

<b>EUROMAP 84.2</b>	<b>OPC UA interfaces for plastics and rubber machinery – Extrusion – Part 2: Extrusion Line</b>
---------------------	---

Release 1.00, 2020-06-01

<p><b>EUROMAP 84.2 (Release 1.00) is identical with OPC 40084-2 (Release 1.00) and VDMA 40084-2:2020-06</b></p>
---

## Contents

	Page
Foreword.....	6
1 Scope .....	6
2 Normative references .....	7
3 Terms, definitions and conventions .....	7
3.1 Overview .....	7
3.2 Conventions used in this document.....	8
3.3 Abbreviations .....	8
4 General information to OPC UA interfaces for plastics and rubber machinery and OPC UA .....	8
5 Use cases .....	8
6 Basic security requirements.....	8
6.1 Application Security .....	8
6.2 User security/Access control .....	8
7 ExtrusionLine_InterfaceType .....	9
7.1 ExtrusionLine_InterfaceType Definition.....	9
7.2 Lineld .....	10
7.3 LineStatus.....	10
7.4 MachineInformation and DeciveClass .....	10
7.5 MachineConfiguration .....	10
7.6 MachineMESConfiguration .....	10
7.7 MESMessage, SetMESMessage, ClearMESMessage .....	10
7.8 Users .....	11
7.9 ConfigurationParameters.....	11
7.10 MaterialList .....	11
8 ProductionParameters .....	11
9 Management of Jobs with job groups and jobs .....	13
9.1 JobGroups.....	14
9.2 JobGroupType .....	17
9.3 JobType .....	22
10 ProductionDatasetManagement.....	25
11 Profiles and Conformance Units .....	25
12 Namespaces .....	25
12.1 Namespace Metadata .....	25
12.2 Handling of OPC UA Namespaces .....	26
Annex A (normative) OPC 40084-2 Namespace and mappings.....	27
Annex B (informative) Examples for job descriptions .....	28

## Figures

Figure 1 – ExtrusionLine_MES_InterfaceType Overview .....	9
Figure 2 – Job Management Overview .....	13

## Tables

Table 1 – ExtrusionLine_InterfaceType Definition .....	10
Table 2 – Standardized configuration parameters for OPC 40084-2 .....	11
Table 3 – ProductionParametersType Definition .....	11
Table 4 – JobGroupsType Definition .....	14
Table 5 – AddJobGroup Method Arguments .....	15
Table 6 – AddJobGroup Method AddressSpace Definition .....	15
Table 7 – RemoveJobGroupById Method Arguments .....	15
Table 8 – RemoveJobGroupById Method AddressSpace Definition .....	15
Table 9 – StartJobGroupById Method Arguments .....	15
Table 10 – StartJobGroup Method AddressSpace Definition .....	16
Table 11 – InterruptJobGroupById Method Arguments .....	16
Table 12 – InterruptJobGroupById Method AddressSpace Definition .....	16
Table 13 – FinishJobGroupById Method Arguments .....	16
Table 14 – FinishJobGroupById Method AddressSpace Definition .....	16
Table 15 – JobGroupType Definition .....	17
Table 16 – MaterialMappingType Definition .....	18
Table 17 – AddJob Method Arguments .....	19
Table 18 – AddJob Method AddressSpace Definition .....	19
Table 19 – RemoveJobById Method Arguments .....	19
Table 20 – RemoveJobById Method AddressSpace Definition .....	19
Table 21 – StartJobById Method Arguments .....	20
Table 22 – StartJob Method AddressSpace Definition .....	20
Table 23 – InterruptJobById Method Arguments .....	20
Table 24 – InterruptJobById Method AddressSpace Definition .....	20
Table 25 – FinishJobById Method Arguments .....	20
Table 26 – FinishJobById Method AddressSpace Definition .....	21
Table 27 – JobGroupStatusChangedEventType Definition .....	21
Table 28 – JobType Definition .....	22
Table 29 – JobStatusChangedEventType Definition .....	23
Table 30 – UnitFinishedEventType Definition .....	24
Table 31 – LotFinishedEventType Definition .....	24
Table 32 – Profile URIs for OPC 40084-2 .....	25
Table 33 – OPC 40084-2 Basic Server Facet Definition .....	25
Table 34 – OPC 40084-2 Jobs Server Facet Definition .....	25
Table 35 – OPC 40084-2 ProductionDatasetManagement Server Facet Definition .....	25
Table 36 – NamespaceMetadata Object for this Specification .....	26
Table 37 – Namespaces used in an OPC 40084-2 Server .....	26
Table 38 – Namespaces used in this specification .....	26

## **OPC Foundation / EUROMAP**

---

### **AGREEMENT OF USE**

#### **COPYRIGHT RESTRICTIONS**

- This document is provided "as is" by the OPC Foundation and EUROMAP.
- Right of use for this specification is restricted to this specification and does not grant rights of use for referred documents.
- Right of use for this specification will be granted without cost.
- This document may be distributed through computer systems, printed or copied as long as the content remains unchanged and the document is not modified.
- OPC Foundation and EUROMAP do not guarantee usability for any purpose and shall not be made liable for any case using the content of this document.
- The user of the document agrees to indemnify OPC Foundation and EUROMAP and their officers, directors and agents harmless from all demands, claims, actions, losses, damages (including damages from personal injuries), costs and expenses (including attorneys' fees) which are in any way related to activities associated with its use of content from this specification.
- The document shall not be used in conjunction with company advertising, shall not be sold or licensed to any party.
- The intellectual property and copyright is solely owned by the OPC Foundation and EUROMAP.

#### **PATENTS**

The attention of adopters is directed to the possibility that compliance with or adoption of OPC or EUROMAP specifications may require use of an invention covered by patent rights. OPC Foundation or EUROMAP shall not be responsible for identifying patents for which a license may be required by any OPC or EUROMAP specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OPC or EUROMAP specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

#### **WARRANTY AND LIABILITY DISCLAIMERS**

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OPC FOUNDATION NOR EUROMAP MAKES NO WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OPC FOUNDATION NOR EUROMAP BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you.

#### **RESTRICTED RIGHTS LEGEND**

This Specification is provided with Restricted Rights. Use, duplication or disclosure by the U.S. government is subject to restrictions as set forth in (a) this Agreement pursuant to DFARs 227.7202-3(a); (b) subparagraph (c)(1)(i) of the Rights in Technical Data and Computer Software clause at DFARs 252.227-7013; or (c) the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 subdivision (c)(1) and (2), as applicable. Contractor / manufacturer are the OPC Foundation, 16101 N. 82nd Street, Suite 3B, Scottsdale, AZ, 85260-1830

#### **COMPLIANCE**

The combination of EUROMAP and OPC Foundation shall at all times be the sole entities that may authorize developers, suppliers and sellers of hardware and software to use certification marks, trademarks or other special designations to indicate compliance with these materials as specified within this document. Products developed using this specification may claim compliance or conformance with this specification if and only if the software satisfactorily meets the certification requirements set by EUROMAP or the OPC Foundation. Products that do not meet these requirements may claim only that the product was based on this specification and must not claim compliance or conformance with this specification.

## TRADEMARKS

Most computer and software brand names have trademarks or registered trademarks. The individual trademarks have not been listed here.

## GENERAL PROVISIONS

Should any provision of this Agreement be held to be void, invalid, unenforceable or illegal by a court, the validity and enforceability of the other provisions shall not be affected thereby.

This Agreement shall be governed by and construed under the laws of Germany.

This Agreement embodies the entire understanding between the parties with respect to, and supersedes any prior understanding or agreement (oral or written) relating to, this specification.

## Foreword

This specification was created by a joint working group of the OPC Foundation and EUROMAP. It is adopted identically as VDMA Specification.

## EUROMAP

EUROMAP is the European umbrella association of the plastics and rubber machinery industry which accounts for annual sales of around 13.5 billion euro and a 40 per cent share of worldwide production. Almost 75 per cent of its European output is shipped to worldwide destinations. With global exports of 10.0 billion euro, EUROMAP's around 1,000 machinery manufacturers are market leaders with nearly half of all machines sold being supplied by EUROMAP members.

EUROMAP provides technical recommendations for plastics and rubber machines. In addition to standards for machine descriptions, dimensions and energy measurement, interfaces between machines feature prominently. The provision of manufacturer independent interfaces ensures high levels of machine compatibility.

## OPC Foundation

OPC is the interoperability standard for the secure and reliable exchange of data and information in the industrial automation space and in other industries. It is platform independent and ensures the seamless flow of information among devices from multiple vendors. The OPC Foundation is responsible for the development and maintenance of this standard.

