialog box is displayed with the data of the previously selected parameter:



5. Change the target drive, if required, and input the file name. The control unit automatically adds the extension '.csv' to the file name while saving it.
6. Save
   If the file does not yet exist in the target drive, it is created by the control unit. Otherwise the control unit adds the current URI to the file.

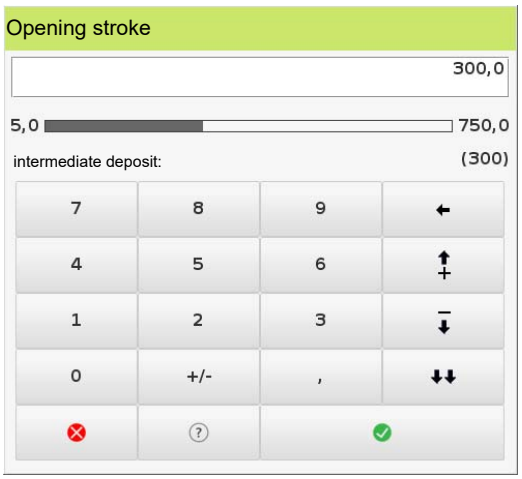
If the control unit does not find the 'variable', the following message is displayed:

6.1.2 Determining parameter URI with component model browser

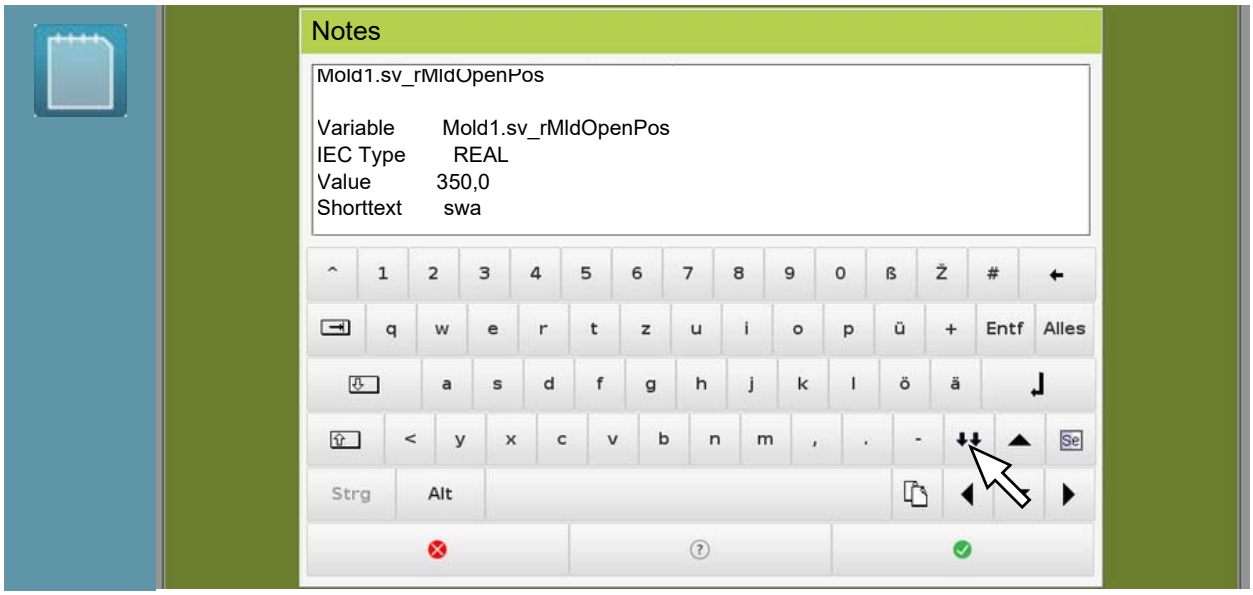
Software version <V4.72

To determine the URI for a specific parameter proceed as follows:

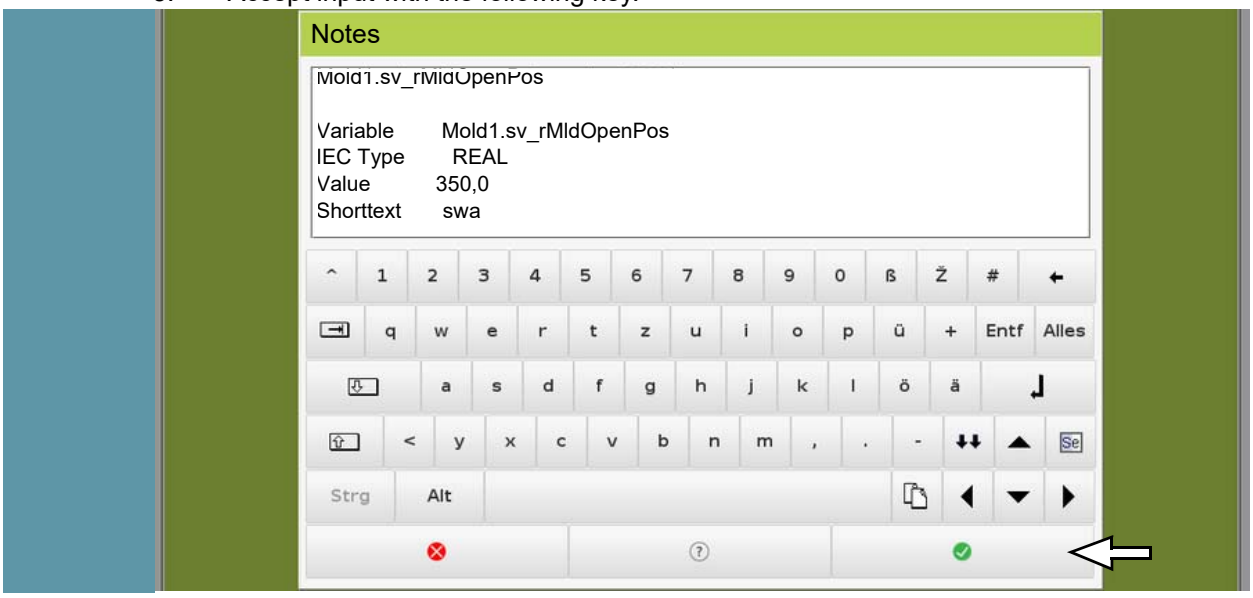
1. Log on as a user with Level 13 rights.
2. Tap the desired parameter on the screen so that the input keyboard is displayed.



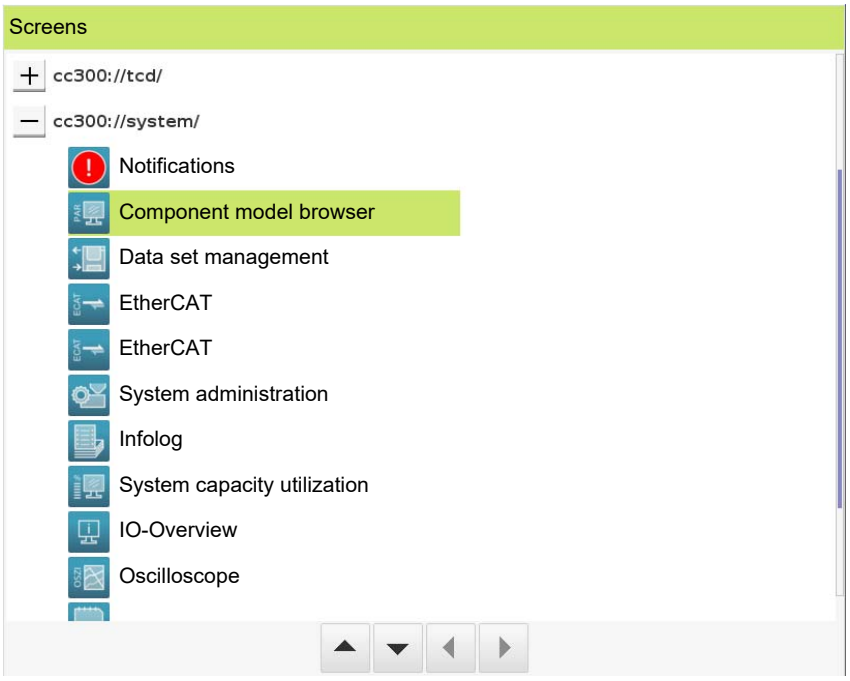
3. Close the dialog box again.
   Information for the parameter is thereby copied to the clipboard.
4. Open the 'Notes' screen page and insert the contents from the clipboard using the following key.
   The first line of the text contains the NativeKey (e.g. B. Mold1.sv_rMldOpenPos) which will be needed later.



5. Accept input with the following key.



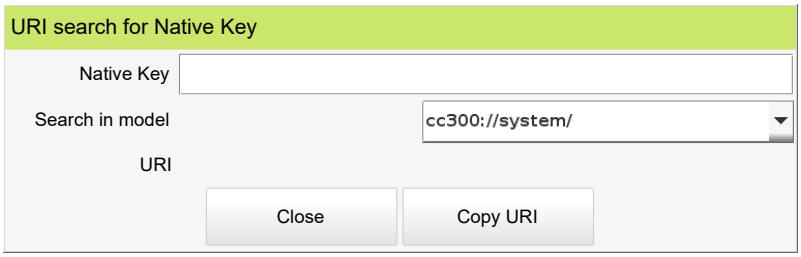
6. In the function menu, press **[Screen page selection]**.
7. In the dialog box, select the 'Component model browser' and **[Show]**.



☐ The 'Component model browser' appears.

8. In the menu, select **[Edit]** and **[Find URI]**.

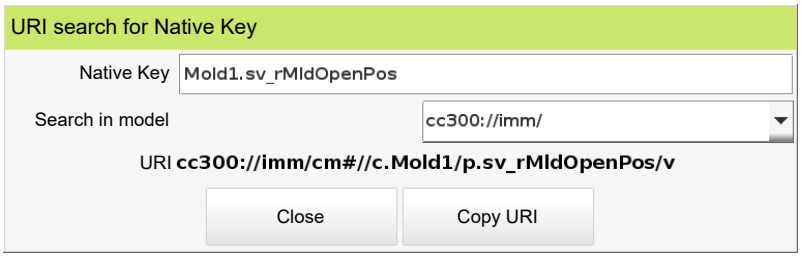
☐ The following dialog box appears:



9. Depending on the NativeKey for determining the URI, make the following setting in the dialog box:

- Native Key: Mold1.sv_rMldOpenPos
- Search in model: cc300://imm/

10. The determined URI can now be copied to the clipboard with **[Copy URI]**.



6.1.3 Determine parameter URI based on its native key

The key can be converted in a generic way. This is possible for almost all parameters except the heaters and e-flomo.

Example:

`Ejector1.sv_rEjeFPos`
is
`cc300://imm/cm#//c.Ejector1/p.sv_rEjeFPos/v`

Example:

`Ejector1.sv_BackwardAvailableInSequence[1]`
is
`cc300://imm/cm#//c.Ejector1/p.sv_BackwardA-vailableInSequence/v/p.[1]/v`

6.1.3.1 Definition of the parameters of the heating zones

The definition of the heating parameters depends on the individual machine configuration. This configuration also depends on the delivery state of the machine. In addition, the heating configuration can be adapted by the customer.
Therefore, it is not possible for ENGEL to offer a generally valid list of heating parameters. It must be generated individually for each machine, based on its heating configuration.
This part of the documentation describes how the parameter URI of the heating parameters is composed.

The parameters for the CC300 machine control unit are structured as follows:

`cc300://imm/cm#//c.TemperingComponent/group-number/zone-number/v/parameter/v`

Group number	p.TemperingGroup	n = group number of the heating group
Zone number	p.TemperingZone	n = zone number within the group starting with 1
Parameters	p.ActValue	Actual value
	p.SetValue	Set value
	pActionSignal	Heating power real value [%]

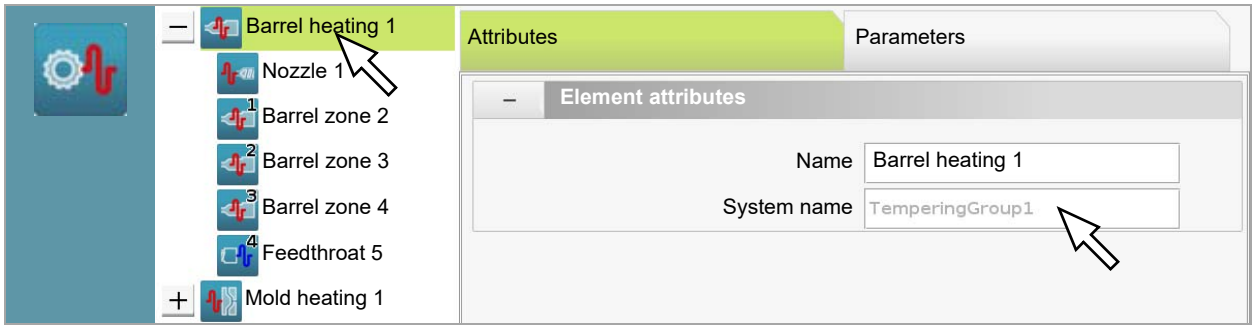
Example:

The parameter for the actual value of the nozzle heating (=first zone of the barrel heating group) is therefore:

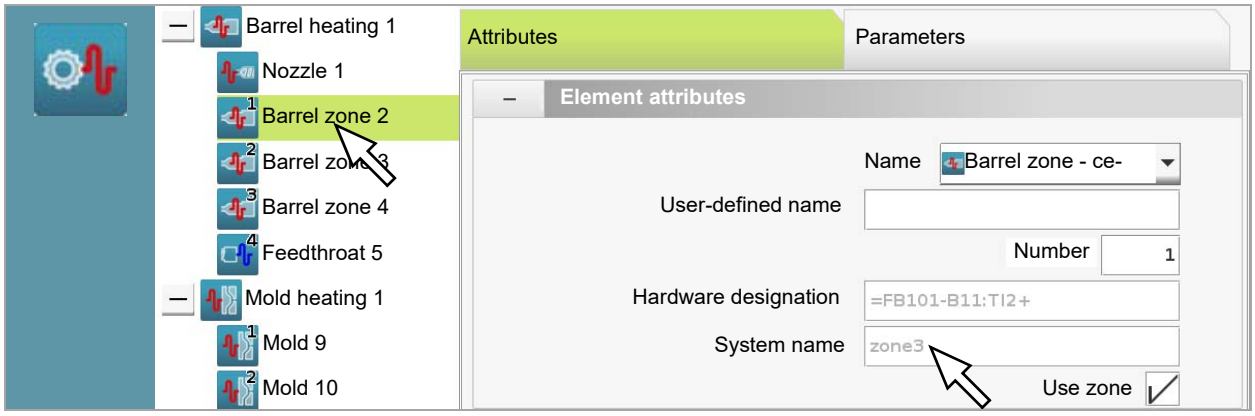
```
cc300://imm/cm#//c.TemperingComponent/p.Tem-
peringGroup1/p.TemperingZone1/v/p.ActValue/v
```

This is how you can check the zone number on the machine control unit
You can read the group and zone numbers on the 'Heating configuration' screen page on the CC300 machine control unit.

