

**VDMA 40001-3**

ICS 25.020; 35.240.50

Replaces  
VDMA 40001-3:2024-04**OPC UA for Machinery –  
Part 3: Job Management**OPC UA for Machinery –  
Teil 3: Auftragsverwaltung**VDMA 40001-3:2024-05 is identical with OPC 40001-3 (Release 1.0.1)**

Document comprises 35 pages

VDMA

## Contents

	Page
Forewords.....	8
1 Scope .....	9
2 Normative references .....	9
3 Terms, definitions and conventions .....	9
3.1 Overview .....	9
3.2 OPC UA for Machinery – Job Management terms .....	9
3.3 Abbreviated terms .....	10
3.4 Conventions used in this document.....	10
4 General information to Machinery and OPC UA.....	10
5 Use cases .....	10
6 Machinery Job Management Information Model overview.....	10
6.1 Overview .....	10
6.2 JobManagementType .....	10
6.3 Integration into a MachineryItem .....	11
6.4 Order of Execution of Job orders .....	12
6.5 Predefined parameters for job management .....	13
7 Predefined Job-Order-Input and Job-Order-Response Information .....	13
7.1 Overview .....	13
7.2 Predefined JobOrderParameters and JobResponseData .....	13
7.3 Personnel.....	15
7.4 Equipment .....	15
7.5 Physical Assets .....	16
7.6 Material .....	16
8 OPC UA ObjectTypes .....	17
8.1 JobManagementType ObjectType Definition.....	17
9 OPC UA DataTypes.....	17
9.1 JobExecutionMode.....	17
9.2 JobResult.....	18
9.3 OutputInformationDataType .....	18
9.4 OutputInfoType .....	19
9.5 BOMComponentInformationDataType .....	19
9.6 BOMInformationDataType .....	20
9.7 OutputPerformanceInfoDataType .....	20
9.8 ProcessIrregularity .....	21
10 Profiles and ConformanceUnits .....	21
10.1 Conformance Units.....	21
10.2 Profiles.....	23

<b>10.2.1</b>	<b>Profile list .....</b>	<b>23</b>
<b>10.2.2</b>	<b>Server Facets .....</b>	<b>23</b>
<b>10.2.3</b>	<b>Client Facets .....</b>	<b>24</b>
<b>11</b>	<b>Namespaces.....</b>	<b>25</b>
<b>11.1</b>	<b>Namespace Metadata .....</b>	<b>25</b>
<b>11.2</b>	<b>Handling of OPC UA Namespaces.....</b>	<b>25</b>
<b>Annex A (normative)</b>	<b>Machinery – Job Management Namespace and mappings .....</b>	<b>27</b>
<b>Annex B (informative)</b>	<b>Examples .....</b>	<b>28</b>

**Figures**

Figure 1 – Overview JobManagementType ..... 11

Figure 2 – Integration of Job Management into MachineryItem ..... 12

Figure 3 – Example of a planned job order ..... 31

## Tables

Table 1 – Predefined JobOrderParameters and JobResponseData .....	13
Table 2 – Predefined additional Properties for ISA95EquipmentDataType .....	16
Table 3 – Predefined additional Properties for ISA95MaterialDataType .....	16
Table 4 – JobManagementType Definition.....	17
Table 5 – JobManagementType Attribute values for child Nodes.....	17
Table 6 – JobExecutionMode Items .....	17
Table 7 – JobExecutionMode Definition .....	18
Table 8 – JobResult Items.....	18
Table 9 – JobResult Definition .....	18
Table 10 – OutputInformationDataType Structure .....	18
Table 11 – OutputInformationDataType Definition.....	19
Table 12 – OutputInfoType Items.....	19
Table 13 – OutputInfoType Definition.....	19
Table 14 – BOMComponentInformationDataType Structure .....	19
Table 15 – BOMComponentInformationDataType Definition .....	20
Table 16 – BOMInformationDataType Structure .....	20
Table 17 – BOMInformationDataType Definition.....	20
Table 18 – OutputPerformanceInfoDataType Structure.....	20
Table 19 – OutputPerformanceInfoDataType Definition .....	20
Table 20 – ProcessIrregularity Items.....	21
Table 21 – ProcessIrregularity Definition .....	21
Table 22 – Conformance Units for Machinery – Job Management.....	21
Table 23 – Profile URIs for Machinery – Job Management .....	23
Table 24 – Machinery Job Management Base Server Facet.....	24
Table 25 – NamespaceMetadata Object for this Document .....	25
Table 26 – Namespaces used in a Machinery Job Management Server .....	26
Table 27 – Namespaces used in this document .....	26
Table 28 – Example of a job order result (Glass).....	28
Table 29 – Example of a job order .....	31
Table 30 – Example of a job order result .....	33