OPC UA is a platform independent service-oriented architecture that integrates all the functionality of the individual OPC Classic specifications into one extensible framework. This multi-layered approach accomplishes the original design specification goals of:

- Platform independence: from an embedded microcontroller to cloud-based infrastructure
- Secure: encryption, authentication, authorization and auditing
- Extensible: ability to add new features including transports without affecting existing applications
- Comprehensive information modelling capabilities: for defining any model from simple to complex

## 1 Scope

OPC 40084-2 describes the interface between extrusion lines and manufacturing execution systems (MES) for data exchange. MES are used for collecting the information generated by extrusion lines at a central point for easier quality assurance and job and dataset management. The target of OPC 40084-2 is to provide a unique interface for extrusion lines and MES from different manufacturers to ensure compatibility.

The following functionalities are covered:

- General information about the extrusion lines (manufacturer, model, serial number...), current configuration and status of the extruder line.
- Job management: Information on the jobs running on the machine and the parameters of the production and methods to send jobs from the MES to the extrusion line and to release the production.
- Recipe management: Extrusion lines store their configurations in so-called recipes. These include information on nominal process parameters (temperatures, dosing volumes ...). OPC 40084-2 allows transferring datasets between extrusion lines and MES for building a central repository of recipes.

Following functions are not included:

- Safety related signals like emergency stop
- Direct control of machine movements by the MES

This part of OPC 40084 deals with extrusion lines and functionalities related to the whole system. Information models for its components (extruder, dies, winders...) are defined in the other parts.

## 2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies

OPC 10000-1, *OPC Unified Architecture - Part 1: Overview and Concepts*

<http://www.opcfoundation.org/UA/Part1/>

OPC 10000-2, *OPC Unified Architecture - Part 2: Security Model*

<http://www.opcfoundation.org/UA/Part2/>

OPC 10000-3, *OPC Unified Architecture - Part 3: Address Space Model*

<http://www.opcfoundation.org/UA/Part3/>

OPC 10000-4, *OPC Unified Architecture - Part 4: Services*

<http://www.opcfoundation.org/UA/Part4/>

OPC 10000-5, *OPC Unified Architecture - Part 5: Information Model*

<http://www.opcfoundation.org/UA/Part5/>

OPC 10000-6, *OPC Unified Architecture - Part 6: Mappings*

<http://www.opcfoundation.org/UA/Part6/>

OPC 10000-7, *OPC Unified Architecture - Part 7: Profiles*

<http://www.opcfoundation.org/UA/Part7/>

OPC 10000-8, *OPC Unified Architecture - Part 8: Data Access*

<http://www.opcfoundation.org/UA/Part8/>

OPC 10000-9, *OPC Unified Architecture - Part 9: Alarms and Conditions*

<http://www.opcfoundation.org/UA/Part9/>

OPC 10000-11, *OPC Unified Architecture - Part 11: Historical Access*

<http://www.opcfoundation.org/UA/Part11/>

OPC 10001-1, *OPC Unified Architecture V1.04 - Amendment 1: AnalogItem Types*

<http://www.opcfoundation.org/UA/Amendment1/>

OPC 10001-3, *OPC Unified Architecture V1.04 - Amendment 3: Method Metadata*

<http://www.opcfoundation.org/UA/Amendment3/>

OPC 10000-100, *OPC Unified Architecture - Part 100: Devices*

<http://www.opcfoundation.org/UA/Part100/>

OPC 40083: OPC UA interfaces for plastics and rubber machinery – General Type definitions (version 1.02)

<http://www.opcfoundation.org/UA/PlasticsRubber/GeneralTypes>

OPC 40084-1: OPC UA interfaces for plastics and rubber machinery – Extrusion – Part 1: General Type Definitions (version 1.0)

<http://www.opcfoundation.org/UA/PlasticsRubber/Extrusion/GeneralTypes/>

## 3 Terms, definitions and conventions

### 3.1 Overview

It is assumed that basic concepts of OPC UA information modelling are understood in this specification. This specification will use these concepts to describe the OPC 40084-2 Information Model. For the purposes of this document, the terms and definitions given in the documents referenced in Clause 2 apply.

Note that OPC UA terms and terms defined in this specification are *italicized* in the specification.

### **3.2 Conventions used in this document**

The conventions described in OPC 40083 apply.

### **3.3 Abbreviations**

MES Manufacturing Execution System

## **4 General information to OPC UA interfaces for plastics and rubber machinery and OPC UA**

For general information on OPC UA interfaces for plastics and rubber machinery and OPC UA see OPC 40083.

## **5 Use cases**

The following functionalities are covered:

- General information about the extrusion lines (manufacturer, model, serial number...), current configuration and status of the extruder line.
- Job management: Information on the jobs running on the machine and the parameters of the production and methods to send jobs from the MES to the extrusion line and to release the production.
- Recipe management: Extrusion lines store their configurations in so-called recipes. These include information on nominal process parameters (temperatures, dosing volumes ...). OPC 40084-2 allows transferring datasets between extrusion lines and MES for building a central repository of recipes.

## **6 Basic security requirements**

### **6.1 Application Security**

For the communication between the extrusion line and MES the OPC UA application authentication via X509 certificates shall be used. OPC UA provides functionalities for using self-signed certificates that have to be manually added to a "trust list" as well as for certificates issued by a certificate authority (CA).

The minimum requirements of the protocol level for a OPC 40084-2 compliant connection are:

- Use of (self-signed) certificates for OPC UA application authentication
- Security Policy: Basic256
- Message Security Mode: sign

NOTE: It is not fixed by this specification if the certificate includes a fixed IP address and/or the host name. However, if the certificate includes a host name, a DNS server is expected to resolve the host name. An OPC UA GDS (Global Discovery Server) can be used to manage the connections and certificates.

### **6.2 User security/Access control**

#### **6.2.1 On extrusion line**

On the extrusion line authentication via user name and password is commonly used.

#### **6.2.2 On MES**

For the users and roles of the connection the following applies:

- User names can be manufacturer dependent.
- Standard roles are
  - "OPC40084": read and write access for selected parameters
  - "OPC40084\_read\_only": no writing permissions
- Manufacturers can add additional roles. They may not start with "OPC40084". For these roles, more parameters can be writeable than for the OPC40084 role.



- The standard user “OPC40084” has the role “OPC40084” (and no other additional role), “OPC4004\_read\_only” has the roll “OPC40084\_read\_only” (and no other additional role); the passwords for the standard users are defined by the manufacturers (they may be empty).

NOTE: OPC UA also allow an anonymous-token (e.g. for testing)

## 7 ExtrusionLine\_InterfaceType

### 7.1 ExtrusionLine\_InterfaceType Definition

This OPC UA *ObjectType* is used for the root *Object* representing an extrusion line as overall system. It is formally defined in Table 1.

The instance(s) of *ExtrusionLine\_InterfaceType* shall be located under the *DeviceSet Object* of the Server (see OPC UA Part 100).

NOTE: In most cases the OPC UA server will be implemented in the line control of the extrusion line so only one instance of *ExtrusionLine\_InterfaceType* will be created. But it is also possible that one OPC UA server is connected to several extrusion lines as one interface to the MES. In this case several instances of *ExtrusionLine\_InterfaceType* will be created.