Group number
Select the heating group and read the group number on the Attributes tab.
In this example: TemperingGroup1.



Zone number
The zone number results from the position of the zone within the heating group.
In the following example, 'Barrel zone 2' has the zone number 3 because it is in the second position within the 'Barrel heating 1' group.



The set value of 'Barrel zone 2' of the 'Barrel heating_1' heating group (system name TemperingGroup1) is the following parameter:

```
cc300://imm/cm#//c.TemperingComponent/p.Tem-
peringGroup1/p.TemperingZone3/v/p.SetValue/v
```

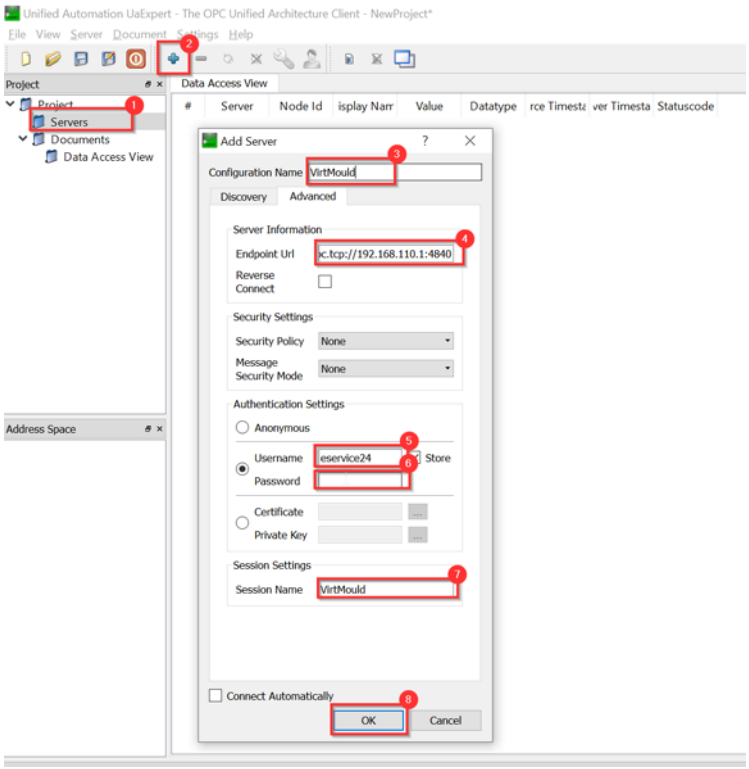
6.2 Application examples with UaExpert

This section of the document shows how you can use the EUROMAP 77 data interface with the UaExpert client software from Unified Automations GmbH. UaExpert is not an ENGEL product and therefore not included in the scope of delivery of the EUROMAP 77 data interface.

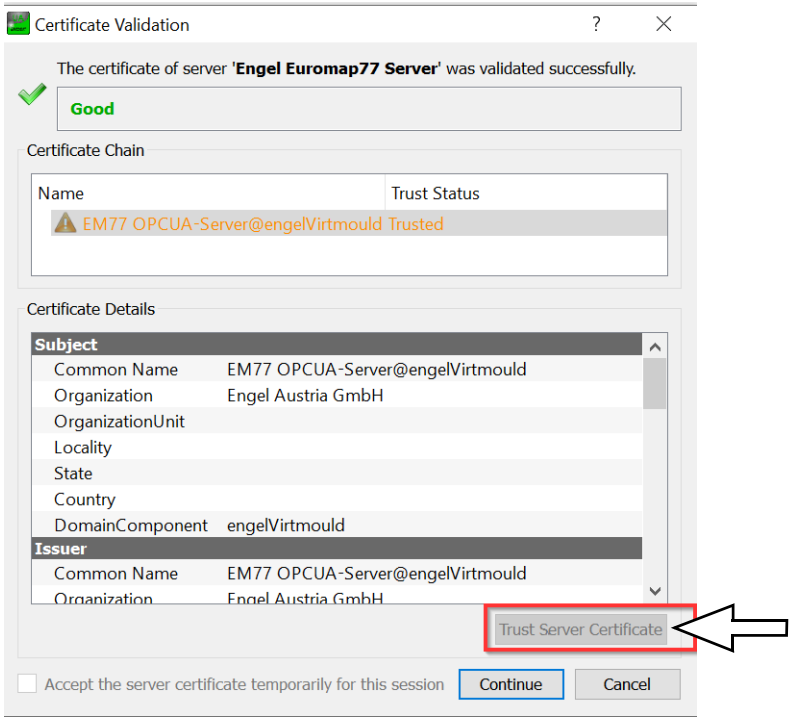
6.2.1 Connection build-up with UaExpert

Add an OPC UA server to the UA Client configuration with the following steps:

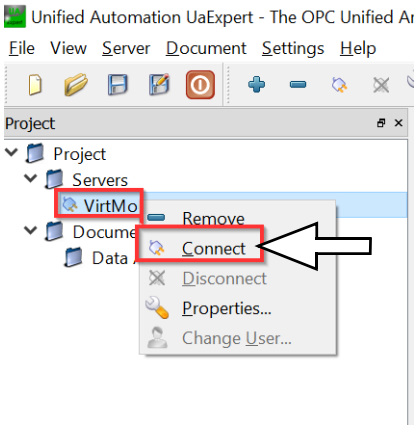
1. Select the 'Servers' folder.
2. '+' Select the icon for Add Server.
3. Enter a free name for configuration.
4. Define end point URL with IP address of machine opc.tcp://<IP-address>:4840
5. Enter the user who is available on the control unit.
   [See EUROMAP 77 Authentication on page 8.]
6. Enter password
7. Enter a free name for the session.
8. Confirm entries with OK.



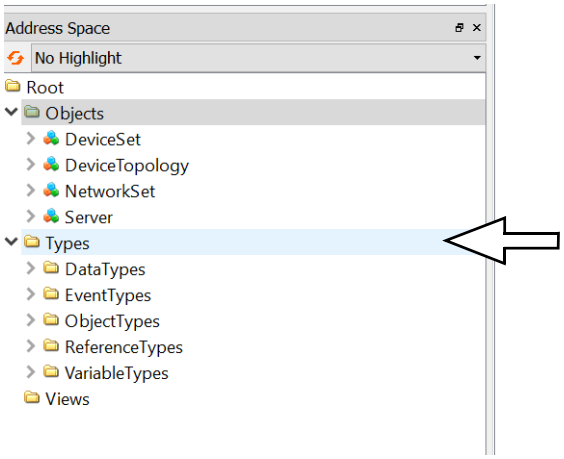
A certificate is initially exchanged between CC300 and UA Client to establish the connection. However, the one from the CC300 is self-signed, i.e. without an official certification authority. This must be accepted once as trustworthy by UaExpert.



The connection to the server is then established by means of "Connect":

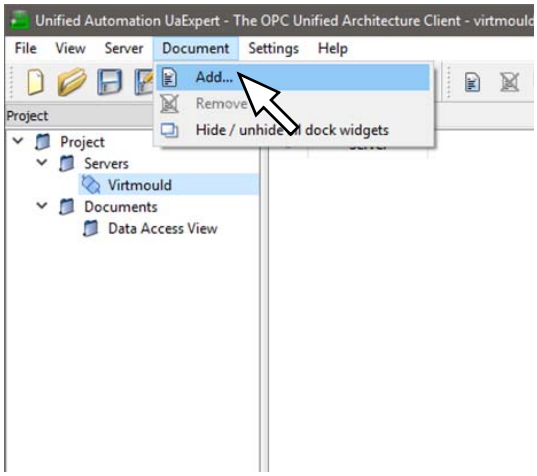


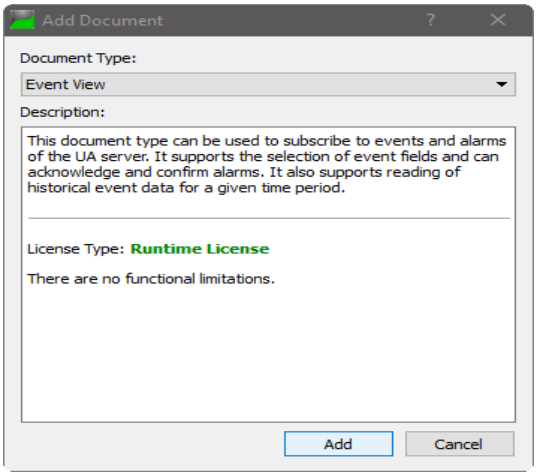
After a successful ’Connect’, the address space of the specific machine appears:



6.2.2 Subscribing to events

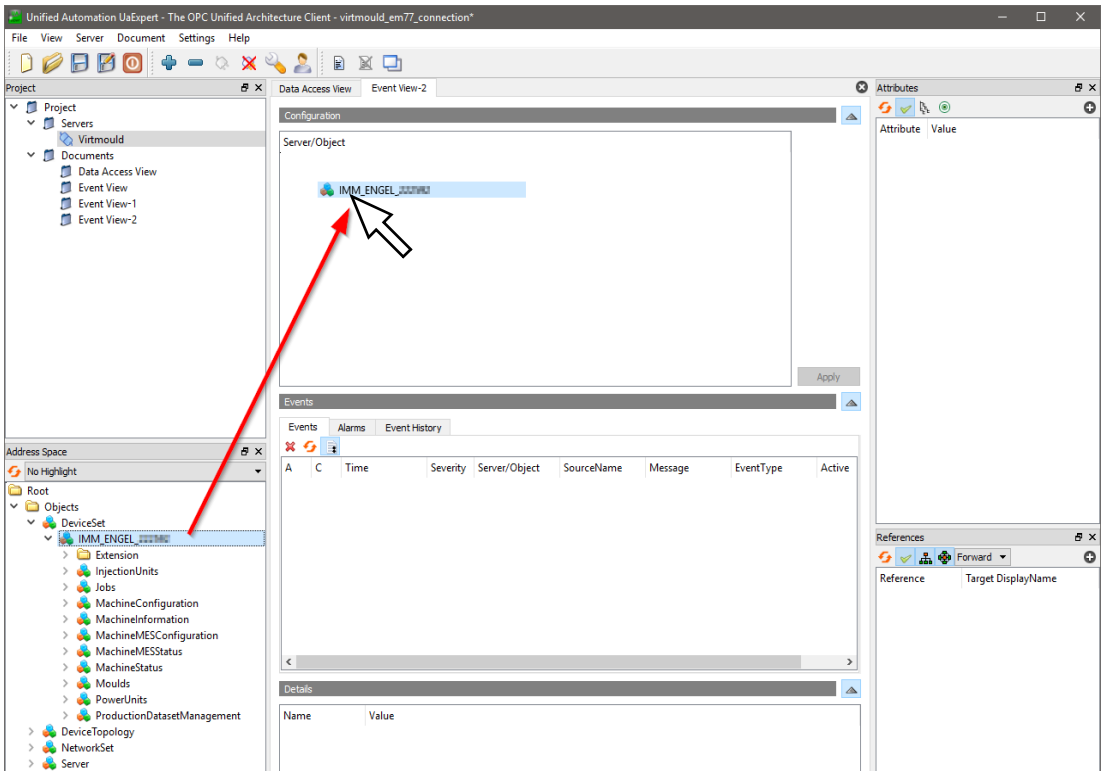
Part of the data provided by the EUROMAP 77 data interface is represented in the form of OPC UA events, e.g. cycle parameters or alarms. Access to data provided in the form of events is different from other data provided as nodes in the server's address space.
After connecting to the EUROMAP 77/OPC UA server on the injection molding machine, you need to add a new 'Document' called 'Event View' to the UaExpert GUI. In the 'Document' menu, click the 'Add...' menu item. In the dialog box, select 'Event View' as the document type and then click 'Add'.