## OPC Foundation / VDMA

---

### AGREEMENT OF USE

#### COPYRIGHT RESTRICTIONS

- This document is provided "as is" by the OPC Foundation and VDMA.
- 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 VDMA 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 VDMA 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 VDMA.

#### PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OPC or VDMA specifications may require use of an invention covered by patent rights. OPC Foundation or VDMA shall not be responsible for identifying patents for which a license may be required by any OPC or VDMA specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OPC or VDMA 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 VDMA 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 VDMA 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 VDMA 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 VDMA 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.

## Forewords

Mechanical engineering is a broad-based industry, which is mainly associated with machines such as machine tools, woodworking machines or robots. Many other products such as measuring and testing equipment are also relevant to this field.

Since the information models in this document are intended to apply not only to machines (see in ISO 12100:2010, [1]), but to all other applications and products in the entire machinery industry. Each case cannot be represented individually, therefore the term “machine” is used uniformly for all in this document.

Compared with previous versions, the following changes have been made:

Version	Changes
OPC 40001-3 1.0 (identical with VDMA 40001-3:2024-04)	Initial release
OPC 40001-3 1.0.1 (identical with VDMA 40001-3:2024-05)	Fixed Figure 1: ISA95JobResponseProviderObjectType has now two Methods instead of one as in the referenced specification. Fixed the same in UaNodeSet on JobManagementType.

This specification was created by a joint working group of the OPC Foundation and VDMA.

### 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

### VDMA

The VDMA is Europe's largest industry association with over 3600 member companies of the mechanical engineering industry. These companies integrate the latest technologies in products and processes. VDMA was founded in November 1892 and is the most important voice for the mechanical engineering industry today. With the headquarters located in Frankfurt, it represents the issues of the mechanical and plant engineering sector in Germany and Europe. The standard OPC UA has established itself in this industry sector. The VDMA defines OPC UA Companion Specifications for various sectors of the mechanical engineering industry, with more than 500 companies involved. Consequently, one of the main tasks is to harmonise and create consistency.



## 1 Scope

The OPC UA for Machinery specification contains various building blocks for Machinery that allow to address use cases across different types of machines and components of machines defined in various companion specifications.

For the general scope of the OPC UA for Machinery specification see OPC 40001-1.

This part contains a building block for

- Job Management

## 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 and errata) applies

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

<http://www.opcfoundation.org/documents/10000-1/>

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

<http://www.opcfoundation.org/documents/10000-3/>

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

<http://www.opcfoundation.org/documents/10000-5/>

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

<http://www.opcfoundation.org/documents/10000-6/>

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

<http://www.opcfoundation.org/documents/10000-7/>

OPC 40001-1, *OPC UA for Machinery - Part 1: Basic Building Blocks*

<http://www.opcfoundation.org/documents/40001-1/>

OPC 10031-4, *OPC UA for ISA-95 – Part 4: Job Control*

<http://www.opcfoundation.org/documents/10031-4/>

## 3 Terms, definitions and conventions

### 3.1 Overview

It is assumed that basic concepts of OPC UA information modelling, OPC 40001-1 and OPC 10031-4 are understood in this specification. This specification will use these concepts to describe the Machinery – Job Management Information Model. For the purposes of this document, the terms and definitions given in OPC 10000-1, OPC 10000-3, OPC 10000-5, OPC 10000-7, OPC 40001-1, OPC 10031-4 as well as the following apply.