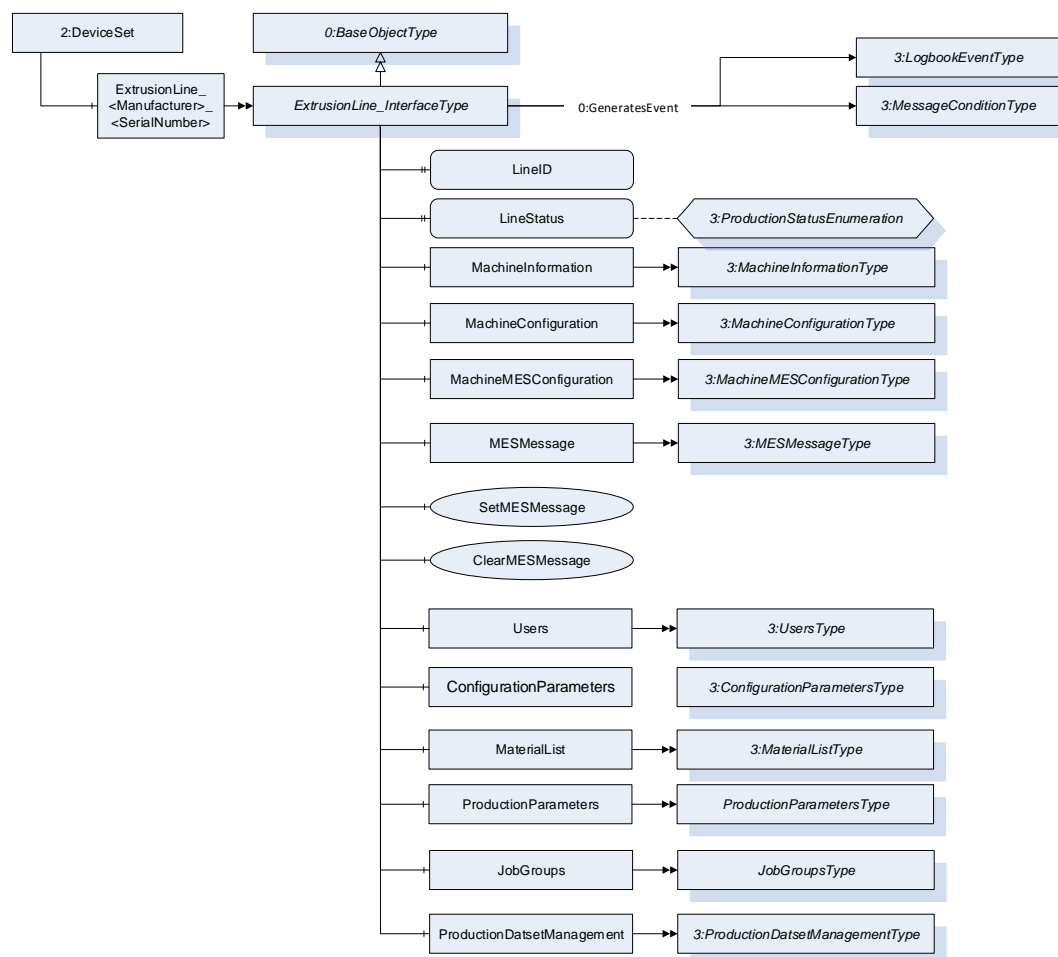


Figure 1 – ExtrusionLine\_MES\_InterfaceType Overview

**Table 1 – ExtrusionLine\_InterfaceType Definition**

Attribute	Value				
BrowseName	ExtrusionLine_InterfaceType				
IsAbstract	False				
References	Node Class	BrowseName	Data Type	TypeDefinition	Other
Subtype of 0:BaseObjectType defined in OPC UA Part 5					
0:HasProperty	Variable	LineId	String	0:PropertyType	M, RW
0:HasProperty	Variable	LineStatus	ProductionStatus Enumeration	0:PropertyType	M
0:HasComponent	Object	MachineInformation		3:MachineInformationType	M
0:HasComponent	Object	MachineConfiguration		3:MachineConfigurationType	M
0:HasComponent	Object	MachineMESConfiguration		3:MachineMESConfigurationType	M
0:HasComponent	Object	MESMessage		3:MESMessageType	O
0:HasComponent	Method	SetMESMessage			O
0:HasComponent	Method	ClearMESMessage			O
0:HasComponent	Object	Users		3:UsersType	M
0:HasProperty	Variable	ConfigurationParameters	ConfigurationParameterType[]	0:PropertyType	M, RO
0:HasComponent	Object	MaterialList		3:MaterialListType	M
0:HasComponent	Object	ProductionParameters		ProductionParametersType	M
0:HasComponent	Object	JobGroups		JobGroupsType	O
0:HasComponent	Object	ProductionDataset Management		3:ProductionDataset ManagementType	O
0:GeneratesEvent	ObjectType	3:LogbookEventType	Defined in OPC 40083		
0:GeneratesEvent	ObjectType	3:MessageConditionType	Defined in OPC 40083		

The *BrowseName* of the object instance shall be "ExtrusionLine\_<Manufacturer>\_<SerialNumber>"

Example: "ExtrusionLine\_battenfeld\_4".

## 7.2 LineId

This *Property* is the identifier of the extrusion line. Components (extruders, dies, winders, ...) use the *LineId* to indicates to which extrusion line they belong to (e.g. "blown film line 2").

## 7.3 LineStatus

This *Property* represents the current status of the extrusion line. Possible values are defined in *ProductionStatusEnumeration* of OPC 40083.

NOTE: For extrusion lines the status START\_UP\_2 shall also apply during material/product change, cleaning, shutting down. DRY\_RUN\_5 is not applicable for extrusion lines.

## 7.4 MachineInformation and DeviceClass

The *MachineInformationType* is defined in OPC 40083 and provides basic information on a machine/device.

The *DeviceClass Property* in the *MachineInformation* Object shall have the value "ExtrusionLine".

## 7.5 MachineConfiguration

The *MachineConfigurationType* is defined in OPC 40083 and provides information on the current configuration of a machine/device.

## 7.6 MachineMESConfiguration

The *MachineMESConfigurationType* is defined in OPC 40083. Here a list of standstill reasons for later selection by the operator can be defined.

## 7.7 MESMessage, SetMESMessage, ClearMESMessage

The *Object MESMessage* and the methods *SetMESMessage*, *ClearMESMessage* are defined in OPC 40083 and used for sending text message from the MES to be shown on the machine.

NOTE: The *Object MESMessage* and the methods *SetMESMessage*, *ClearMESMessage* are taken out of the *MachineMESStatusType*, because not all functionalities defined there are relevant for extrusion lines.

## 7.8 Users

The *UsersType* is defined in OPC 40083 and provides information on the current users on the machine/device.

## 7.9 ConfigurationParameters

The *ConfigurationParameters Property* contains a list of parameters, that can be used within the job description to configure the production of the extrusion line. The *ConfigurationParametersType* is defined in OPC 40083.

The following configuration parameters are standardized for OPC 40084-2:

**Table 2 – Standardized configuration parameters for OPC 40084-2**

Id	Name (for English version of LocalizedText)	DataType
1	Length	Double
2	Width	Double
3	Thickness	Double
4	Diameter	Double
5	Wall thickness	Double
6	Pieces	Double
7	Weight	Double

Manufacturer dependent configuration parameters shall have an *Id* ≥ 100.

## 7.10 MaterialList

This *Object* is used to provide a list of materials which are intended to be used on the extrusion line.

The *MaterialListType* is defined in OPC 40083.

## 8 ProductionParameters

The *ProductionParametersType* contains several production parameters which are related to the complete extrusion line. Process parameters of the components (e.g. core extruder, winder) are located inside the models for these components.

**Table 3 – ProductionParametersType Definition**

Attribute	Value				
BrowseName	ProductionParametersType				
IsAbstract	False				
References	Node Class	BrowseName	DataType	TypeDefinition	Other
Subtype of 0:BaseObjectType defined in OPC UA Part 5					
0:HasComponent	Variable	GoodProduct	0:Boolean	0:BaseDataVariableType	M, RO
0:HasComponent	Variable	Throughput	0:Double	0:AnalogUnitType	O, RO
0:HasComponent	Variable	ProductWeight	0:Double	0:AnalogUnitType	O, RO
0:HasComponent	Variable	LineSpeed	0:Double	0:AnalogUnitType	O, RO
0:HasComponent	Object	ElectricalEnergy		3:EnergyType	O, RO
0:HasComponent	Object	FluidEnergy		3:EnergyType	O, RO
0:HasComponent	Object	PressureAir		3:EnergyType	O, RO

### 8.1.1 GoodProduct

Indication if the produced product has a good quality. Can be set by the machine or the operator. When it changes from true to false the line control shall create an alarm.

This *Property* gives the general information on the complete extrusion line. The *JobType* (see 0) contains also a *Property GoodProduct* which is specific for the running job. When two or more jobs are running in parallel, the product quality of each strand can be different. That means, if in one of the jobs *GoodProduct* becomes False, then the *GoodProduct* in *ProductionParameters* shall also be false.

### **8.1.2 Throughput**

Throughput of the extrusion line in mass per time (e.g. kg/h).

### **8.1.3 ProductWeight**

Weight of the produced product in mass per length or volume (e.g. g/m, kg/m<sup>3</sup>).

### **8.1.4 LineSpeed**

Production speed of the line in length per time (e.g. m/s).

### **8.1.5 ElectricalEnergy**

Electrical energy of the extrusion line. The *EnergyType* is defined in OPC 40083.

### **8.1.6 FluidEnergy**

Energy for the cooling of the whole extrusion line with fluid. The *EnergyType* is defined in OPC 40083.

### **8.1.7 PressureAir**

Consumption of pressure air of the whole extrusion line (volume at standard conditions). The *EnergyType* is defined in OPC 40083.

## 9 Management of Jobs with job groups and jobs

The definition which product (family) is produced is done by a job group. This defines which equipment, production dataset and materials are used and which configuration parameters (e.g. width, length of foil) are offered. The concrete products (roll of film with defined width and length) are models as jobs inside this group.

NOTE: Examples for different combinations of jobs are presented in the Annex.