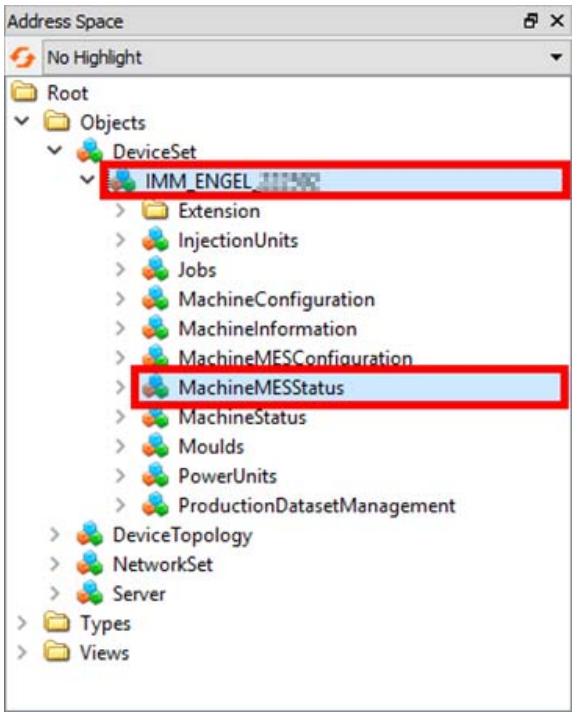
After that, a new tab called 'Event View' will be displayed in the middle of the UaExpert GUI.

Any node that acts as a creator of events can be dragged into the 'Configuration' section. According to the EUROMAP 77 specification, the main object that represents the injection molding machine /Objects/2:DeviceSet/1:IMM_ENGEL_<Serialnumber> is, among other things, an EventNotifier for the CycleParametersEventType events.

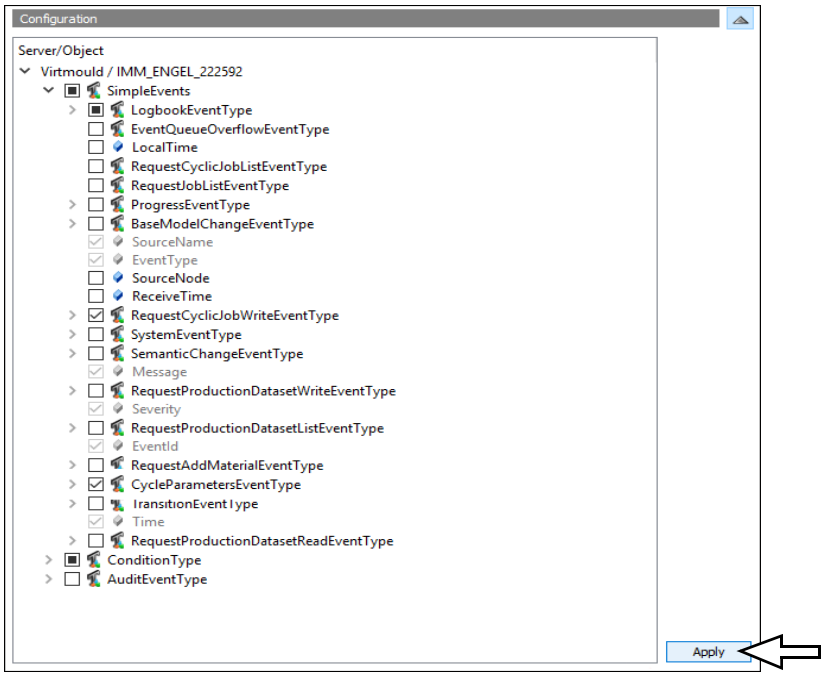


For the EUROMAP 77 interface there are two nodes that act as creators of events:

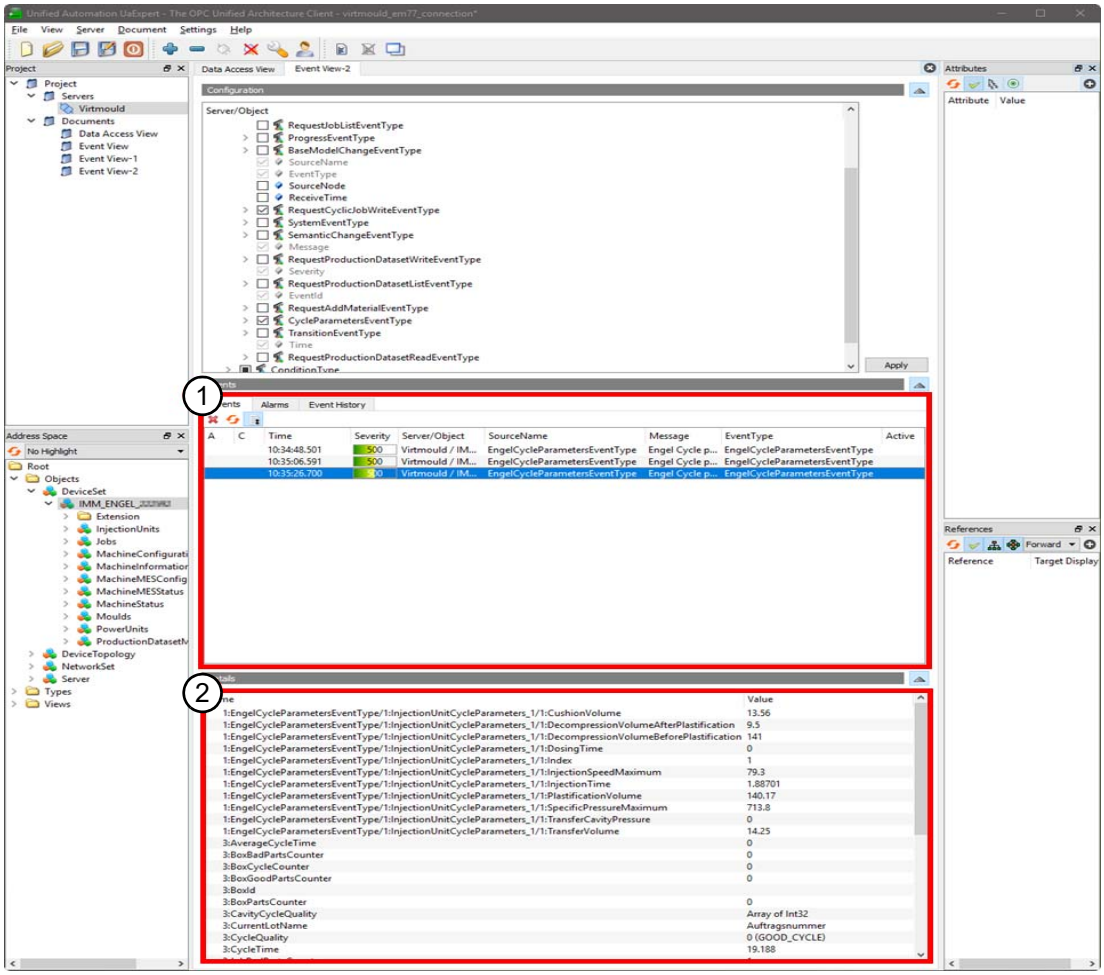
- /Objects/2:DeviceSet/1:IMM_ENGEL_<Serialnumber>
- /Objects/2:DeviceSet/1:IMM_ENGEL_<Serialnumber>/1:MachineMESStatus



To subscribe to specific events, expand the node you just dragged into the 'Event View' by clicking the arrow on the left side of the entry. You will find a hierarchical structure of events based on the 'Type heritage' (subtypes are included in their supertype). Check the boxes of the event types you want to subscribe to and then click the 'Apply' button.



The 'Events' section shows a list of these event notifications. You can select one of these events and check the data sent with the event in the 'Details' section.



[1] Events

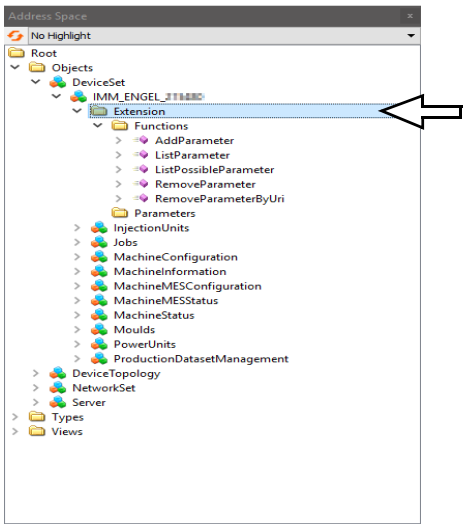
[2] Details

6.2.3 Using the ENGEL Extension

This chapter explains how to use the ENGEL specific feature ’ENGEL Extension’ in the UaExpert Client. As mentioned above, the ’ENGEL Extension’ is located inside the main computer instance /Objects/2:DeviceSet/1:IMM_ENGEL_<Serialnumber> in the ’Extension’ folder.

This folder contains two folders, one called ’Functions’ with several OPC UA methods and one called ’Parameters’. The ’Parameters’ folder is initially empty, but contains OPC UA variable nodes after parameters have been added via interaction with the methods provided in the extension.

ENGEL Extension in UaExpert

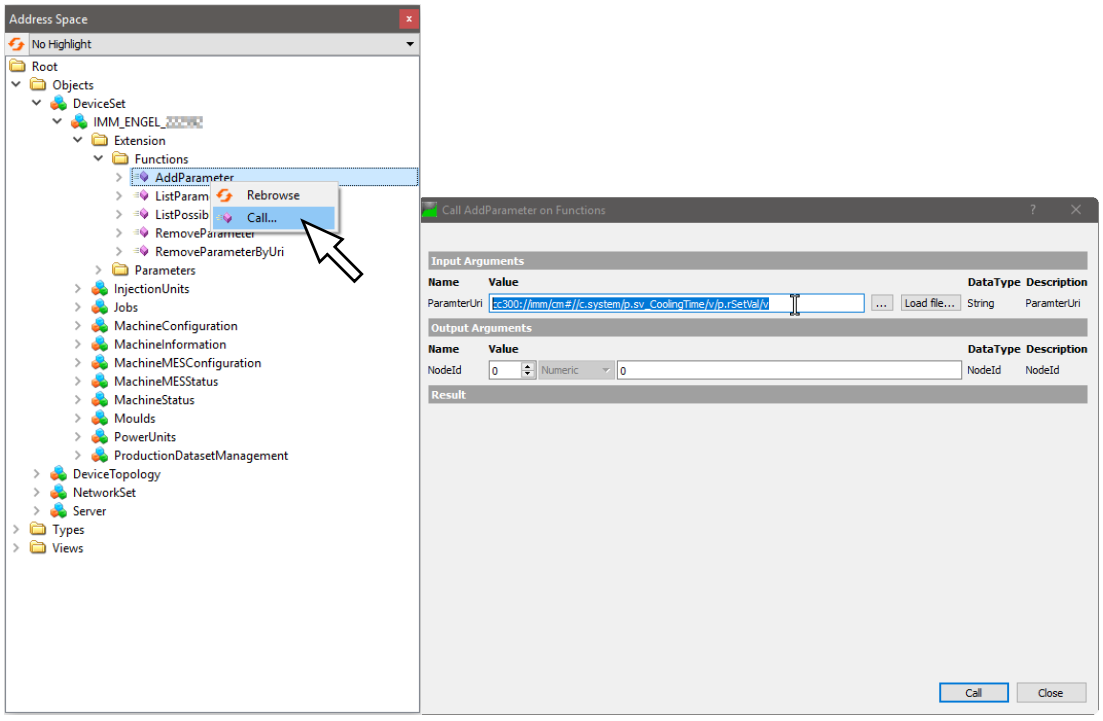


6.2.3.1 Adding a parameter with a known parameter URI

Suppose you want to add a parameter to the address space whose parameter URI is already known, for example: cc300://imm/cm#//c.system/p.sv_CoolingTime/v/p.rSetVal/v

To add a parameter, call the OPC UA method ’AddParameter’. Right-click on the method in ’Address Space’ to open the context menu and select ’Call...’. This opens a dialog box where possible input arguments for the method are to be inserted.

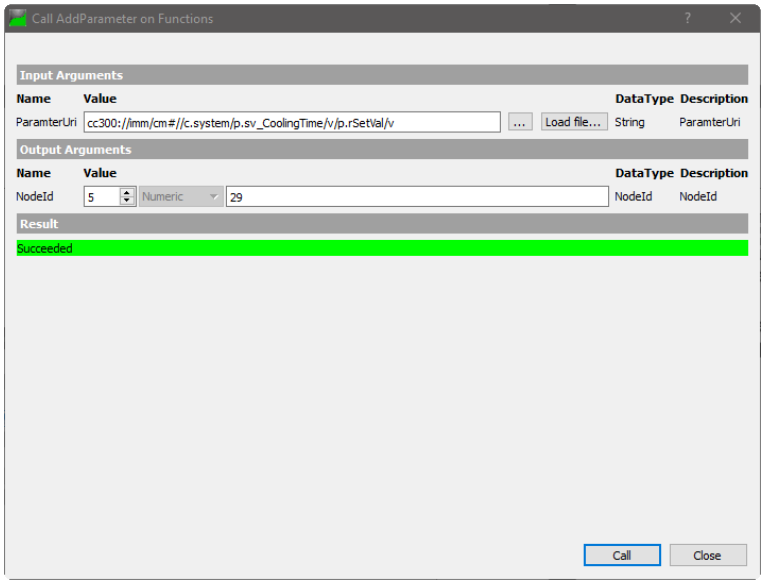
Calling AddParameter method



The AddParameter method requires an input argument called ParameterUri. After entering the URI of the above parameter, the method is called by clicking the ’Call...’ button of the dialog box.