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

### 3.2 OPC UA for Machinery – Job Management terms

#### 3.2.1

##### **job order**

unit of scheduled work that is dispatched for execution (IEC 62264-3:2016(en))

Note: A job order is the concrete implementation of a set of information that a machine needs to execute a task. This information can be composed of one or more programs or recipes, or it can consist only of metadata and parameters (production-specific) or of all of these. Several job orders can be running at the same time.

### 3.2.2

#### run

A completed execution or completed subset of the execution carried out according to the data provided with the job order.

### 3.2.3

#### Item

an object produced as one quantity unit

Note: If pieces are produced, the quantity unit is always one piece. If no pieces are produced (e.g., kg or m), the quantity unit of an item can be determined by the planned quantity per run.

Examples:

- 15 pieces produced in 3 runs (5 pieces per run) -> quantity unit is one piece, 15 items produced
- 5,6kg produced in one run -> quantity unit is 5,6kg, 1 item produced
- 20m produced in two runs (10m per run) -> quantity unit is 10m, 2 items produced

## 3.3 Abbreviated terms

BOM Bill of materials

ERP Enterprise-Resource-Planning

MES Manufacturing Execution System

MOM Manufacturing Operations Management

## 3.4 Conventions used in this document

For conventions used in this document see OPC 40001-1.

# 4 General information to Machinery and OPC UA

For general information to Machinery and OPC UA see OPC 40001-1.

# 5 Use cases

The user would like to provide job orders to a *MachineryItem* to get executed.

The user would like to control job orders by updating the job order, setting the job order to get executed or revoke the execution, pause and resume the execution, and abort or stop the executing.

The user would like to get information about the state of execution, retrieve intermediate results and the end result of the job order execution.

The user would like to delete the job order results from the *MachineryItem* after execution and receiving the job order.

# 6 Machinery Job Management Information Model overview

## 6.1 Overview

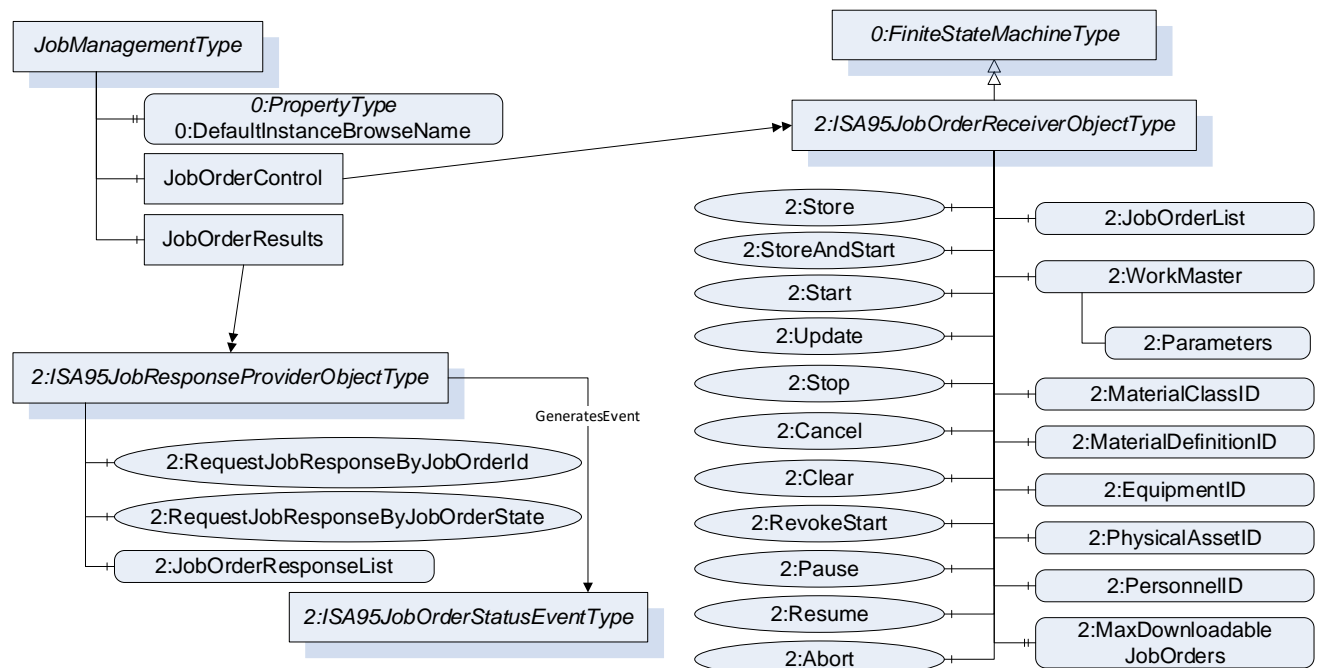
This chapter gives an overview over the job management information model. In 6.2, the *ObjectTypes* for managing job orders are described. 6.3 describes the integration into a *MachineryItem*. In 6.4, the mechanism for defining the order for the execution of job orders is defined, and in 6.5 the concept of predefined parameters for the description of job orders is introduced.