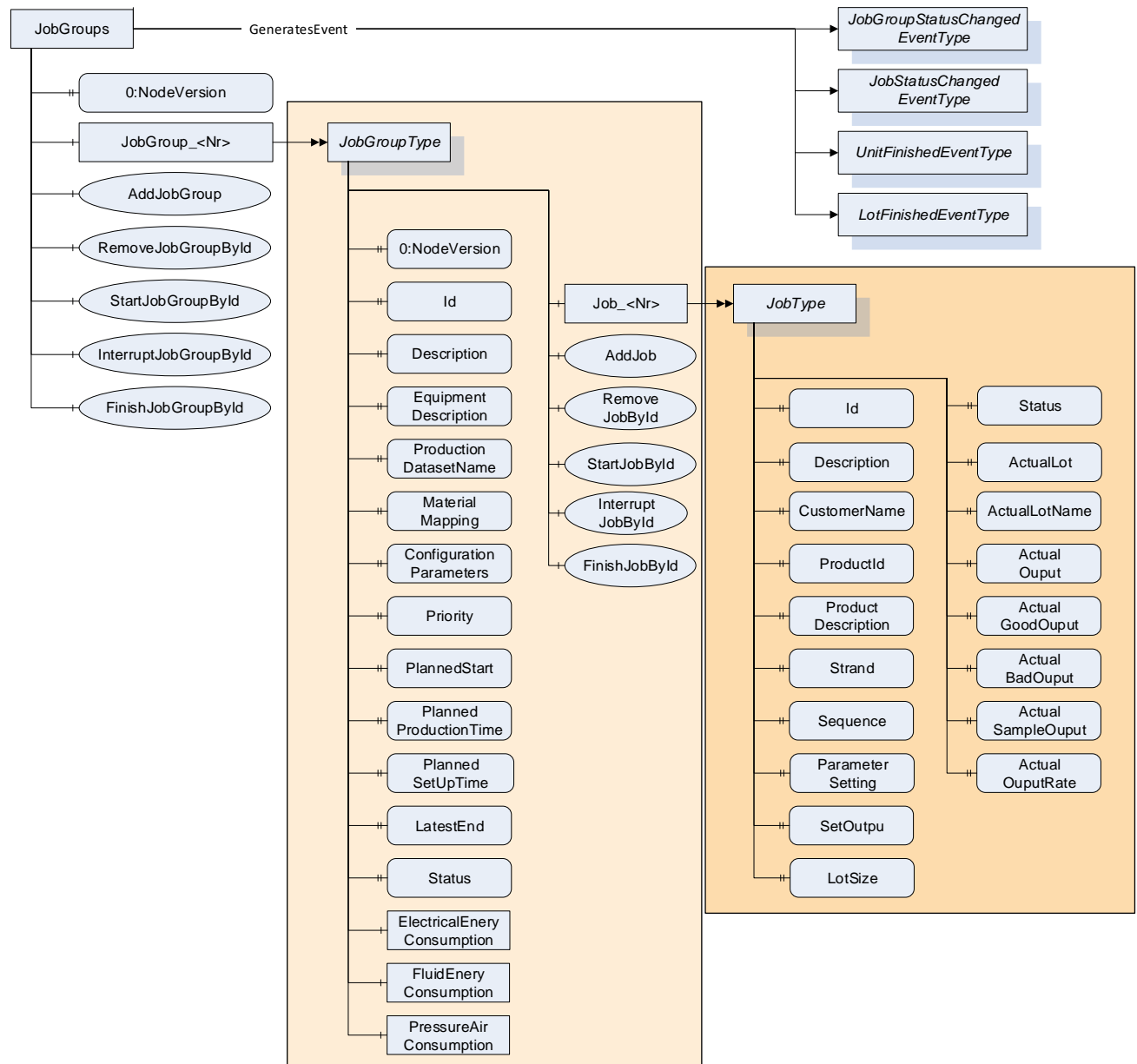


Figure 2 – Job Management Overview

## 9.1 JobGroups

This is a container for the job groups.

**Table 4 – JobGroupsType Definition**

Attribute	Value				
BrowseName	JobGroupsType				
IsAbstract	False				
References	Node Class	BrowseName	DataType	TypeDefinition	Other
Subtype of 0:BaseObjectType defined in OPC UA Part 5					
0:HasProperty	Variable	0:NodeVersion	0:String	0:PropertyType	M, RO
0:HasComponent	Object	JobGroup_<Nr>		JobGroupType	OP
0:HasComponent	Method	AddJobGroup			O
0:HasComponent	Method	RemoveJobGroupById			O
0:HasComponent	Method	StartJobGroupById			O
0:HasComponent	Method	InterruptJobGroupById			O
0:HasComponent	Method	FinishJobGroupById			O
0:GeneratesEvent	ObjectType	JobGroupStatusChangedEventType	Defined in 9.2.23		
0:GeneratesEvent	ObjectType	JobStatusChangedEventType	Defined in 9.3.17		
0:GeneratesEvent	ObjectType	UnitFinishedEventType	Defined in 9.3.18		
0:GeneratesEvent	ObjectType	LotFinishedEventType	Defined in 9.3.19		
0:GeneratesEvent	ObjectType	0:GeneralModelChangeEvent			

### 9.1.1 NodeVersion

The *NodeVersion Property* as defined in OPC UA Part 3 is used to inform the client about model changes. Here it informs about added or removed instances of *JobGroupType*.

### 9.1.2 JobGroup\_<Nr>

This is a placeholder for the job groups (see 0). When instances are created the *BrowseNames* shall be "JobGroup\_<Nr>" where <Nr> is a three-digit number with leading zeros, starting with "001".

### 9.1.3 AddJobGroup

This method adds a new job group with the needed Properties.

#### Signature

```
AddJobGroup (
    [in]    String      Id
    [in]    String      Description
    [in]    String      EquipmentDescription
    [in]    String      ProductionDatasetName
    [in]    MaterialMappingType[] MaterialMapping
    [in]    UInt32       Priority
    [in]    UtcTime      PlannedStart
    [in]    Duration     PlannedProductionTime
    [in]    Duration     PlannedSetUpTime
    [in]    UtcTime      LatestEnd
    [out]   NodeId       JobGroupId);
```

**Table 5 – AddJobGroup Method Arguments**

Argument	Description
Id	See 0, where the Properties of a job group are defined.
Description	See 0, where the Properties of a job group are defined.
EquipmentDescription	See 0, where the Properties of a job group are defined.
ProductionDatasetName	See 0, where the Properties of a job group are defined.
MaterialMapping	See 0, where the Properties of a job group are defined.
Priority	See 0, where the Properties of a job group are defined.
PlannedStart	See 0, where the Properties of a job group are defined.
PlannedProductionTime	See 0, where the Properties of a job group are defined.
PlannedSetUpTime	See 0, where the Properties of a job group are defined.
LatestEnd	See 0, where the Properties of a job group are defined.
JobGroupNodeId	The method returns the <i>NodeId</i> of the created instance for the job group

**Table 6 – AddJobGroup Method AddressSpace Definition**

Attribute	Value				
BrowseName	AddJobGroup				
References	Node Class	BrowseName	DataType	TypeDefinition	Modelling Rule
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory
HasProperty	Variable	OutputArguments	Argument[]	PropertyType	Mandatory

#### 9.1.4 RemoveJobGroupById

This method removes a job group instance.

##### Signature

```
RemoveJobGroupById (
    [in]      String      Id);
```

**Table 7 – RemoveJobGroupById Method Arguments**

Argument	Description
Id	Id of the job group that shall be removed

**Table 8 – RemoveJobGroupById Method AddressSpace Definition**

Attribute	Value				
BrowseName	RemoveJobGroupById				
References	Node Class	BrowseName	DataType	TypeDefinition	Modelling Rule
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory

#### 9.1.5 StartJobGroupById

With this method the client requests to start the production of a specific job group.

##### Signature

```
StartJobGroupById (
    [in]      String      Id);
```

**Table 9 – StartJobGroupById Method Arguments**

Argument	Description
Id	Id of the job group that shall be started

**Table 10 – StartJobGroup Method AddressSpace Definition**

Attribute	Value				
BrowseName	StartJobGroupById				
References	Node Class	BrowseName	Data Type	TypeDefinition	Modelling Rule
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory

### 9.1.6 InterruptJobGroupById

With this method the client requests to interrupt the production of a specific job group.

#### Signature

```
InterruptJobGroupById (
    [in] String Id);
```

**Table 11 – InterruptJobGroupById Method Arguments**

Argument	Description
Id	Id of the job group that shall be interrupted

**Table 12 – InterruptJobGroupById Method AddressSpace Definition**

Attribute	Value				
BrowseName	InterruptJobGroupById				
References	Node Class	BrowseName	Data Type	TypeDefinition	Modelling Rule
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory

To continue the production of the job group, *StartJobGroupById* is called.

### 9.1.7 FinishJobGroupById

With this method the client requests to finish the production of a specific job group.

#### Signature

```
FinishJobGroupById (
    [in] String Id);
```

**Table 13 – FinishJobGroupById Method Arguments**

Argument	Description
Id	Id of the job group that shall be finished

**Table 14 – FinishJobGroupById Method AddressSpace Definition**

Attribute	Value				
BrowseName	FinishJobGroupById				
References	Node Class	BrowseName	Data Type	TypeDefinition	Modelling Rule
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory

### 9.1.8 Events

The *JobGroups Object* fires all *Events* related to the included job groups and jobs. With this, a client needs to subscribe to these events only at this node. The different events are described in 0 and 0.



## 9.2 JobGroupType

This *ObjectType* represent a job group.