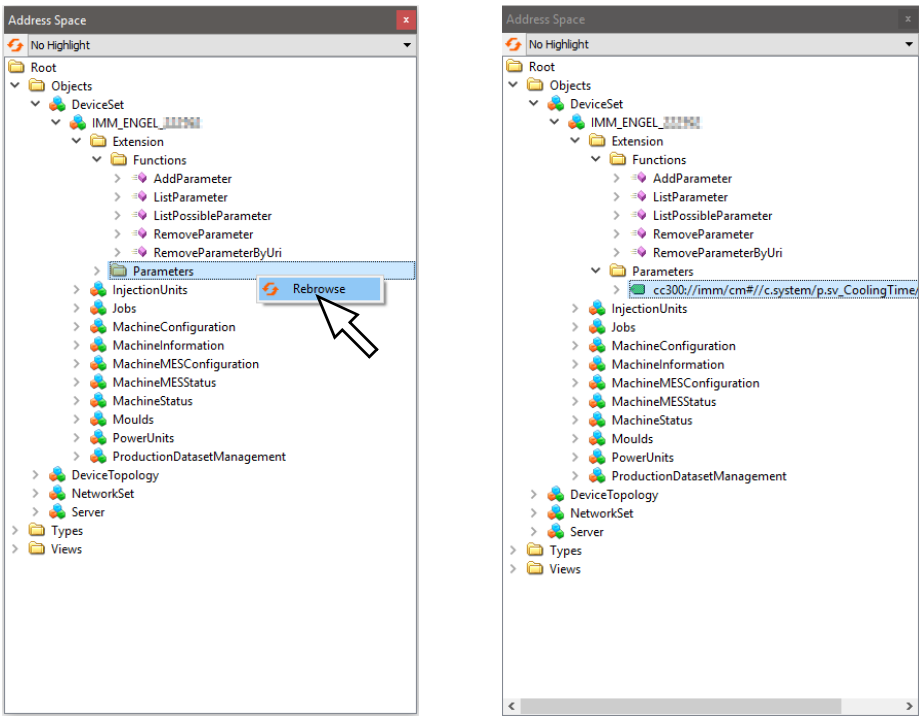
If the method is executed successfully, it returns the 'Succeeded' status as well as values for its output arguments. Otherwise, the status contains an error code indicating the reason why the method failed. The output argument of the AddParameter method is the node ID of the OPC UA variable node that was added to the address space. This node ID can be used to access the variable node directly.

Result of the AddParameter method



The newly added OPC UA variable node representing the parameter is located in the 'Parameters' folder inside the 'Extension' folder. Normally, you need to update the 'Parameters' folder to view the updated contents. Right-click the 'Parameters' folder to open its context menu and select the 'Rebrowse' item. After that, you should see the new variable node as shown below.

Updating parameters folder



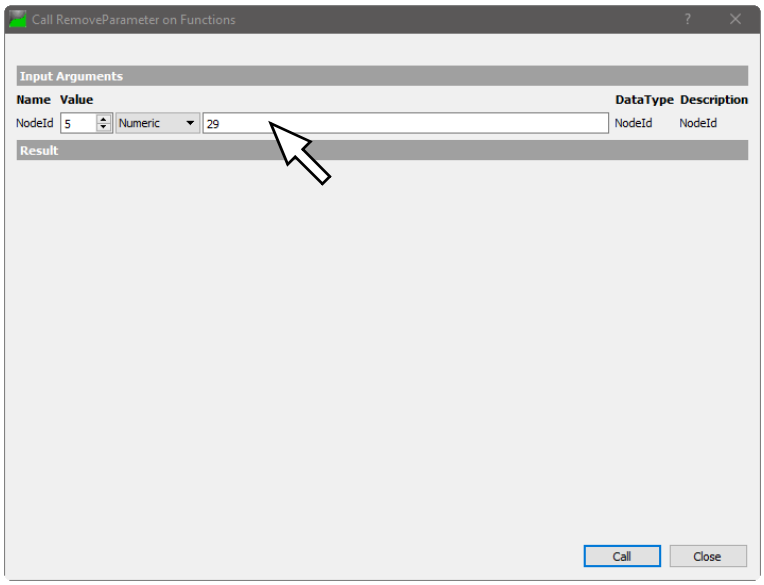
Information

Please note that the number of parameters that can be added is limited for performance reasons. Currently, this limit is 100 parameters. Therefore, you cannot add more than 100 parameters at a time.

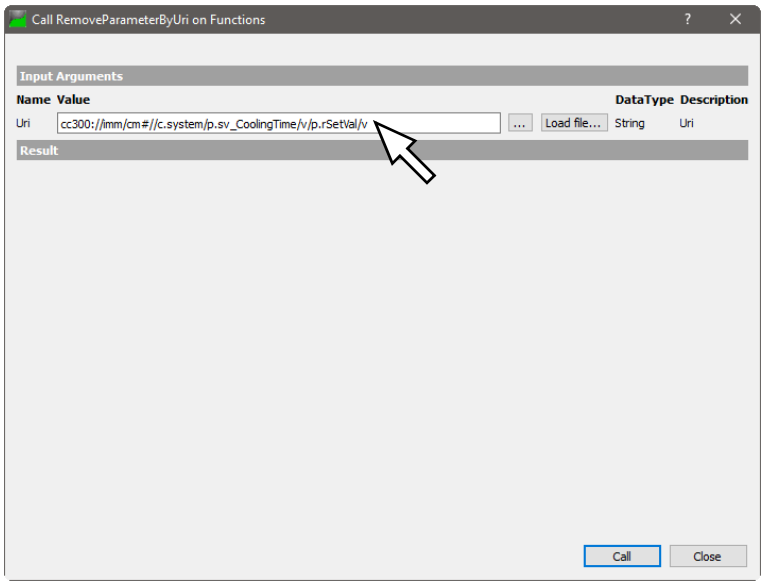
6.2.3.2 Removing a parameter

Removing a parameter added to the address space can be called either with 'RemoveParameter' or 'RemoveParameterByUri'. The former requires the node ID of the variable node that represents the parameter as an input argument. The latter requires the parameter URI as an input argument instead.
These methods have no output arguments other than the mandatory status code that indicates whether the method succeeded or failed.

RemoveParameter



RemoveParameterByUri



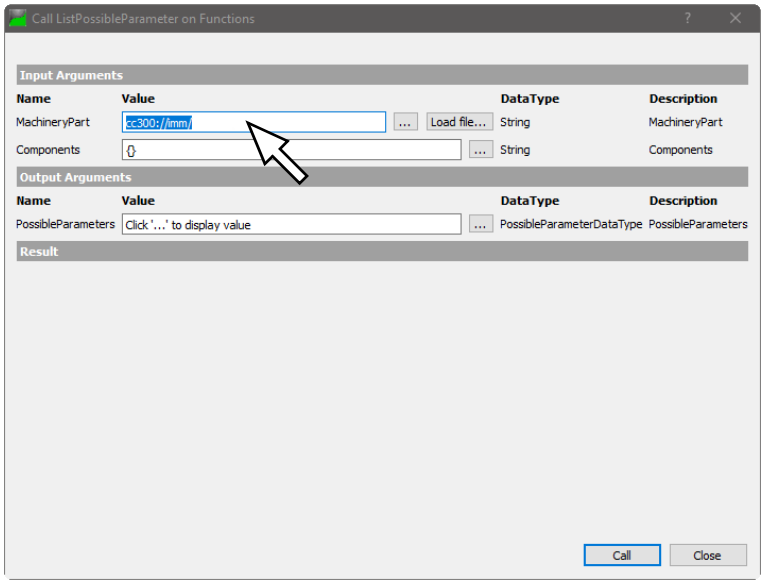
6.2.3.3 Listing the added parameters

The 'ListParameter' method returns a list of parameter URIs of all added parameters. The result is the same as when browsing the 'Parameters' folder.

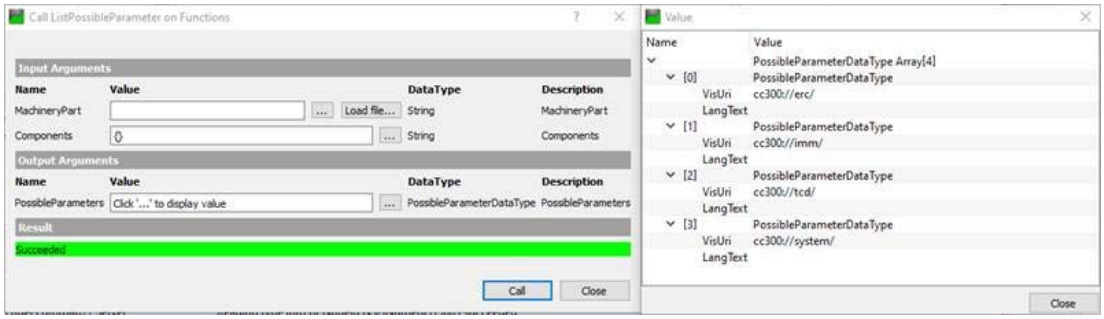
6.2.3.4 Listing possible parameters and determining the parameter URI

Using the 'ListPossibleParameter' method you can determine all available parameters and specify their parameter URI so that you can add them by calling the 'AddParameter' method. [See ENGEL Extensions on page 44.]

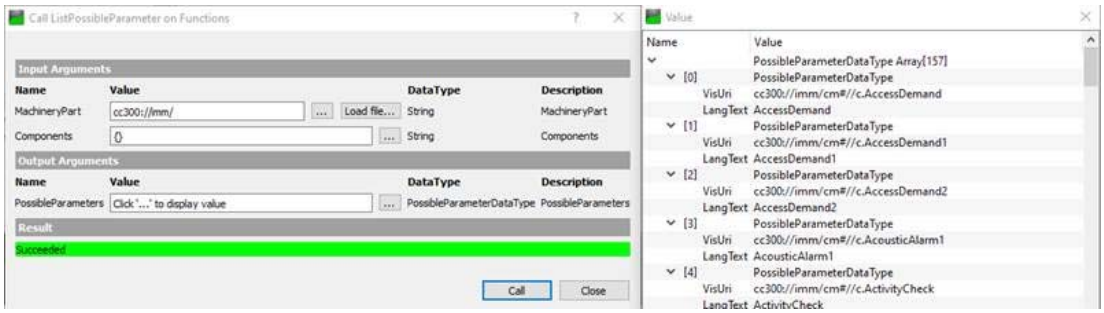
ListPossibleParameter



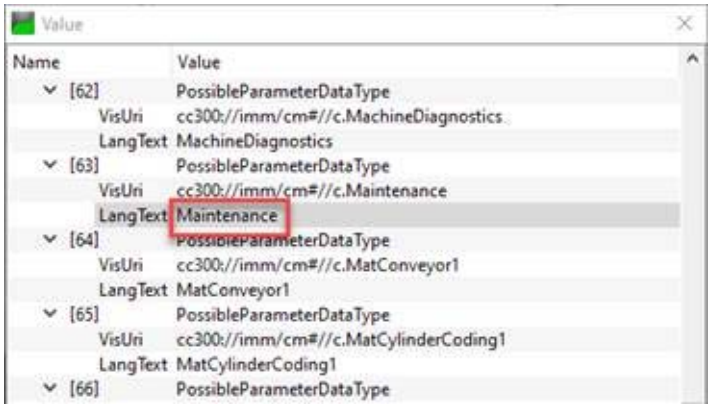
Call the 'ListPossibleParameter' function without input values to retrieve the available 'MachineryParts'.



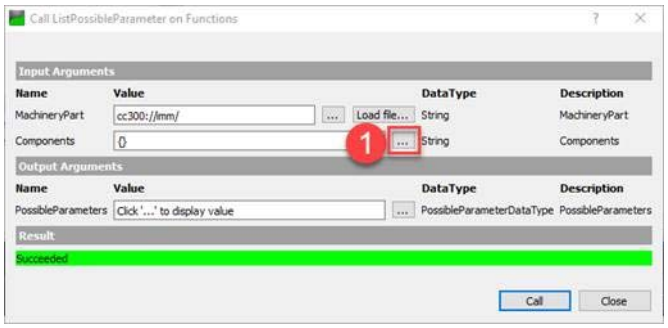
Call the function with the correct 'MachineryPart' to retrieve the underlying components.



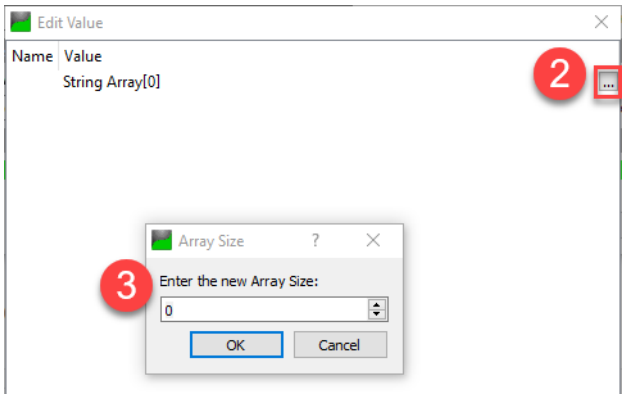
Search the long text of the selected component in the output.



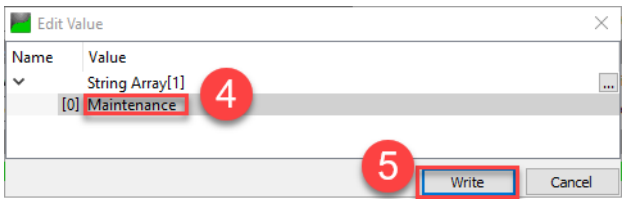
Access the Edit value function of the Components field.



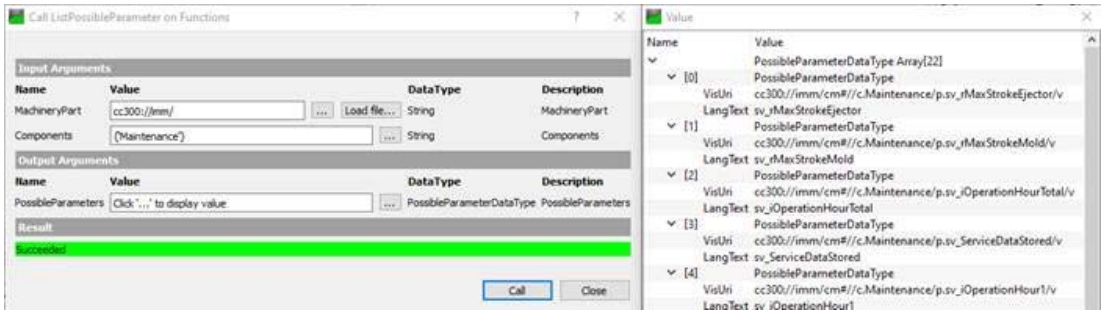
Enter the array size (1 - n) depending on how many components are to be requested.