## 6.2 JobManagementType

In Figure 1 an overview of the *JobManagementType* is given. It is formally defined in 8.1. The *JobOrderControl Object* of *2:ISA95JobOrderReceiverObjectType* provides functionality to add and control job orders to a *MachineryItem*. It also provides information about all job orders currently managed by the *MachineryItem* (in the *2:JobOrderList*). It contains the description of a *StateMachine* that is applied to the job orders managed in the *2:JobOrderList* (not shown in the diagram) that can be specialized and restricted. All job orders managed by the

*JobOrderControl* support the same, potentially specialized *StateMachine*. Details of the 2:ISA95*JobOrderReceiverObjectType* are defined in OPC 10031-4.

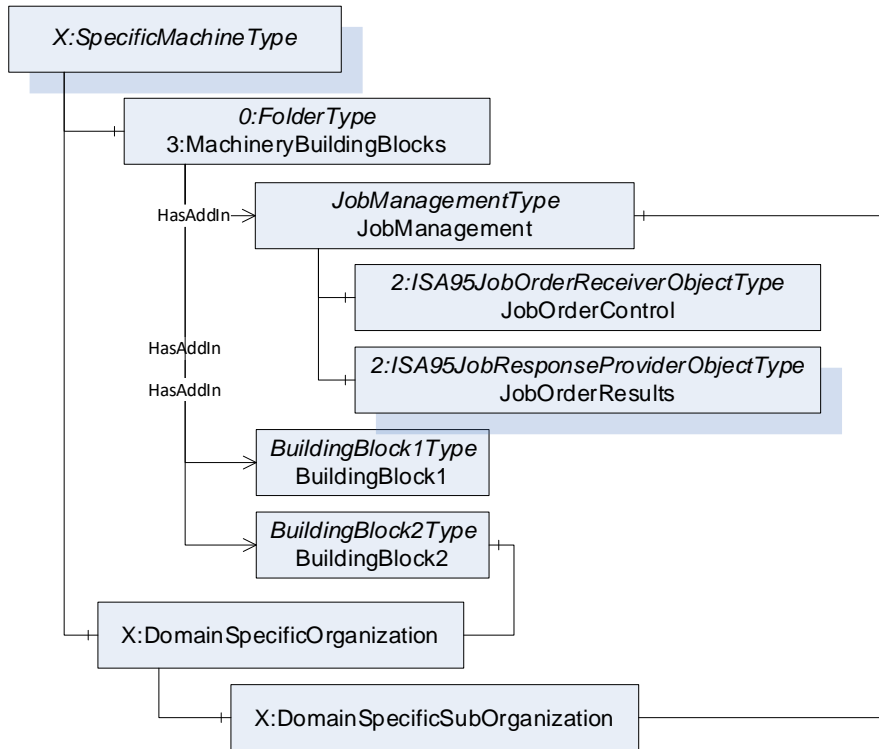
Once a job order starts to be executed by the *MachineryItem*, it becomes available in the *JobOrderResults Object* to provide intermediate results or the final results of the job order execution. During that phase, the job orders can still be controlled by the *JobOrderControl Object*. The *JobOrderResults Object* is of *2:ISAJobResponseProviderObjectType* and provides different mechanisms (*Method*, *Event*, or *Variable*) to expose the results (see OPC 10031-4 for details). Once the *Client* has received the end results, it can remove the results from the *MachineryItem*, and thereby also remove the meta data of the job order managed by the *JobOrderControl Object*.