**Table 15 – JobGroupType Definition**

Attribute	Value				
BrowseName	JobGroupType				
IsAbstract	False				
References	Node Class	BrowseName	DataType	TypeDefinition	Other
Subtype of 0:BaseObjectType defined in OPC UA Part 5					
0:HasProperty	Variable	0:NodeVersion	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	Id	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	Description	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	EquipmentDescription	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	ProductionDatasetName	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	MaterialMapping	MaterialMappingType[]	0:PropertyType	M, RO
0:HasProperty	Variable	ConfigurationParameters	ConfigurationParameterType[]	0:PropertyType	M, RO
0:HasProperty	Variable	Priority	0:UInt32	0:PropertyType	M, RO
0:HasProperty	Variable	PlannedStart	0:UtcTime	0:PropertyType	M, RO
0:HasProperty	Variable	PlannedProductionTime	0:Duration	0:PropertyType	M, RO
0:HasProperty	Variable	PlannedSetUpTime	0:Duration	0:PropertyType	M, RO
0:HasProperty	Variable	LatestEnd	0:UtcTime	0:PropertyType	M, RO
0:HasProperty	Variable	Status	3:JobStatusEnumeration	0:PropertyType	M, RO
0:HasComponent	Variable	ElectricalEnergyConsumption	0:Double	0:AnalogUnitType	O, RO
0:HasComponent	Variable	FluidEnergyConsumption	0:Double	0:AnalogUnitType	O, RO
0:HasComponent	Variable	PressureAirConsumption	0:Double	0:AnalogUnitType	O, RO
0:HasComponent	Object	Job_<Nr>		JobType	OP
0:HasComponent	Method	AddJob			O
0:HasComponent	Method	RemoveJobById			O
0:HasComponent	Method	StartJobById			O
0:HasComponent	Method	InterruptJobById			O
0:HasComponent	Method	FinishJobById			O
0:GeneratesEvent	ObjectType	0:GeneralModelChange EventType			

### 9.2.1 NodeVersion

The *NodeVersion Property* as defined in OPC UA Part 3 is used to inform the client about model changes. Here it informs about added or removed instances of *JobType* in the group.

### 9.2.2 Id

Id of the job group.

### 9.2.3 Description

Description of the job group.

### 9.2.4 EquipmentDescription

Description of the equipment needed for job. This is only for information (e.g. for operator)

NOTE: It makes sense only to mention the interchangeable equipment of the extrusion line here (e.g. die, screw).

### 9.2.5 ProductionDatasetName

Name of the production dataset which is needed for the job.

### 9.2.6 MaterialMapping

Information, which material is needed for the production. The *MaterialMappingType* is defined in Table 16.

**Table 16 – MaterialMappingType Definition**

Name	Type	Description
MaterialMappingType	structure	
MaterialId	String	Id of the material as specified in the MaterialList (see 7.10)
MaterialLot	String	Lot of the material (empty string allowed if not relevant)
HopperId	String	Id of the hopper in which the material shall be put (empty string allowed if not supported)

### 9.2.7 Priority

Priority of the job group: 1 = highest priority, higher values represent lower priorities.

### 9.2.8 PlannedStart

Planned start of the job (in UTC time). If not relevant the value shall be 0.

### 9.2.9 PlannedProductionTime

Planned time for the production of the job group. If not relevant the value shall be 0.

### 9.2.10 PlannedSetUpTime

Planned total time needed for set up and tear down of the extrusion line for the job group. If not relevant the value shall be 0.

### 9.2.11 LatestEnd

Latest end of the job (in UTC time). If not relevant the value shall be 0.

### 9.2.12 ConfigurationParameters

Array of offered configuration parameters. The *ConfigurationParameterType* is defined in OPC 40083. Which ConfigurationParameters are offered to the jobs in the group depends on the machine / implementation / production dataset.

### 9.2.13 Status

Status of the job group. The *JobStatusEnumeration* is defined in OPC 40083.

### 9.2.14 ElectricalEnergyConsumption

Cumulated electrical energy consumption of the extrusion line for the job group in kWh.

### 9.2.15 FluidEnergyConsumption

Cumulated energy for the cooling of the whole extrusion line with fluid for the job group in kWh.

### 9.2.16 PressureAirConsumption

Cumulated consumption of pressure air of the whole extrusion line (volume at standard conditions) for the job group in kWh.

### 9.2.17 Job\_<Nr>

This is a placeholder for the jobs inside the group (see 0). When instances are created the *BrowseNames* shall be "Job\_<Nr>" where <Nr> is a three-digit number with leading zeros, starting with "001".

### 9.2.18 AddJob

This method adds a new job to the group with the needed Properties.

**Signature**

```

AddJob (
    [in]    String          Id
    [in]    String          Description
    [in]    String          CustomerName
    [in]    String          ProductName
    [in]    String          ProductDescription
    [in]    UInt32          Strand
    [in]    UInt32          Sequence
    [in]    ParameterSettingType[] ParameterSetting
    [in]    Double          SetOutput
    [in]    Double          LotSize
    [out]   NodeId          JobNodeId);

```

**Table 17 – AddJob Method Arguments**

Argument	Description
Id	See 0, where the Properties of a job are defined.
Description	See 0, where the Properties of a job are defined.
CustomerName	See 0, where the Properties of a job are defined.
ProductName	See 0, where the Properties of a job are defined.
ProductDescription	See 0, where the Properties of a job are defined.
Strand	See 0, where the Properties of a job are defined.
Sequence	See 0, where the Properties of a job are defined.
ParameterSetting	See 0, where the Properties of a job are defined.
SetOutput	See 0, where the Properties of a job are defined.
LotSize	See 0, where the Properties of a job are defined.
JobNodeId	The method returns the <i>NodeId</i> of the created instance for the job group

**Table 18 – AddJob Method AddressSpace Definition**

Attribute	Value				
BrowseName	AddJob				
References	Node Class	BrowseName	DataType	TypeDefinition	Modelling Rule
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory
HasProperty	Variable	OutputArguments	Argument[]	PropertyType	Mandatory

**9.2.19 RemoveJobById**

This method removes a job from a group.

**Signature**

```

RemoveJobById (
    [in]    String          Id);

```

**Table 19 – RemoveJobById Method Arguments**

Argument	Description
Id	Id of the job that shall be removed

**Table 20 – RemoveJobById Method AddressSpace Definition**

Attribute	Value				
BrowseName	RemoveJobById				
References	Node Class	BrowseName	DataType	TypeDefinition	Modelling Rule
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory

### 9.2.20 StartJobById

With this method the client requests to start the production of a specific job in the group.

#### Signature

```
StartJobById (
    [in]      String      Id);
```

**Table 21 – StartJobById Method Arguments**

Argument	Description
Id	Id of the job that shall be started

**Table 22 – StartJob Method AddressSpace Definition**

Attribute	Value				
BrowseName	StartJobById				
References	Node Class	BrowseName	DataType	TypeDefinition	Modelling Rule
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory

### 9.2.21 InterruptJobById

With this method the client requests to interrupt the production of a specific job.

#### Signature

```
InterruptJobById (
    [in]      String      Id);
```

**Table 23 – InterruptJobById Method Arguments**

F	Description
Id	Id of the job that shall be interrupted

**Table 24 – InterruptJobById Method AddressSpace Definition**

Attribute	Value				
BrowseName	InterruptJobById				
References	Node Class	BrowseName	DataType	TypeDefinition	Modelling Rule
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory

To continue the production of the job, *StartJobById* is called.

### 9.2.22 FinishJobById

With this method the client requests to finish the production of a specific job.

#### Signature

```
FinishJobById (
    [in]      String      Id);
```

**Table 25 – FinishJobById Method Arguments**

Argument	Description
Id	Id of the job that shall be finished.

**Table 26 – FinishJobById Method AddressSpace Definition**

Attribute	Value				
BrowseName	FinishJobById				
References	Node Class	BrowseName	Data Type	TypeDefinition	Modelling Rule
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory

**9.2.23 JobGroupStatusChangedEventType**

This *EventType* is used to inform the client about the change of the *Status Property* of a job group. It is fired by the parent *JobGroups object* to have only one Node for the subscription.

**Table 27 – JobGroupStatusChangedEventType Definition**

Attribute	Value				
BrowseName	JobGroupStatusChangedEventType				
IsAbstract	True				
References	Node Class	BrowseName	Data Type	TypeDefinition	Other
Subtype of 0:BaseEventType defined in OPC UA Part 5					
0:HasProperty	Variable	Id	0:String	0:PropertyType	M
0:HasProperty	Variable	LastStatus	3:JobStatusEnumeration	0:PropertyType	M
0:HasProperty	Variable	ActiveStatus	3:JobStatusEnumeration	0:PropertyType	M

*Id*:        *Id* of the job group.

*LastStatus*:    Value of the *Status Property* of the job group before the change

*ActiveStatus*:    Value of the *Status Property* of the job group after the change

The *SourceNode* (part of *BaseEventType*) shall be the *NodeId* of the relevant job group.