Enter the long text and press Write.



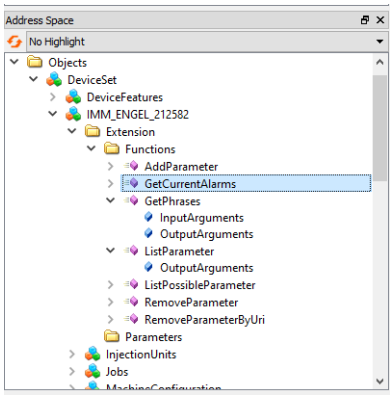
Call the function and it will output all possible parameters under the selected component.

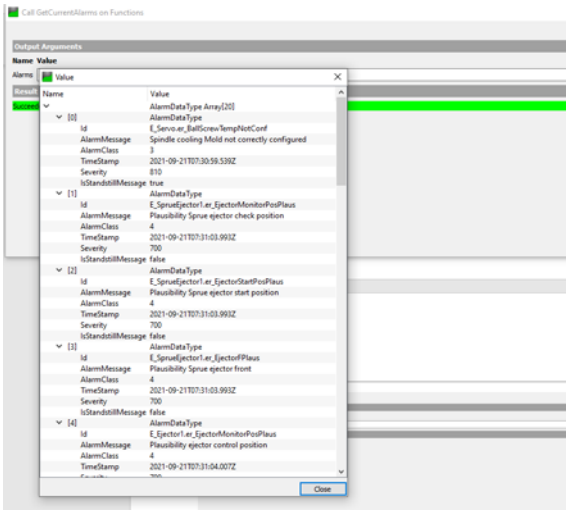


6.3 Current alarm display.

The current alarms are mapped in the ’ENGEL Extensions’. They can be retrieved again and again by the client at a freely definable interval.

Right-click on the ’GetCurrentAlarms’ function to make a ’Call’ without parameters. The currently pending alarms are then displayed in the output arguments (see the following figures).




Call GetCurrentAlarms on Functions
Output Arguments
Name Value
Alarms
Result
Success
Value
Name
Value
AlarmDataType Array[20]
[0]
AlarmDataType
Id
E_Servo.er_BallScrewTempNotConf
AlarmMessage
Spindle cooling Mold not correctly configured
AlarmClass
3
TimeStamp
2021-09-21T07:30:59.539Z
Severity
810
IsStandstillMessage
true
[1]
AlarmDataType
Id
E_SprueEjector1.er_EjectorMonitorPosPlaus
AlarmMessage
Plausibility Sprue ejector check position
AlarmClass
4
TimeStamp
2021-09-21T07:31:03.993Z
Severity
700
IsStandstillMessage
false
[2]
AlarmDataType
Id
E_SprueEjector1.er_EjectorStartPosPlaus
AlarmMessage
Plausibility Sprue ejector start position
AlarmClass
4
TimeStamp
2021-09-21T07:31:03.993Z
Severity
700
IsStandstillMessage
false
[3]
AlarmDataType
Id
E_SprueEjector1.er_EjectorFPlaus
AlarmMessage
Plausibility Sprue ejector front
AlarmClass
4
TimeStamp
2021-09-21T07:31:03.993Z
Severity
700
IsStandstillMessage
false
[4]
AlarmDataType
Id
E_Ejector1.er_EjectorMonitorPosPlaus
AlarmMessage
Plausibility ejector control position
AlarmClass
4
TimeStamp
2021-09-21T07:31:04.007Z
Close

7 Use case - production data log

A very common use case for the EUROMAP 77 data interface is the continuous acquisition of production data for each production cycle of the injection molding machine.

7.1 Specification

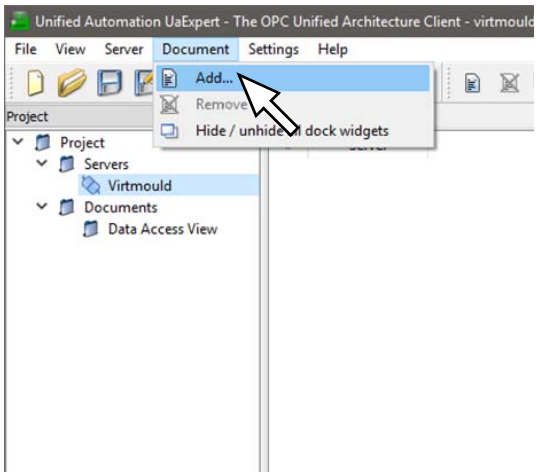
The EUROMAP 77 specification stipulates that this should be done via OPC UA events. The specification defines the CycleParametersEventType, which contains basic information about the production cycle, such as cycle time, cycle quality, etc. For more details, see Chapter 16 of the EUROMAP 83, Release 1.01, January 28, 2019 document.
The specification requires the actual EUROMAP 77 implementation to create a subtype of CycleParametersEventType, which should be extended by an InjectionUnitCycleParameters-Type instance for each injection unit of the injection molding machine and a MouldCyclePara-metersType instance for each mold of the injection molding machine (see Chapter 16.14 of the EUROMAP 83, Release 1.01, January 28, 2019document). In the ENGEL implementation the BrowseName of this subtype is EngelCycleParametersEventType.

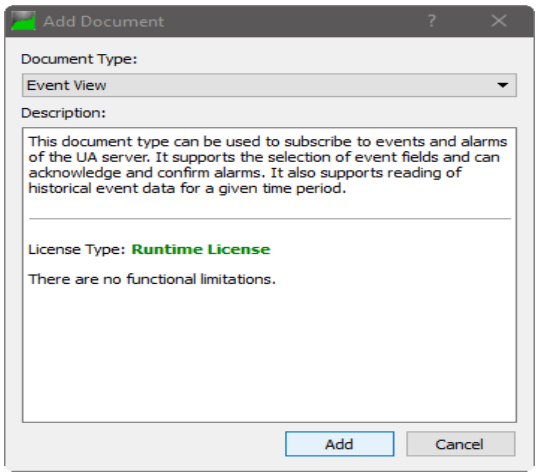
7.2 Subscribing to the events

This chapter shows how to access CycleParametersEventType event data, but this can be done for other events in a similar way. This showcase is done with the UaExpert client software from Unified Automations GmbH. This software is not a product of ENGEL and therefore not in the scope of delivery of our EUROMAP 77 implementation. The following screenshots only serve to show the general process of subscribing to events in a very commonly used OPC UA client. Experiences with other OPC UA clients may vary.

7.2.1 Step 1 - Opening event view

After connecting to the EUROMAP 77/OPC UA server on the injection molding machine, you need to add a new 'Document' called 'Event View' to the UaExpert GUI. In the 'Document' menu, click the 'Add...' menu item. In the dialog box, select 'Event View' as the document type and then click 'Add'.

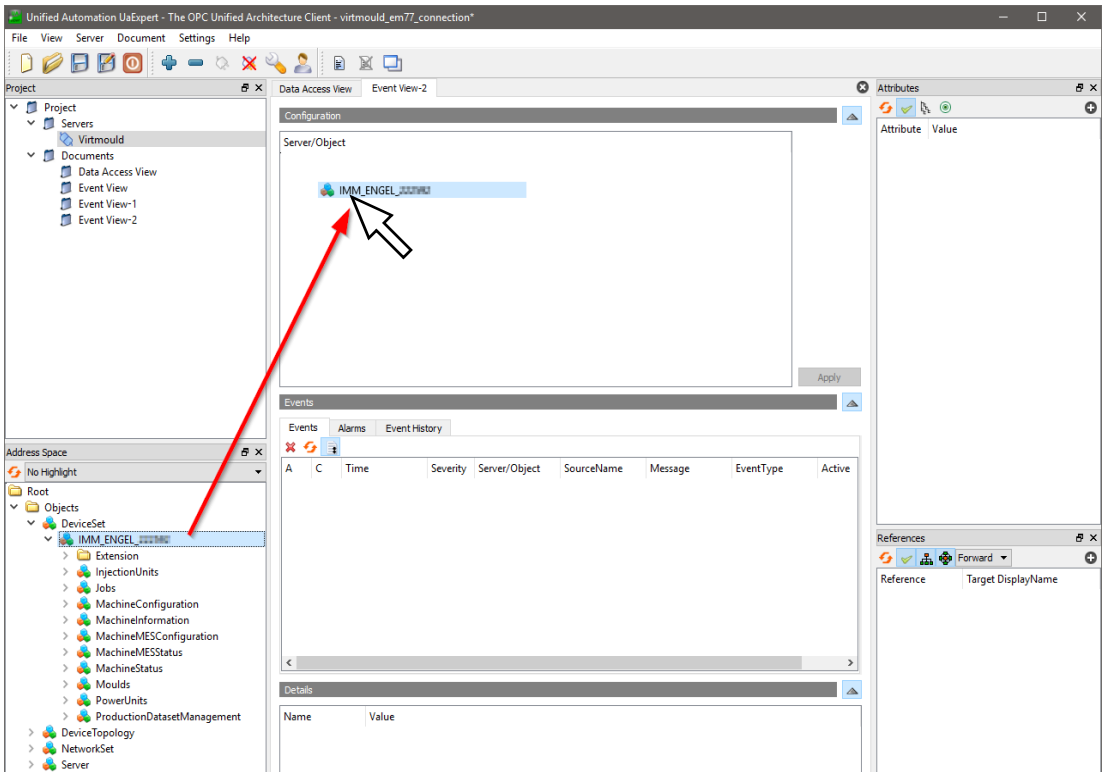




After that, you should see a new tab called 'Event View' in the middle of the UaExpert GUI.

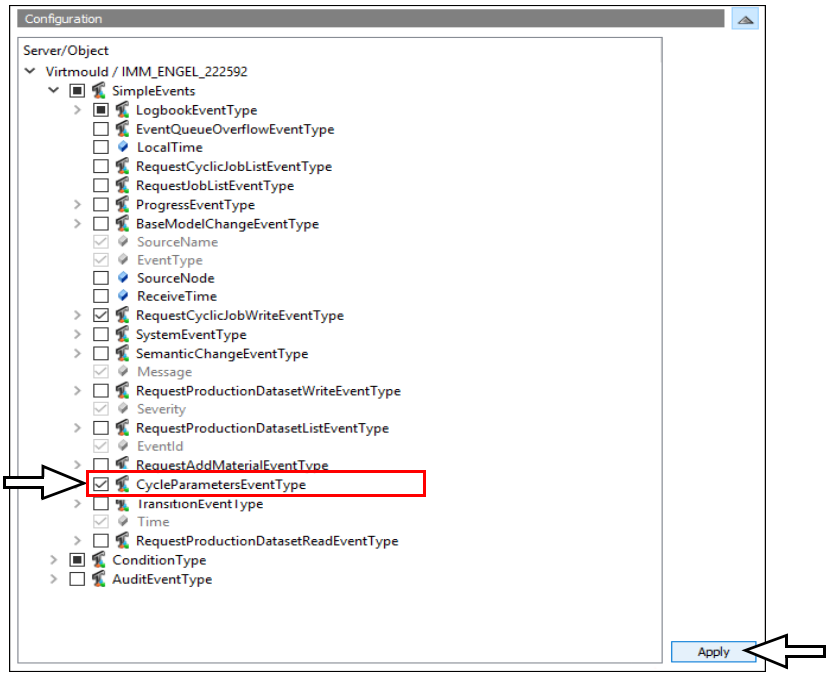
7.2.2 Step 2 - Dragging event notifier to Event View

Any node that acts as a creator of events can be dragged into the ’Configuration’ section.



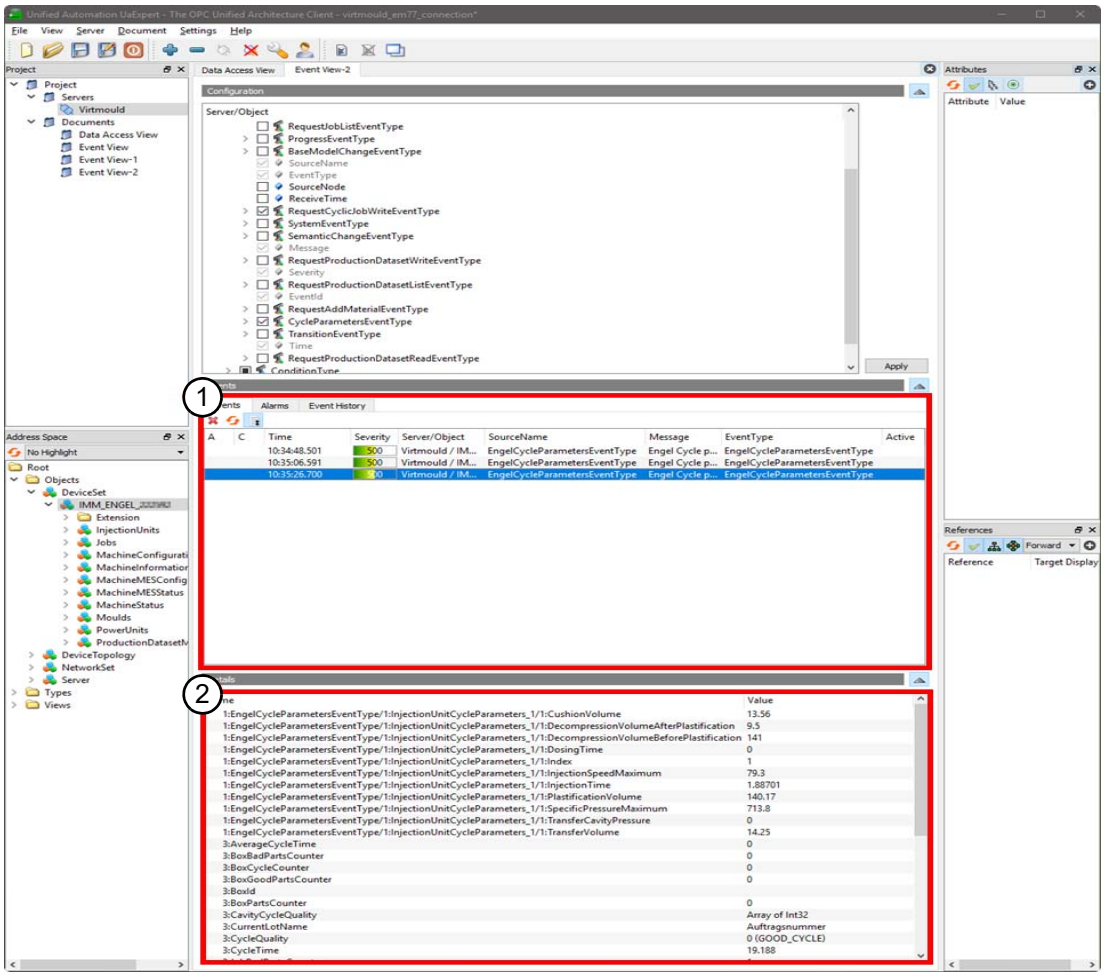
7.2.3 Step 3 - Subscribing to the CycleParametersEventType event

To subscribe to specific events, expand the node you just dragged into the ’Event View’ by clicking the arrow on the left side of the entry. You will find a hierarchical structure of events based on the ’Type heritage’ (subtypes are included in their supertype). Check the boxes of the event types you want to subscribe to and then click the ’Apply’ button.