### Figure 1 – Overview JobManagementType

### 6.3 Integration into a MachineryItem

In Figure 2 an example is shown, how the job management can be integrated into a *MachineryItem*. The building block, using its `0:DefaultInstanceBrowseName JobManagement`, is added as *AddIn* into the `3:MachineryBuildingBlocks` Object. In addition, it is referenced by some domain-specific organization.



**Figure 2 – Integration of Job Management into MachineryItem**

Note that a *MachineryItem* may also add several job management instances in order to support different specialized *StateMachines* for the job orders. In this case, the *2:DefaultInstanceBrowseName* should only be used once.

#### 6.4 Order of Execution of Job orders

*MachineryItems* may execute several job orders in parallel or just one job order at a time. Depending on the application, the order of the execution may be defined by different mechanisms. In some cases, the order is determined by a global system and provided with the job order meta data, in other cases, the operator of a machine might choose of a list of job orders.

The general rules on how the execution of job orders is determined:

- Job orders can only be executed if they are in the corresponding state (see OPC 10031-4). Applications may choose to only set one job order into that state explicitly to start a job order (e.g. by the operator of the *MachineryItem*).
- If the job order is in the state to be executed, and more than one job order is in that state, the *2:StartTime* of *2:ISA95JobOrderDataType* indicates the order the job orders shall be executed. The earliest time indicates that the job order shall be started next. Applications may choose to set the same *2:StartTime* to several job orders in order to indicate that all those job orders could be executed next.
- If there are several job orders having the same *2:StartTime*, the *2:Priority* of *2:ISA95JobOrderDataType* indicates, which job order shall be executed next. The job order with the highest *2:Priority* shall be executed next. Applications may choose not to set a priority or let several job orders have the same *2:Priority* (and same *2:StartTime*).
- In case neither the *2:StartTime* nor the *2:Priority* indicate the order; it is application specific which job order starts next. Companion Specifications may define additional mechanisms.

The order of the job order execution shall be reflected in the *2:JobOrderList*. That is, the array shall start with all executed job orders (in the order the execution started), followed by all job orders currently executed or interrupted (in the order the execution started), followed by the job orders that can start to be executed, followed by the job orders that currently cannot start to be executed.

The job orders that can be executed shall be ordered based on *2:StartTime* and *2:Priority*, also the job orders that currently cannot start to be executed. Note that the order of both is not necessarily absolute, i.e. several

entries in the array may be on the same level but put into an arbitrary order and may not be started in the order of the array.

## 6.5 Predefined parameters for job management

In OPC 10031-4 the bare minimum of meta data is defined describing a job order and an extensibility mechanism is built into the data structures allowing to provide additional meta data. This specification uses the mechanism to define additional meta data in 7. Companion specifications may define additional meta data. Profiles defined in 10 or other companion specifications may require the presence of specific meta data.

## 7 Predefined Job-Order-Input and Job-Order-Response Information

### 7.1 Overview

ISA95 job control (OPC 10031-4) defines mechanisms to add job order information using the *2:ISA95JobOrderDataType* and mechanisms getting the result or current status of the job order using the *2:ISA95JobResponseDataType*. Both *DataTypes* define arrays of properties of a job order: general, personnel, equipment, physical assets, and material. The *2:ISA95JobOrderDataType* uses the general properties to describe the job order and the other properties to define the requirements, whereas the *2:ISA95JobResponseDataType* uses the general properties to describe the output and the other properties to provide the information what has been used.