### 9.3 JobType

This *ObjectType* represent a single job inside a job group.

**Table 28 – JobType Definition**

Attribute	Value				
BrowseName	JobType				
IsAbstract	False				
References	Node Class	BrowseName	DataType	TypeDefinition	Other
Subtype of 0:BaseObjectType defined in OPC UA Part 5					
0:HasProperty	Variable	Id	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	Description	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	CustomerName	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	ProductId	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	ProductDescription	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	Strand	0:UInt32	0:PropertyType	M, RO
0:HasProperty	Variable	Sequence	0:UInt32	0:PropertyType	M, RO
0:HasProperty	Variable	ParameterSetting	ParameterSettingType[]	0:PropertyType	M, RO
0:HasProperty	Variable	SetOutput	0:Double	0:PropertyType	M, RO
0:HasProperty	Variable	LotSize	0:Double	0:PropertyType	M, RO
0:HasProperty	Variable	Status	3:JobStatusEnumeration	0:PropertyType	M, RO
0:HasComponent	Variable	GoodProduct	0:Boolean	0:BaseDataVariableType	M, RO
0:HasComponent	Variable	Throughput	0:Double	0:AnalogUnitType	O, RO
0:HasComponent	Variable	ProductWeight	0:Double	0:AnalogUnitType	O, RO
0:HasComponent	Variable	LineSpeed	0:Double	0:AnalogUnitType	O, RO
0:HasComponent	Variable	ActualLot	0:UInt32	0:BaseDataVariableType	M, RO
0:HasComponent	Variable	ActualLotName	0:String	0:BaseDataVariableType	O, RW
0:HasComponent	Variable	ActualOutput	0:Double	0:BaseDataVariableType	M, RO
0:HasComponent	Variable	ActualGoodOutput	0:Double	0:BaseDataVariableType	O, RO
0:HasComponent	Variable	ActualBadOutput	0:Double	0:BaseDataVariableType	O, RO
0:HasComponent	Variable	ActualSampleOutput	0:Double	0:BaseDataVariableType	O, RO
0:HasComponent	Variable	ActualOutputRate	0:Double	0:BaseDataVariableType	M, RO

#### 9.3.1 Id

Id of the job. It shall be unique inside the relevant job group. Two jobs inside different job groups may have the same Id.

#### 9.3.2 Description

Description of the job.

#### 9.3.3 CustomerName

Name of the customer for that the job is produced.

#### 9.3.4 ProductId

Id of the product produced by the job.

#### 9.3.5 ProductDescription

Description of the product produced by the job.

#### 9.3.6 Strand

*Strand* is used when several products (two pipes is parallel, foil cut into several smaller foils) are produced in parallel. When only one product is produced at the same time, the value is always 1.

#### 9.3.7 Sequence

Planned sequence when the job group contains multiple job for one strand. When there is only one job for a string in the production group the value is always 1.

**9.3.8 ParameterSetting**

Array for the configuration of the job by setting values for the offered configuration parameters. The *ParameterSettingType* is defined in OPC 40083. If an entry in the *ConfigurationParameters* Array in the job group has no corresponding entry in the *ParameterSetting* Array, the default value is used.

**9.3.9 SetOutput**

Set output of the job. The output is always counted in units, where a unit can be a piece of pipe/profile, roll of film, bag of compound, but also a defined bundle (e.g. one unit consist of 4 bags). What a unit is, may be configurable by the configuration parameters.

**9.3.10 LotSize**

Size of a production lot. This used to control the

- frequency of the *Events* of *LotFinishedEventType* (see 9.3.19)
- production of different jobs in a group: When LotSize is reached, the next job in the group is produced (see examples in Annex).

**9.3.11 Status**

Status of the job. The *JobStatusEnumeration* is defined in OPC 40083.

**9.3.12 GoodProduct**

Indication if the produced product has a good quality. Can be set by the machine or the operator. When it changes from true to false the line control shall create an alarm.

**9.3.13 ActualLot**

Which lot (number) is currently in production.

**9.3.14 ActualLotName**

Optional name for the actual lot.

**9.3.15 ActualOutput, ActualGoodOutput, ActualBadOutput, ActualSampleOutput**

These *Variables* represent the total number of produced (finished) units and the numbers of good, bad and test sample units in the current job. (see explanation above for unit in *SetOutput*).

**9.3.16 ActualOutputRate**

Average output rate of the job in units per hour.

**9.3.17 JobStatusChangedEventType**

This *EventType* is used to inform the client about the change of the *Status Property* of a job. It is fired by the *JobGroups* object to have only one Node for the subscription.

**Table 29 – JobStatusChangedEventType Definition**

Attribute	Value				
BrowseName	JobStatusChangedEventType				
IsAbstract	True				
References	Node Class	BrowseName	Data Type	Type Definition	Other
Subtype of 0:BaseEventType defined in OPC UA Part 5					
0:HasProperty	Variable	JobGroupId	0:String	0:PropertyType	M
0:HasProperty	Variable	JobId	0:String	0:PropertyType	M
0:HasProperty	Variable	LastStatus	3:JobStatusEnumeration	0:PropertyType	M
0:HasProperty	Variable	ActiveStatus	3:JobStatusEnumeration	0:PropertyType	M

*JobGroupId*: *Id* of the job group.

*JobId*: *Id* of the job.

*LastStatus*: Value of the *Status Property* of the job group before the change

*ActiveStatus*: Value of the *Status Property* of the job group after the change

The *SourceNode* (part of *BaseEventType*) shall be the *NodeId* of the relevant job.

### 9.3.18 UnitFinishedEventType

This *EventType* is used to inform the client a finished product unit. It is fired by the parent *JobGroups object* to have only one Node for the subscription.

**Table 30 – UnitFinishedEventType Definition**

Attribute	Value				
BrowseName	UnitFinishedEventType				
IsAbstract	True				
References	Node Class	BrowseName	DataType	TypeDefinition	Other
Subtype of 0:BaseEventType defined in OPC UA Part 5					
0:HasProperty	Variable	JobGroupId	0:String	0:PropertyType	M
0:HasProperty	Variable	JobId	0:String	0:PropertyType	M
0:HasProperty	Variable	Unit	0:UInt32	0:PropertyType	M
0:HasProperty	Variable	GoodProduct	0:Boolean	0:PropertyType	M

*JobGroupId*: *Id* of the job group.

*JobId*: *Id* of the job.

*Unit*: Counter, which unit out of the total amount has been finished.  
(e.g. roll number 4 roll is finished)

*GoodProduct*: Information if the finished unit has a good quality.

The *SourceNode* (part of *BaseEventType*) shall be the *NodeId* of the relevant job.

### 9.3.19 LotFinishedEventType

This *EventType* is used to inform the client a finished production lot. It is fired by the parent *JobGroups object* to have only one Node for the subscription.

**Table 31 – LotFinishedEventType Definition**

Attribute	Value				
BrowseName	LotFinishedEventType				
IsAbstract	True				
References	Node Class	BrowseName	DataType	TypeDefinition	Other
Subtype of 0:BaseEventType defined in OPC UA Part 5					
0:HasProperty	Variable	JobGroupId	0:String	0:PropertyType	M
0:HasProperty	Variable	JobId	0:String	0:PropertyType	M
0:HasProperty	Variable	Lot	0:UInt32	0:PropertyType	M

*JobGroupId*: *Id* of the job group.

*JobId*: *Id* of the job.

*Lot*: Counter, which lot has been finished. (e.g. lot number 6 roll is finished)

The *SourceNode* (part of *BaseEventType*) shall be the *NodeId* of the relevant job.

NOTE: The three events *JobStatusChanged*, *UnitFinished* and *LotFinished* may be fired simultaneously. E.g. when a job is finished, then the status is changed, the last product is finished as well as the last lot. If the job is the last in the job group, the *JobGroupStatusChanges* event will be fired in addition. However, the client decides which *EventTypes* are subscribed.



## 10 ProductionDatasetManagement

The *ProductionDatasetManagementType* is defined in OPC 40083 and provides functionalities for the management of recipes/machine settings.

## 11 Profiles and Conformance Units

This chapter defines the corresponding profiles and conformance units for the OPC UA Information Model for OPC 40084-2. *Profiles* are named groupings of conformance units. Facets are profiles that will be combined with other *Profiles* to define the complete functionality of an OPC UA *Server* or *Client*. The following tables specify the facets available for *Servers* that implement the OPC 40084-2 Information Model companion specification.

NOTE: The names of the supported profiles are available in the *Server Object* under *ServerCapabilities.ServerProfileArray*

Table 32 lists all Profiles defined in this document and defines their URIs.