7.2.4 Receiving the events

After subscribing to the CycleParametersEventType events, the OPC UA client receives a notification when the computer completes a production cycle. The 'Events' section shows a list of these event notifications. You can select one of these events and check the data sent with the event in the 'Details' section. The events occur as EngelCycleParametersEventType (a subtype of CycleParametersEventType) and contain the production parameters.



[1] Events

[2] Details

7.3 Event extension

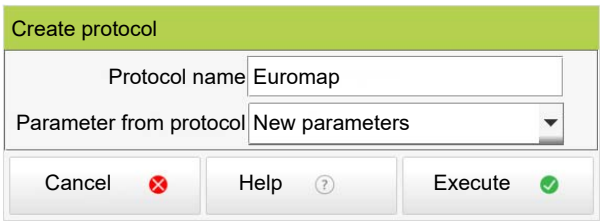
The data provided with the CycleParametersEventType events is limited to commonly available parameters for different manufacturers. The ENGEL control unit software could possibly provide many more parameters than specified in the EUROMAP 77 specification.
Therefore, the ENGEL implementation of EUROMAP 77 provides a mechanism to extend the CycleParametersEventType events with ENGEL specific parameters. Below there is a description of how to extend the CycleParametersEventType events with ENGEL specific parameters.

7.3.1 Create protocol

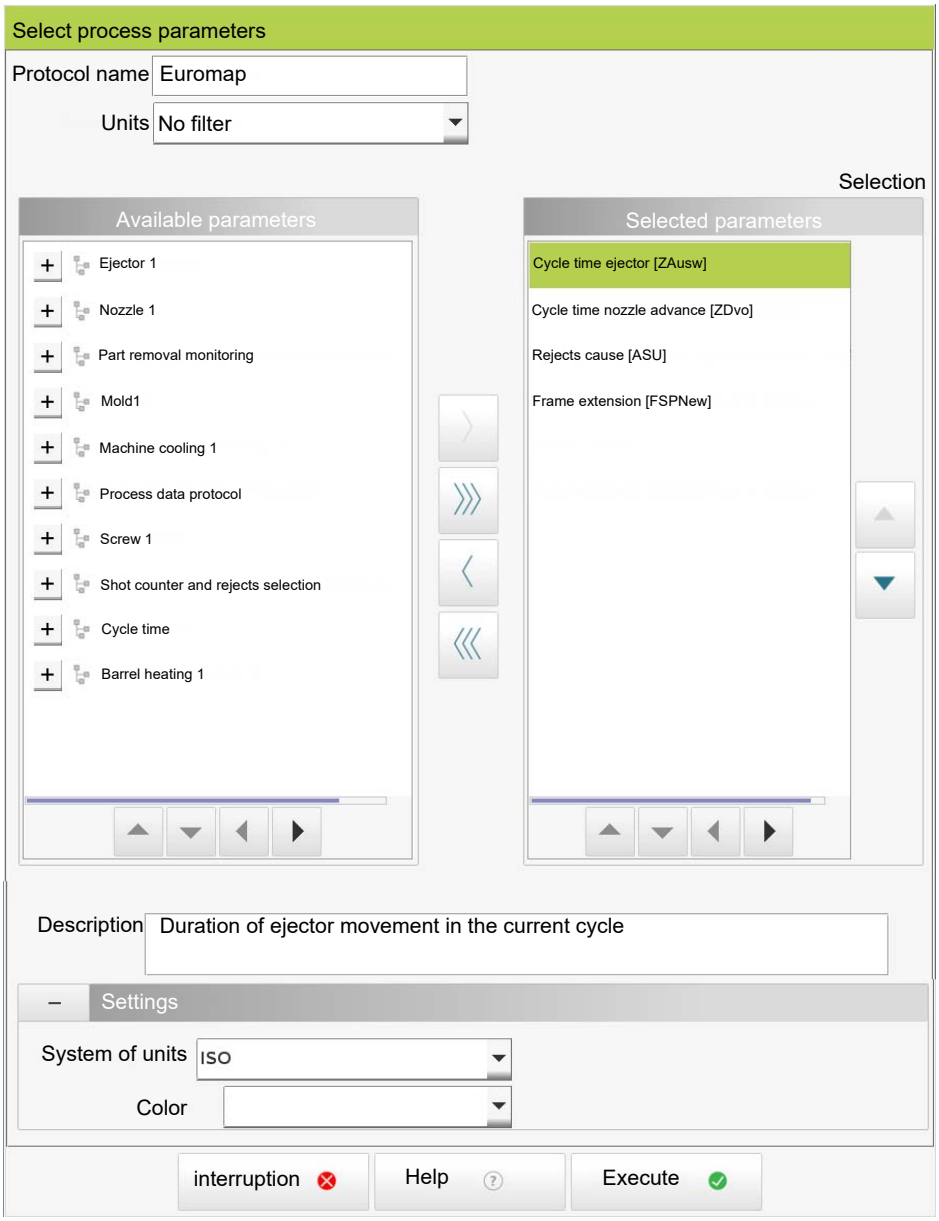
1. Log on to the machine control panel with user level 5.
2. Open the Process data screen.

3. Select **[Protocol]** and **[Create protocol]**.

4. Enter a **protocol name**.
Use upper/lower case letters as appropriate



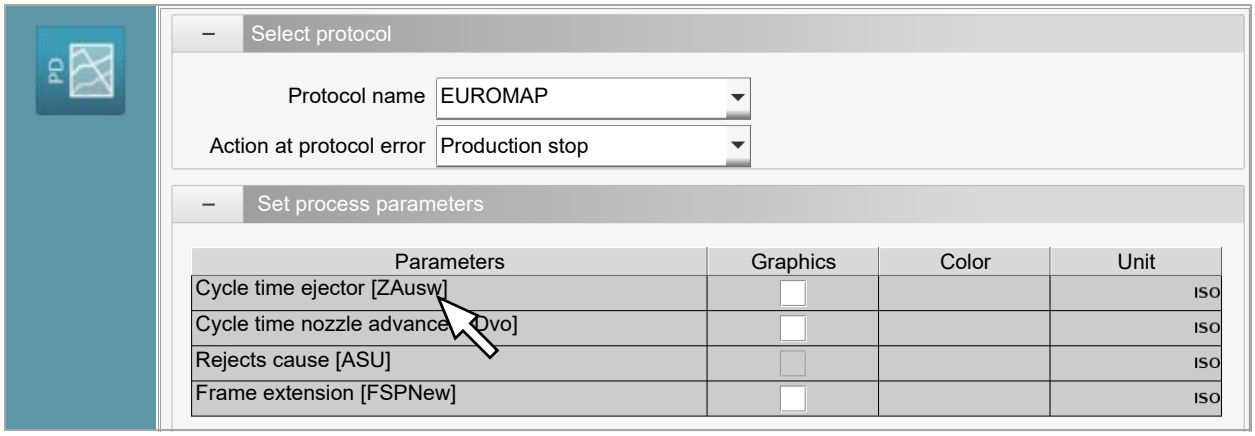
5. Choose whether to create a new parameter list or if you want to use an existing one.
6. Choose **[Execute]**.
7. Select an available parameter or an entire parameter group.
Filters for 'Available parameters' are available in the **Units** selection field.



8. By tapping an **Available parameter** and pressing the arrow key**[>]**, it is shifted to the **Selected parameters** and vice-versa. The **[>>>]** and **[<<<]** keys shift all parameters.
9. If desired, select a system of units under **Settings** and color for the respective **Selected parameter**.
10. Adjust the order of the parameters using the arrow keys next to the **Selected parameters** window.
11. Press **[execute]** to accept a set parameter.

7.3.2 Identifying parameters

The added parameters should now be included in your OPC UA client software. They are added with their English long text and the German abbreviation names.
On the 'Process data' screen page, you can look up the parameter name with the help of the abbreviation name. The abbreviation name is displayed in square brackets after the parameter name.



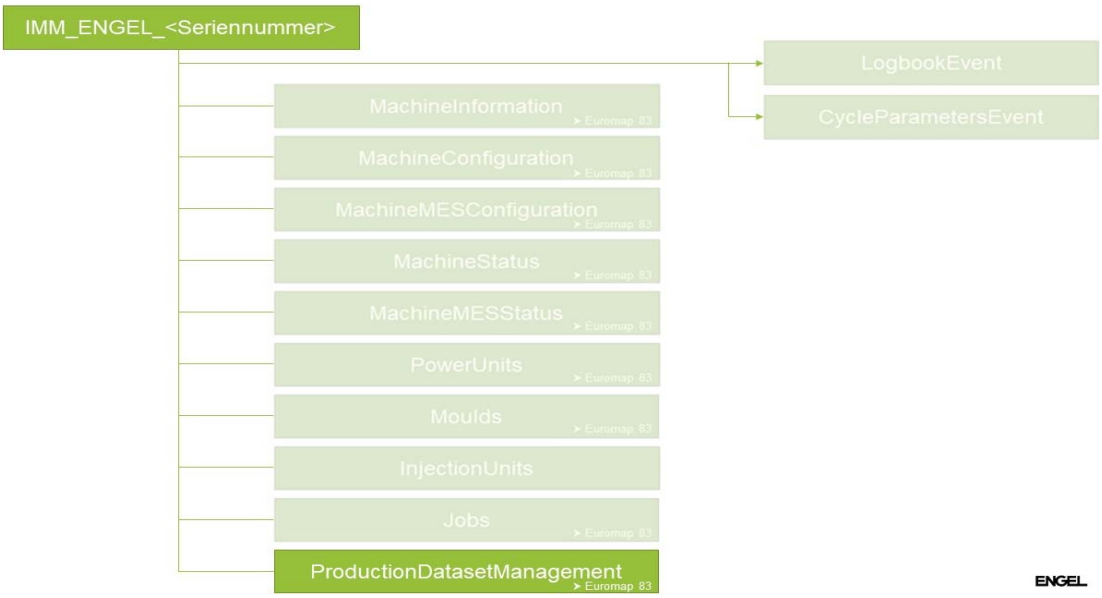
8 Use case - transfer of production data sets

The EUROMAP 77 data interface has the 'Transfer of production data sets' function so that the client software can transfer production data sets of the injection molding machine from and to the Manufacturing Execution System (MES).
The transfer of the production data set is based on the File Transfer API of the OPC UA specification. This document describes a common method for performing a file transfer via the EUROMAP 77 data interface. For further information, please refer to EUROMAP 83 recommendation Release 1.01 in Chapter 17.5 as well as to OPC UA Specification Part 5: Information Model Release 1.04 in Appendix C.

8.1 Transfer of production data sets

The ProductionDatasetTransfer object, located in the ProductionDatasetManagement object, is the core for the data set transfer. It provides several methods for data set transfer, most importantly GenerateFileForRead, GenerateFileForWrite and CloseAndCommit.

Storage location of the ProductionDatasetManagement object



8.2 Transferring parts data from the injection molding machine to the MES

For transferring a data set file of the injection molding machine to the MES, the client software starts by calling the GenerateFileForRead method. GenerateFileForRead uses an input argument of type ProductionDatasetReadOptionsType, which contains two variables that determine the source of the injection molding machine: storage and name.

ProductionDatasetReadOptionsType
Storage: Integral/enumeration Name: String

Storage is an integral representing an enumeration and can have the following values: PRODUCTION (integral value 1) or FILE_SYSTEM (integral value 4). If you set storage to PRODUCTION, the currently loaded production data set is transferred to the MES. If storage is set to PRODUCTION, the 'Name' variable is ignored and possibly omitted. If you set storage to FILE_SYSTEM, a data set file stored in the local file system of the injection molding machine is transferred to the MES.

If storage is set to FILE_SYSTEM, the 'Name' variable determines the file name of the data set file in the local file system (without the file extension '.partdata'). Please note that the file system on the injection molding machine is case sensitive with respect to the file name.