This specification standardizes some of those parameters, which are application-specific from the view of OPC 10031-4.

### 7.2 Predefined JobOrderParameters and JobResponseData

In Table 1, predefined key-value pairs for *2:JobOrderParameters* and *2:JobResponseData* is provided. The table indicates, in which data structure the key-value pair is expected to be used. An "X" in "In" indicates it may be used in *2:JobOrderParameters* and an "X" in "Out" indicates it may be used in *2:JobResponseData*.

**Table 1 – Predefined JobOrderParameters and JobResponseData**

ID	Data Type of Value	Description	Engineering Units	Sub-parameters	In	Out
JobName	0:LocalizedText[]	Human readable name of the job. Array shall always contain the same text, potentially in different languages.	-	-	X	X
OrderNumbers	0:String[]	The OrderNumbers are used to reference company internal ERP orders the job order belongs to. Shall be provided in JobOrderParameters if any planned produced material uses OrderNumber as Identification and shall contain all those OrderNumbers. Shall be provided in JobResponseData if any produced material, any AsBuiltBOM or OutputPerformanceInfo uses OrderNumber as Identification and shall contain all those OrderNumbers.			X	X
Customers	0:String[]	This property is used to show the name of the customers of the Job order.	-	-	X	X
CustomerOrderNumbers	0:String[]	The CustomerOrderNumbers is used to reference to the customer orders this job order belongs to.	-	-	X	X
JobExecutionMode	JobExecutionMode	Describes the execution mode of the machine for that job order.	-	-	X	X
ReasonForStateChange	0:LocalizedText	Reason why a state change occurred. May be filled by the Comment parameter of the methods used to change the state.	-	-	X	X
RunsPlanned	0:UInt32	RunsPlanned indicates the number of runs that should be performed by the targeted machinery item when this job order is	-	-	X	-

		executed. A run can consist of several steps. If the planned number of job order runs is not known to the machinery item it is set to 0. The number of planned job order runs not being known occurs in continuous production, that is if the machinery item is started with the respective job order and job order runs are repeated endlessly. The production process only ends when the machinery item is stopped by an external measure (operator or system).				
PlannedProductionTime	0:Duration	The planned pure time period of the production (or value creation) without the consideration of setup and delay time periods. Equals planned order quantity (POQ) times planned run time per item (PRI) in correspondence to ISO22400.	-	-	X	-
PlannedSetupTime	0:Duration	The planned time period before the production process can be started. (e.g. warm-up)	-	-	X	-
PlannedTimePerRun	0:Duration	The planned time period that is needed for the execution of one run of this job order.	-	-	X	-
PlannedQuantityPerRun	0:Double	The planned quantity produced by one run. This shall be higher than 0. Note: If the quantity is measured in pieces, the EngineeringUnits has to be "piece" (H87).	Product-specific	-	X	-
PlannedOrderQuantity	0:Double	The planned order quantity (POQ) shall be the planned quantity of products for a production order (lot size, production order quantity). [Source: ISO 22400]	Product-specific	Over-production	X	-
Overproduction	0:Boolean	Indicates if the machine continues the production even if the nominal output has been reached.	-	-	X	-
PlannedDuration	0:Duration	Planned time period needed for the execution of the job order. Consists of setup time, production time and planned delays. This information is provided by the MES/MOM.	-	-	X	-
JobAnnotation	0:LocalizedText[]	Description text provided to an operator to indicate the equipment or modification of the machine needed for the job order. Note: This is not interpretable by machines.	-	-	X	-
RunsCompleted	0:UInt32	RunsCompleted is a counter that increases after each completed run of this job order.	-	-	-	X
RunsStarted	0:UInt32	RunsStarted is a counter that indicates how many of the RunsPlanned have been started already. When the processing of all runs was successful, the number of RunsStarted should equal RunsCompleted and RunsPlanned. It is not corresponding to the quality of produced items.	-	-	-	X
ActualQuantityCurrentRun	0:Double	The actual total quantity that was produced in the current run. It can be incremented either sequentially or in steps. Note: If the quantity is measured in pieces, the EngineeringUnits has to be "piece" (H87).	Product-specific	-	-	X
ActualUnitBusyTime	0:Duration	The elapsed time since the start of production of the job order. This value is intended to be updated in a low frequency. Corresponding ISO22400 definition: AUBT (actual unit busy time)	-	-	-	X
ActualUnitSetupTime	0:Duration	The elapsed time for the setup of the job order. This value is intended to be updated in a low frequency.	-	-	-	X