**Table 32 – Profile URIs for OPC 40084-2**

Profile	URI
OPC 40084-2 Basic Server Profile	<a href="http://opcfoundation.org/UA-Profile/PlasticsRubber/Extrusion/ExtrusionLine/Server/Basic">http://opcfoundation.org/UA-Profile/PlasticsRubber/Extrusion/ExtrusionLine/Server/Basic</a>
OPC 40084-2 Jobs Server Facet	<a href="http://opcfoundation.org/UA-Profile/PlasticsRubber/Extrusion/ExtrusionLine/Server/Jobs">http://opcfoundation.org/UA-Profile/PlasticsRubber/Extrusion/ExtrusionLine/Server/Jobs</a>
OPC 40084-2 ProductionDatasetManagement Server Facet	<a href="http://opcfoundation.org/UA-Profile/PlasticsRubber/Extrusion/ExtrusionLine/Server/ProductionDatasetManagement">http://opcfoundation.org/UA-Profile/PlasticsRubber/Extrusion/ExtrusionLine/Server/ProductionDatasetManagement</a>

**Table 33 – OPC 40084-2 Basic Server Facet Definition**

Conformance Unit	Description	Optional/ Mandatory
OPC 40084-2 Basic	Support of <i>ExtrusionLine_InterfaceType</i> and all mandatory child elements giving information on the extrusion line and its status.	M
<b>Profile</b>		
ComplexType Server Facet (defined in OPC UA Part 7)		M
Standard Event Subscription Server Facet (defined in OPC UA Part 7)		M
Method Server Facet (defined in OPC UA Part 7)		M
BaseDevice_Server_Facet (defined in OPC UA Part 100)		M

**Table 34 – OPC 40084-2 Jobs Server Facet Definition**

Conformance Unit	Description	Optional/ Mandatory
OPC 40084-2 Jobs	An instance of <i>JobsType</i> (defined in OPC 40083) is provided for the status and management of jobs. See <i>JobsType</i> , which child elements are mandatory or optional. For the job description and actual values, subtypes for continuous production are used.	M

**Table 35 – OPC 40084-2 ProductionDatasetManagement Server Facet Definition**

Conformance Unit	Description	Optional/ Mandatory
OPC 40084-2 ProductionDatasetManagement	Support of <i>ProductionDatasetManagementType</i> (defined in OPC 40083) for the management and transfer of production datasets between MES and extrusion line	M

## 12 Namespaces

### 12.1 Namespace Metadata

Table 36 defines the namespace metadata for this specification. The *Object* is used to provide version information for the namespace and an indication about static *Nodes*. Static *Nodes* are identical for all *Attributes* in all *Servers*, including the *Value Attribute*. See Part5 for more details.

The information is provided as *Object* of type *NamespaceMetadataType*. This *Object* is a component of the *Namespaces Object* that is part of the *Server Object*. The *NamespaceMetadataType ObjectType* and its *Properties* are defined in Part5.

The version information is also provided as part of the *ModelTableEntry* in the *UANodeSet XML* file. The *UANodeSet XML* schema is defined in Part 6.

**Table 36 – NamespaceMetadata Object for this Specification**

Attribute		Value	
BrowseName		http://opcfoundation.org/UA/PlasticsRubber/Extrusion/ExtrusionLine/	
References	BrowseName	DataType	Value
HasProperty	NamespaceUri	String	http://opcfoundation.org/UA/PlasticsRubber/Extrusion/ExtrusionLine/
HasProperty	NamespaceVersion	String	1.00
HasProperty	NamespacePublicationDate	DateTime	2020-06-01 00:00:00
HasProperty	IsNamespaceSubset	Boolean	False
HasProperty	StaticNodeIdTypes	IdType[]	{Numeric}
HasProperty	StaticNumericNodeIdRange	NumericRange[]	Null
HasProperty	StaticStringNodeIdPattern	String	Null

## 12.2 Handling of OPC UA Namespaces

Namespaces are used by OPC UA to create unique identifiers across different naming authorities. The *Attributes NodeId* and *BrowseName* are identifiers. A *Node* in the *UA AddressSpace* is unambiguously identified using a *NodeId*. Unlike *NodeIds*, the *BrowseName* cannot be used to unambiguously identify a *Node*. Different *Nodes* may have the same *BrowseName*. They are used to build a browse path between two *Nodes* or to define a standard *Property*.

*Servers* may often choose to use the same namespace for the *NodeId* and the *BrowseName*. However, if they want to provide a standard *Property*, its *BrowseName* shall have the namespace of the standards body although the namespace of the *NodeId* reflects something else, for example the *EngineeringUnits Property*. All *NodeIds* of *Nodes* not defined in this document shall not use the standard namespaces.

Table 37 provides a list of mandatory and optional namespaces used in an OPC 40084-2 OPC UA *Server*.

**Table 37 – Namespaces used in an OPC 40084-2 Server**

NamespaceURI	Description	Use
http://opcfoundation.org/UA/	Namespace for <i>NodeIds</i> and <i>BrowseNames</i> defined in the OPC UA specification. This namespace shall have namespace index 0.	Mandatory
Local Server URI	Namespace for nodes defined in the local server. This may include types and instances used in a device represented by the server. This namespace shall have namespace index 1.	Mandatory
http://opcfoundation.org/UA/DI/	Namespace for <i>NodeIds</i> and <i>BrowseNames</i> defined in OPC UA Part 100. The namespace index is server specific.	Mandatory
http://opcfoundation.org/UA/PlasticsRubber/GeneralTypes/	Namespace for <i>NodeIds</i> and <i>BrowseNames</i> defined in OPC 40083. The namespace index is server specific.	Mandatory
http://opcfoundation.org/UA/PlasticsRubber/Extrusion/ExtrusionLine/	Namespace for <i>NodeIds</i> and <i>BrowseNames</i> defined in this specification. The namespace index is server specific.	Mandatory
Vendor specific types and instances	A server may provide vendor specific types like types derived from <i>MachineType</i> or <i>MachineStatusType</i> or vendor specific instances of devices in a vendor specific namespace.	Optional

Table 38 provides a list of namespaces and their index used for *BrowseNames* in this specification. The default namespace of this specification is not listed since all *BrowseNames* without prefix use this default namespace.

**Table 38 – Namespaces used in this specification**

NamespaceURI	Namespace Index	Example
http://opcfoundation.org/UA/	0	0:NodeVersion
http://opcfoundation.org/UA/DI/	2	2:DeviceClass
http://opcfoundation.org/UA/PlasticsRubber/GeneralTypes/	3	3:MachineInformationType

## Annex A (normative)

### OPC 40084-2 Namespace and mappings

#### A.1 Namespace and identifiers for OPC 40084-2 Information Model

This appendix defines the numeric identifiers for all of the numeric *NodeIds* defined in this specification. The identifiers are specified in a CSV file with the following syntax:

<SymbolName>, <Identifier>, <NodeClass>

Where the *SymbolName* is either the *BrowseName* of a *Type Node* or the *BrowsePath* for an *Instance Node* that appears in the specification and the *Identifier* is the numeric value for the *NodeId*.

The *BrowsePath* for an *Instance Node* is constructed by appending the *BrowseName* of the instance *Node* to the *BrowseName* for the containing instance or type. An underscore character is used to separate each *BrowseName* in the path. Let's take for example, the *MachineInformationType ObjectType Node* which has the *ControllerName Property*. The **Name** for the *ControllerName InstanceDeclaration* within the *MachineInformationType* declaration is: *MachineInformationType\_ControllerName*.

The *NamespaceUri* for all *NodeIds* defined here is  
<http://opcfoundation.org/UA/PlasticsRubber/Extrusion/ExtrusionLine/>

The CSV released with this version of the specification can be found here:

- <http://www.opcfoundation.org/UA/schemas/PlasticsRubber/Extrusion/ExtrusionLine/1.00/NodeIds.csv>

NOTE: The latest CSV that is compatible with this version of the specification can be found here:

- <http://www.opcfoundation.org/UA/schemas/PlasticsRubber/Extrusion/ExtrusionLine/NodeIds.csv>

A computer processible version of the complete Information Model defined in this specification is also provided. It follows the XML Information Model schema syntax defined in Part 6.

The Information Model Schema released with this version of the specification can be found here:

- <http://www.opcfoundation.org/UA/schemas/PlasticsRubber/Extrusion/ExtrusionLine/1.00/Opc.Ua.PlasticsRubber.Extrusion.ExtrusionLine.NodeSet2.xml>

NOTE: The latest Information Model schema that is compatible with this version of the specification can be found here:

- <http://www.opcfoundation.org/UA/schemas/PlasticsRubber/Extrusion/Extrusion/Opc.Ua.PlasticsRubber.Extrusion.ExtrusionLine.NodeSet2.xml>

---

## Annex B (informative)

### Examples for job descriptions

This Annex shows possible combinations of job groups and jobs for different applications. While the first example includes a full description of all variables, the other are shortened to highlight the differences.