Example:

By calling GenerateFileForRead with the parameters Storage=4 (FILE_SYSTEM) and Name=xyz you can initiate the transfer of the data set file 'xyz.partdata' from the local file system of the injection molding machine.

If there is no data set file with the specified name in the local file system of the injection molding machine, the method fails with the error code BadInvalidArgument.
By calling the GenerateFileForRead method, a temporary object of the FileType is created in the server display area. This temporary object represents the file specified by the input arguments. The output arguments of the GenerateFileForRead method are:

1. fileNodeId: the node ID of the temporary object
2. fileHandle: a file handle of the temporary file object that will be needed later for subsequent method calls
3. completionStateMachine: the node ID to an object of the FileTransferStateMachineType

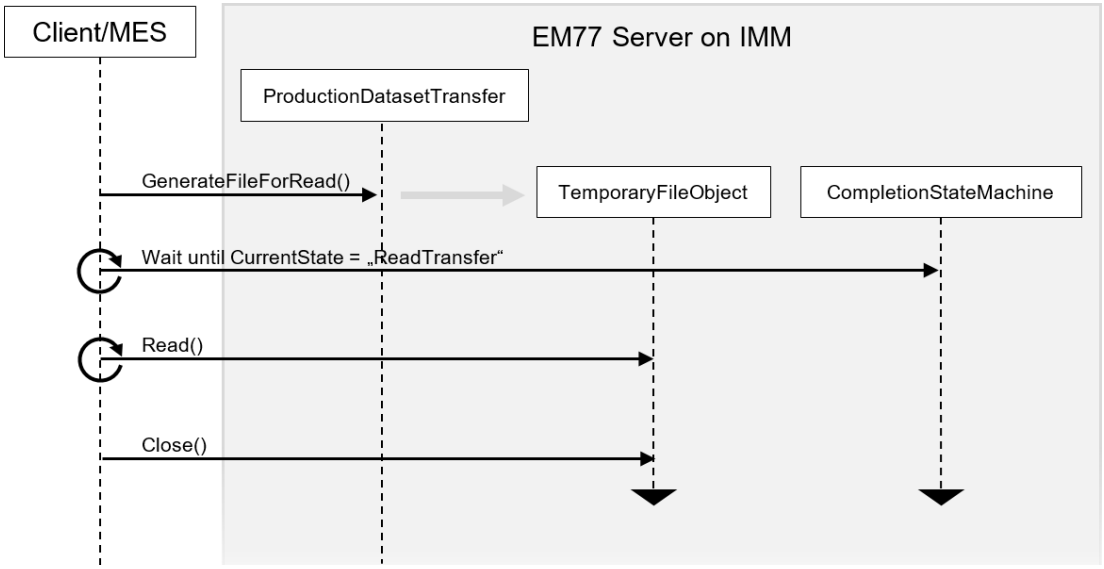
The third output argument completionStateMachine is optional and can be zero. If it is zero, you can continue reading the file contents.

Check status
If the third output argument completionStateMachine is not zero, the node ID refers to a temporary object of FileTransferStateMachineType that indicates whether the source is ready for reading. This is necessary if the source of the file transfer is PRODUCTION, because the injection molding machine usually has to prepare the data before it can transfer it.
Therefore, before reading the file contents, the client should wait until the value of the CurrentState variable of the CompletionStateMachine object has the value 'ReadTransfer'. This status indicates that the file is ready to be read. At this point, no data transfer has taken place.

Read
To retrieve the contents of the file, call the Read method of the temporary file object. The input arguments of the Read method are the file handle (output argument of GenerateFileForRead) and the number of bytes to read.
The output argument is a byte array representing the contents of the file. You can process this byte array (e.g. write it to a database or a file in the file system of the MES). If the file is larger than the specified number of bytes, you must call the Read method again to receive the next byte block. Repeat this process until the byte array from the output arguments contains no more bytes (an empty byte array indicates that the end of the file has been reached).

Close
When you are done reading, call the close method of the temporary file object with fileHandle as the input argument. This causes the server to clean up the temporary object.

Fig. 3 - Transfer parts data from MES to injection molding machine



8.3 Transferring parts data from the MES to the injection molding machine

GenerateFileForWrite

To transfer a data set file from the MES to the injection molding machine, the client software first calls the GenerateFileForWrite method. GenerateFileForWrite uses a GenerateOptions input argument of type ProductionDatasetWriteOptionsType, which contains three variables that determine the destination on the injection molding machine: Storage, Name and Components.

ProductionDatasetWriteOptionsType
Storage: Integral/enumeration Name: String Components: ProductionDatasetComponentEnumeration[]

Storage and Name work the same way as in the GenerateFileForRead method, except that they point to the injection molding machine's file transfer destination (rather than the source). So if you set storage to PRODUCTION, the transferred file will be activated as the current production setting of the injection molding machine (without saving it in the local file system of the injection molding machine). If storage is set to PRODUCTION, the ’Name’ variable is ignored and possibly omitted. If you set storage to FILE_SYSTEM, the transferred parts data file is stored in the local file system of the injection molding machine.
If storage is set to FILE_SYSTEM, the Name variable determines the file name of the parts data file in the local file system (again, without the file extension ’.partdata’). Please note that the file system on the injection molding machine is case sensitive with respect to the file name.
The Components input argument is an array of ProductionDatasetComponentEnumeration that can have one of two values: IMM_0 or ROBOT_1. It specifies which parts of the production data set are to be activated in the machine control unit after the data set file has been transferred, so it is only relevant if storage is set to PRODUCTION. On machines with integrated robots, production data sets can contain settings for the robot in addition to the settings for the injection molding machine. With the ’Components’ argument, you can select which part of the data set should be activated. You can activate only the injection molding machine part, only the robot part, or both. If Components is set to an empty array (array length is 0), the complete production data set is activated.

Example:

By calling GenerateFileForWrite with the parameters Storage=4 (FILE_SYSTEM) and Name=abc you can, for example, initiate the transfer to the parts data file 'abc.partdata' on the local file system of the injection molding machine.

If a data set with the same name already exists, the method fails with the BadUserAccessDenied error code.
By calling the GenerateFileForWrite method, a temporary file object is created in the server's address space. This temporary file object represents the file specified by the input arguments. The output arguments of the GenerateFileForWrite method are:

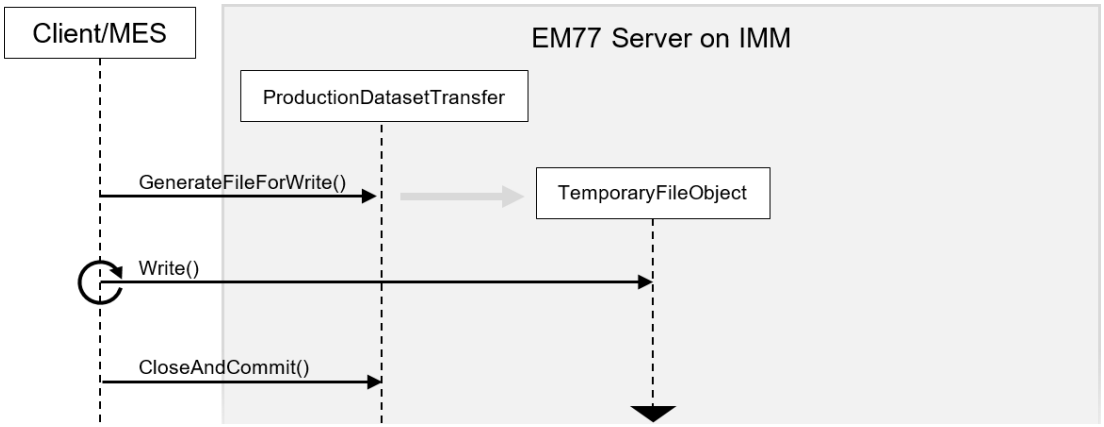
1. fileNodeId: the node ID of the temporary object
2. fileHandle: a file handle of the temporary file object that will be needed later for subsequent method calls

At this point, no data transfer has taken place.

Write
To send the file contents, call the Write method of the temporary file object. The input arguments of the Write method are fileHandle (output argument of GenerateFileForWrite) and the data as ByteString. You can split the file content into several byte blocks and transfer them one after the other. Repeat this process until the complete file content has been transferred.

CloseAndCommit or Close
When you are done writing the file contents, call the CloseAndCommit method of the ProductionDatasetTransfer object with fileHandle as the input argument. This causes the server to save or activate the transferred data set and clean up the temporary file object.
Note that calling the Close method of the temporary file object cancels the file transfer and the data transferred up to that point is discarded.

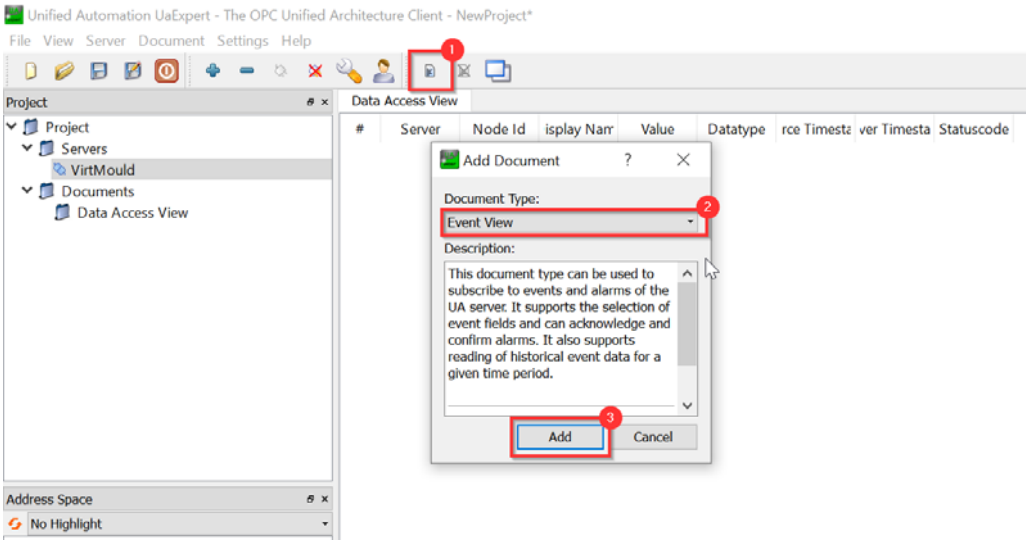
Transferring parts data from MES to injection molding machine



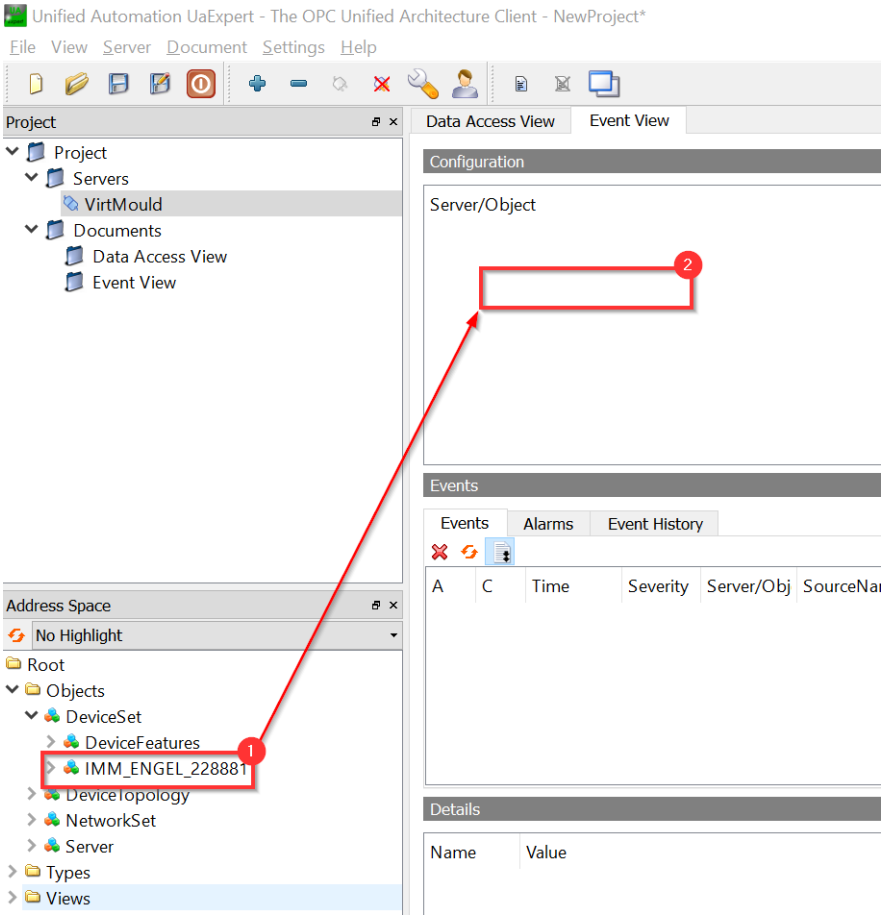
9 Use case - display of set value changes

This chapter uses UaExpert to demonstrate how set value changes can be retrieved from the control unit.

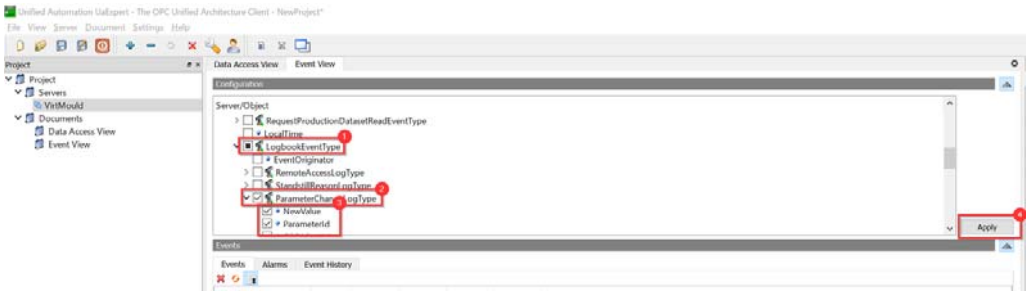
1. Open Event-View.