#### Example 1: Production of a pipe, 100 pieces with 2m length

JobGroup\_1:

Variable	Value
Id	"30"
Description	"Pipe 2m, 100 pieces"
EquipmentDescription	"Die 342 with haul-off 35"
ProductionDatasetName	"Pipe911"
MaterialMapping[]	MaterialMapping[1]: MaterialId="734593" // raw PVC (as specified in MaterialList) MaterialLot="9876" HopperId="Hopper_1" // Id of HopperType defined in OPC 40082-2  MaterialMapping[2]: MaterialId="2534593" // green master batch MaterialLot="123" HopperId="Hopper_2"
Priority	1
PlannedStart	"2018-05-04T08:00:00Z"
PlannedProductionTime	800.000
PlannedSetUpTime	300.000
LatestEnd	"2018-05-05T11:00:00Z"
ConfigurationParameters	ConfigurationParameters[1]: Id=1 Description="Length" DefaultValue=1000 Unit: // see EUInformation DataType definition in OPC UA Part 8 namespaceUri: "http://www.opcfoundation.org/UA/units/un/cefact." unitId: 5066068 displayName: "mm" description: "millimetre"  ConfigurationParameters[2]: Id=4 Description="Diameter" DefaultValue=100 Unit: namespaceUri: "http://www.opcfoundation.org/UA/units/un/cefact." unitId: 5066068 displayName: "mm" description: "millimetre"

Job\_1:

Variable	Value
Id	"397"
Description	"2000mm_Pipe_100pcs"
CustomerName	"Company XY"
ProductId	"P53800"
ProductDescription	"2000mm_Pipe"
Strand	1
Sequence	1
ParameterSetting	ParameterSetting[1]: Id=1      // Length Value=2000  ParameterSetting[2]: Id=4      // Diameter Value=110
SetOutput	100
LotSize	100

Because there is only one job in the group, LotSize could also be another value with no effect on the production. Nevertheless, the value defines, after which number of produced product units, a lot finished event (see 9.3.19) will be fired.

## Example 2: Production of a pipe, first 100 pieces with 2m length, then 100 pieces with 1m length

JobGroup\_1:

Variable	Value
Id	"97"
Description	"Pipe, 100 pieces 2m, 100 pieces 1m"
EquipmentDescription	"Die 342 with haul-off 35"
ProductionDatasetName	"Pipe911"
MaterialMapping[]	PVC + green master match (see case 1)
ConfigurationParameters	Length + Diameter [mm] (see case 1)

Job\_1:

Variable	Value
Id	"413"
Description	"1000mm_Pipe_100pcs"
ProductId	"P53900"
ProductDescription	"1000mm_Pipe"
Strand	1
Sequence	1
ParameterSetting	Length=1000
SetOutput	100
LotSize	100

Job\_2:

Variable	Value
Id	"415"
Description	"2000mm_Pipe_100pcs"
ProductId	"P53800"
ProductDescription	"2000mm_Pipe"
Strand	1
Sequence	2
ParameterSetting	Length=2000
SetOutput	100
LotSize	100

First Job\_1 is produced completely first, because *LotSize* = *SetOutput*.

### Example 3: Production of a pipe with 1m and 2m length in change, 100 pieces for each

JobGroup\_1:

Variable	Value
Id	"102"
ProductionDatasetName	"Pipe911"
MaterialMapping[]	PVC + green master match (see case 1)
ConfigurationParameters	Length + Diameter (see case 1)

Job\_1:

Variable	Value
Id	"613"
Description	"1000mm_Pipe_100pcs"
ProductId	"P53900"
ProductDescription	"1000mm_Pipe"
Strand	1
Sequence	1
ParameterSetting	Length=1000
SetOutput	100
LotSize	1

Job\_2:

Variable	Value
Id	"651"
Description	"2000mm_Pipe_100pcs"
ProductId	"P53800"
ProductDescription	"2000mm_Pipe"
Strand	1
Sequence	2
ParameterSetting	Length=2000
SetOutput	100
LotSize	1

One piece of Job\_1 is produced. Because LotSize = 1 then a piece of Job\_2 is produced. As SetOutput is not reached it starts with Job\_1 again.

### Example 4: Production of two pipes in parallel (multistrand), 200 x 1m, 100 x 2m

JobGroup\_1:

Variable	Value
Id	"83"
ProductionDatasetName	"Pipe911_Multi"
MaterialMapping[]	PVC + green master match (see case 1)
ConfigurationParameters	Length + Diameter [mm] (see case 1)

Job\_1:

Variable	Value
Id	"756"
Description	"1000mm_Pipe_200pcs"
ProductId	"P53900"
ProductDescription	"1000mm_Pipe"
Strand	1
Sequence	1
ParameterSetting	Length=1000
SetOutput	100
LotSize	100

Job\_2:

Variable	Value
Id	"728"
Description	"2000mm_Pipe_100pcs"
ProductId	"P53800"
ProductDescription	"2000mm_Pipe"
Strand	2
Sequence	1
ParameterSetting	Length=2000
SetOutput	100
LotSize	100

Production in parallel because of different *Strands*.

**Example 5: Production of two pipes in parallel (multistrand) left strand: 1m and 4m in change, 100 pieces each right strand: 2m and 3m in change, 100 pieces each**

JobGroup\_1:

Variable	Value
Id	"254"
ProductionDatasetName	"Pipe911_Multi"
MaterialMapping[]	PVC + green master match (see case 1)
ConfigurationParameters	Length + Diameter [mm] (see case 1)

Job\_1:

Variable	Value
Id	"861"
Description	"1000mm_Pipe_100pcs"
ProductId	"P53900"
ProductDescription	"1000mm_Pipe"
Strand	1
Sequence	1
ParameterSetting	Length=1000
SetOutput	100
LotSize	1

Job\_3:

Variable	Value
Id	"529"
Description	"2000mm_Pipe_100pcs"
ProductId	"P53800"
ProductDescription	"2000mm_Pipe"
Strand	2
Sequence	1
ParameterSetting	Length=2000
SetOutput	100
LotSize	1

Job\_2:

Variable	Value
Id	"894"
Description	"4000mm_Pipe_100pcs"
ProductId	"P53700"
ProductDescription	"4000mm_Pipe"
Strand	1
Sequence	2
ParameterSetting	Length=4000
SetOutput	100
LotSize	1

Job\_4:

Variable	Value
Id	"516"
Description	"3000mm_Pipe_100pcs"
ProductId	"P53600"
ProductDescription	"3000mm_Pipe"
Strand	2
Sequence	2
ParameterSetting	Length=3000
SetOutput	100
LotSize	1

## Example 6: Production of granules, first 50 bags with 250kg, then 50 bags with 400kg

JobGroup\_1:

Variable	Value
Id	"918"
ProductionDatasetName	"Granules_Lite"
MaterialMapping[]	PVC
ConfigurationParameters	Id=7 Description="Weight" DefaultValue=250kg Unit: namespaceUri: "http://www.opcfoundation.org/UA/units/un/cefact." unitId: 4933453 displayName: "kg" description: " kilogram"

Job\_1:

Variable	Value
Id	"646"
Description	"50 bags 250kg Granules "
ProductId	"P46250"
ProductDescription	"250kg Granules"
Strand	1
Sequence	1
ParameterSetting	Weight=250 <i>// could also be empty because =DefaultValue</i>
SetOutput	50
LotSize	50

Job\_2:

Variable	Value
Id	"647"
Description	"50 bags 400kg Granules "
ProductId	"P46400"
ProductDescription	"400kg Granules "
Strand	1
Sequence	2
ParameterSetting	Weight=400
SetOutput	50
LotSize	50



**Example 7: Production of foil which is split into two lane left lane: 770x20mm (30 Pieces), 120x20mm (70 Pieces) right lane: 100x50mm (65 Pieces), 80x50mm (35 Pieces)**

JobGroup\_1:

Variable	Value
Id	"814"
ProductionDatasetName	"Foil_0.3 "
MaterialMapping[]	PET + blue master match
ConfigurationParameters	Length + Width [mm]

Job\_1:

Variable	Value
Id	"174"
Description	"30pcs_Foil_70/20"
ProductId	"P86600"
ProductDescription	"Foil_70/20"
Strand	1
Sequence	1
ParameterSetting	Length=70 Width=20
SetOutput	30
LotSize	1

Job\_3:

Variable	Value
Id	"152"
Description	"65pcs_Foil_100/50"
ProductId	"P86800"
ProductDescription	"Foil_70/20"
Strand	2
Sequence	1
ParameterSetting	Length=100 Width=50
SetOutput	65
LotSize	1

Job\_2:

Variable	Value
Id	"162"
Description	"70pcs_Foil_120/20"
ProductId	"P86700"
ProductDescription	"Foil_70/20"
Strand	1
Sequence	2
ParameterSetting	Length=120 Width=20
SetOutput	70
LotSize	1

Job\_4:

Variable	Value
Id	"198"
Description	"35pcs_Foil_80/50"
ProductId	"P86900"
ProductDescription	"Foil_80/50"
Strand	2
Sequence	2
ParameterSetting	Length=80 Width=50
SetOutput	35
LotSize	1