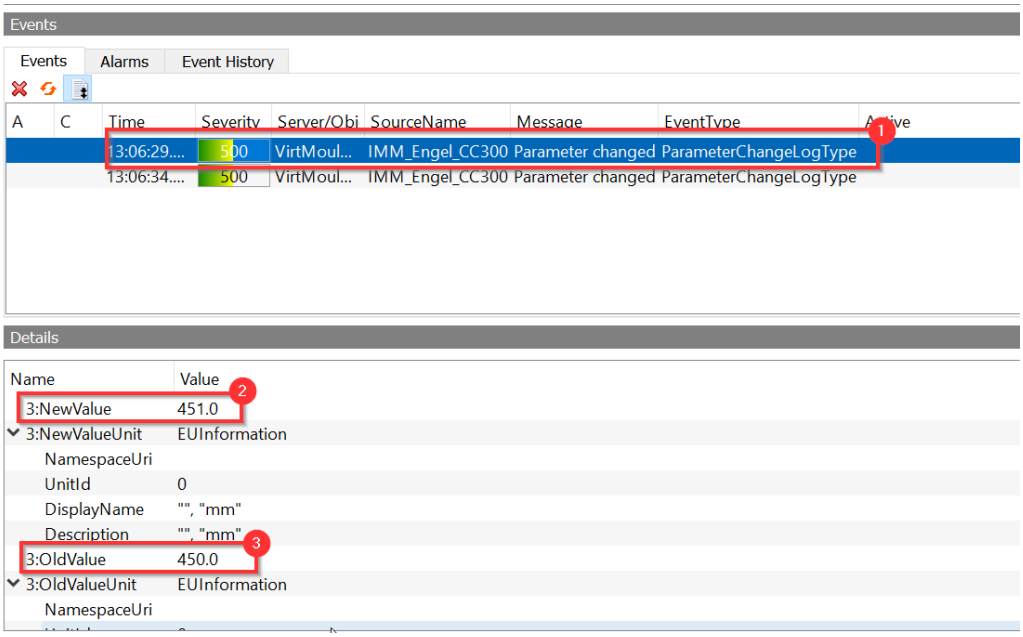
2. From the ’DeviceSet’, drag the machine object into the ’Event View’.



3. Select the appropriate ’EventType’ and activate it.



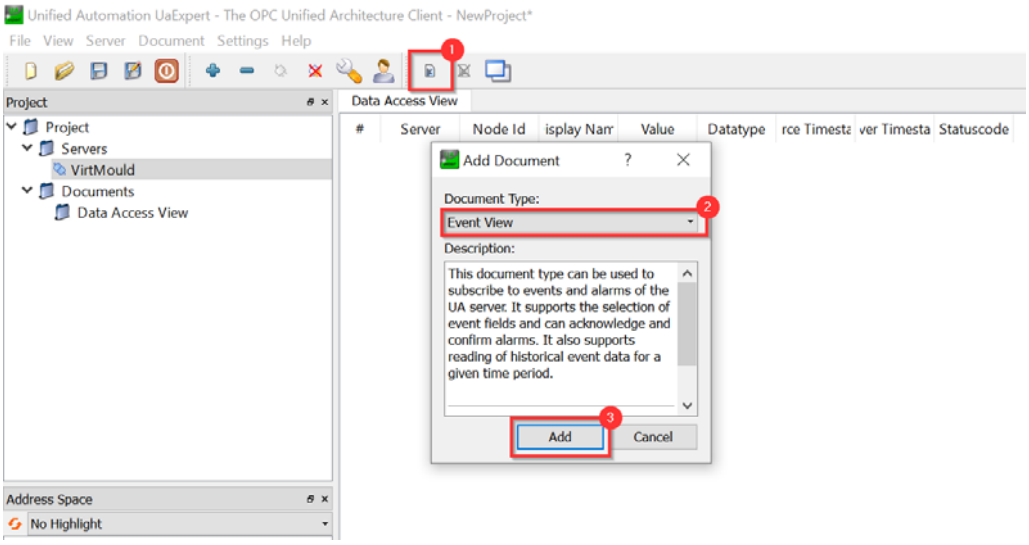
4. The value changes that follow from now on are now displayed.



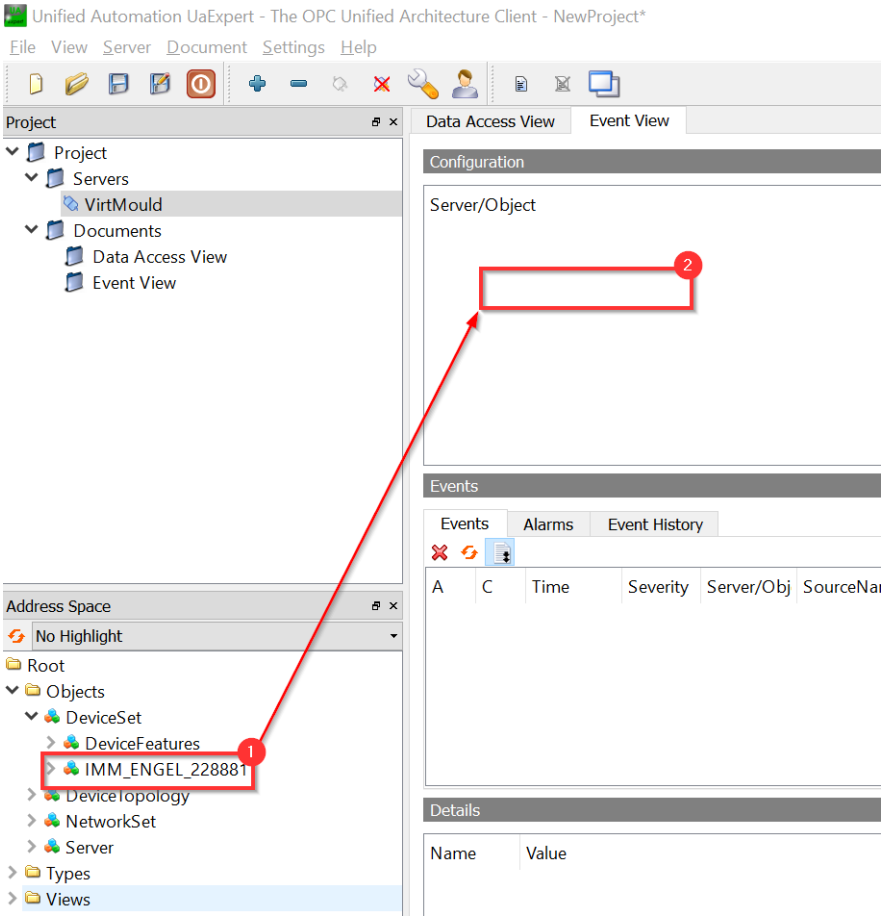
10 Use case - display of historical alarms

Historical occurring and cleared alarms are supplied by 'LogbookEventType'.

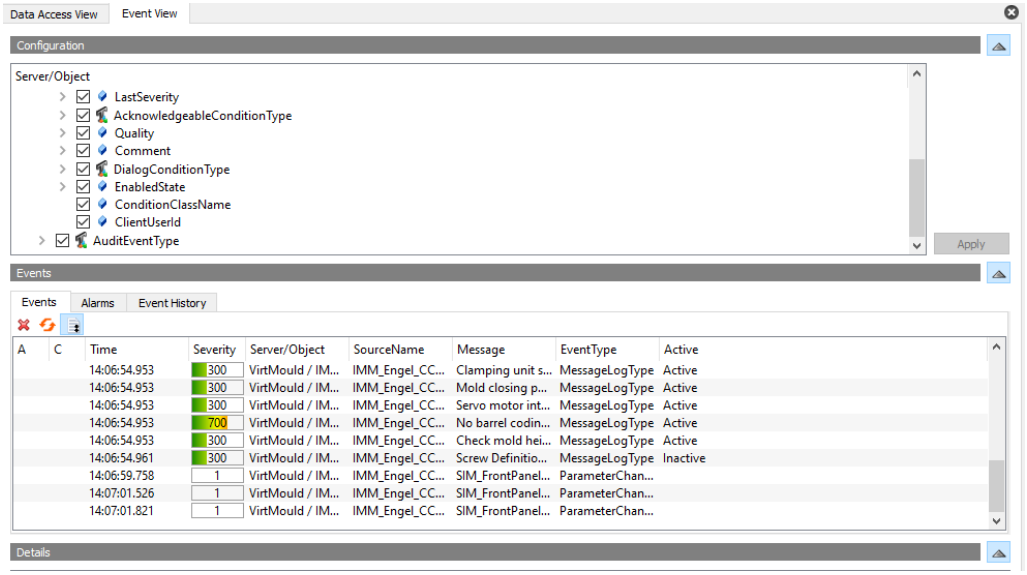
1. Open Event-View.



2. From the DeviceSet, drag the machine object into the Event View



3. Select all sources in the event view. Only the data that is also sent by sources is displayed.



11 Behavior with Browse-Path

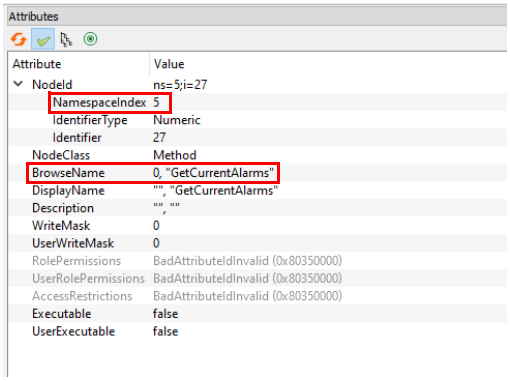
Example:

```
[ObjectsFolder]/2:DeviceSet/1:IMM_ENGEL_218828/4:Jobs/3:ActiveJobValues/3:JobPartsCounter
```

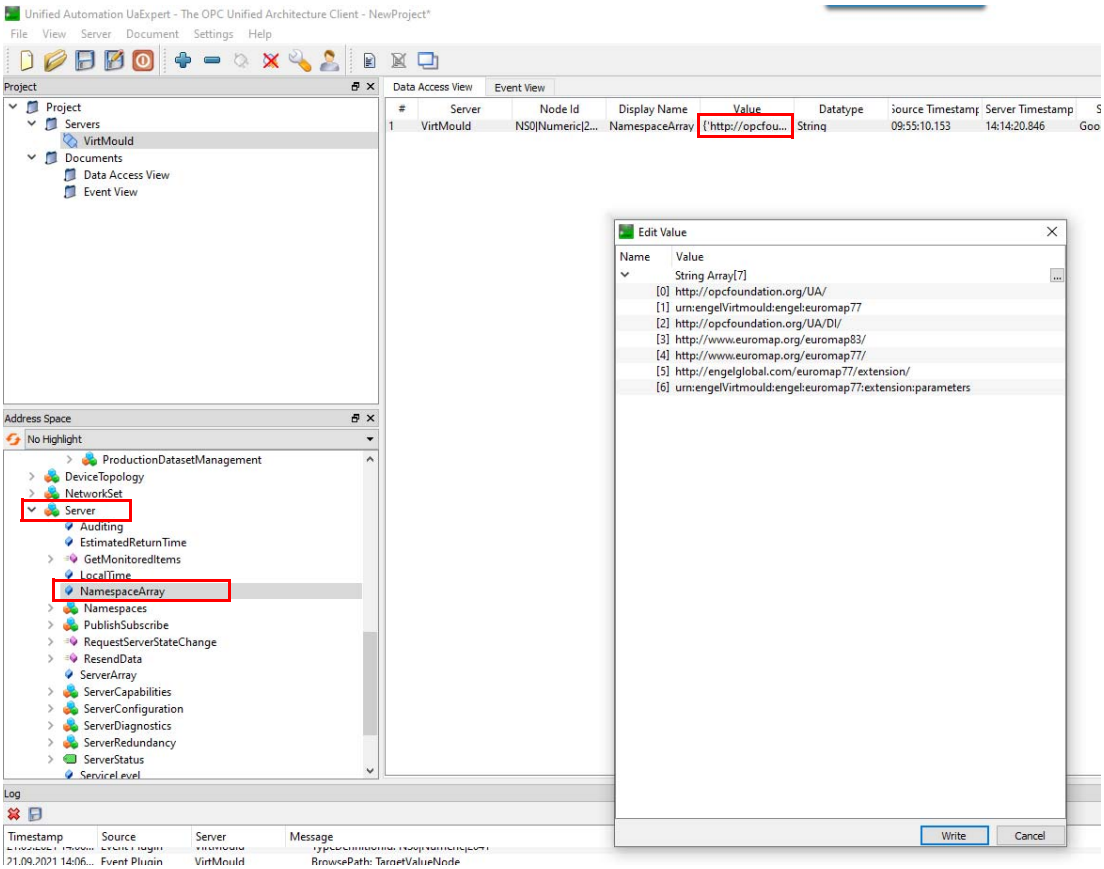
Syntax:

<root>/<NamespaceIndex>:<BrowseName>/...

In each intermediate level, the NamespaceIndex and the BrowseName can be determined. These together (according to the syntax) result in the BrowsePath.



There is also the theoretical possibility that the BrowsePath can change as well. The cause is the NamespaceArray (figure below), which has the potential to change over time (e.g. when the machine is restarted). However, this is more of a theoretical option, which should probably not play a role in practice.



Index

ENGEL Austria GmbH | 4311 Schwertberg | tel: +43 (0)50 620 0 fax: +43 (0)50 620 3609
e-mail: service@engel.at | www.engelglobal.com