		Corresponding ISO22400 definition: AUST (actual unit setup time)				
ActualUnitDelayTime	0:Duration	The elapsed time due to delays. This value is intended to be updated in a low frequency. Corresponding ISO22400 definition: ADET (actual unit delay time)	-	-	-	X
ActualProductionTime	0:Duration	The actual production time shall be the actual time during which the work unit produces the job. It includes only the value adding functions. Corresponding ISO22400 definition: APT (actual production time) Note: The actual production time includes time losses depending on speed and quality issues.	-	-	-	X
ProducedQuantity	0:Double	The produced quantity reflects the quantity that a work unit has produced in relation to a production order including the count of good quantity and scrap quantity and rework quantity. [Source: ISO 22400] Corresponding ISO22400 definition: PQ (produced quantity)]	Product-specific	-	-	X
EstimatedRemainingTime	0:Duration	Planned remaining time that is needed to complete this job order. This is an approx. value that can't always be determined.	-	-	-	X
JobResult	JobResult	High-level result of the execution of the job order.	-	-	-	X
GoodQuantity	0:Double	The good quantity shall be the produced quantity that meets quality requirements. Note: Measuring work units use good quantity as number of successful executed measurement programs. Source ISO 22400 A quantity is considered as good as long as there is no contradicting evidence. Note that such evidence may arise in subsequent processing steps (on different machines), even if a quantity was considered as good. In this case, the data on the OPC UA Server are not changed retrospectively. Corresponding ISO22400 definition: GQ (good quantity)	Product-specific	-	-	X
AsBuiltBOM	BOMInformation-DataTypes[]	The bill of material of the produced output of the job order.	-	-	-	X
OutputPerformanceInfo	OutputPerformanceInfoDataTypes[]	Performance information of the executed job order.	-	-	-	X

Note: PRI (planned run time per item) as defined in ISO22400 equals *PlannedTimePerRun* divided by *PlannedQuantityPerRun*.

### 7.3 Personnel

This specification does not define any standardized entries for *2:PersonnelRequirements* or *2:PersonnelActuals*. However, the operators executing the job order may be returned in the *2:PersonnelActuals*, where each entry represents one operator (in the ID of *2:ISA95PersonnelDataType*). The optional fields may be used as defined in OPC 10031-4.

### 7.4 Equipment

This specification does not define any standardized entries for *2:EquipmentRequirements* and *2:EquipmentActuals*. However, the planned equipment may be provided in the *2:EquipmentRequirements*, and the actual used in *2:EquipmentActuals*.

If an equipment is a component of a *MachineryItem* modelled in the OPC UA Server, and the *4:AssetId* is provided, the *4:AssetId* shall be used as ID in the *2:ISA95EquipmentDataType*.

If the equipment are components of a *MachineryItem* modelled in the OPC UA Server, and the *4:ComponentName* is provided, it should be provided as additional entry in the *2:Properties* array of the *2:ISA95EquipmentDataType*, as specified in Table 2.

**Table 2 – Predefined additional Properties for ISA95EquipmentDataType**

ID	DataType of Value	Description	EngineeringUnits	Subparameters	In	Out
ComponentName	0:LocalizedText[]	Name of the MachineryItem provided by the end user. Array shall always contain the same text, potentially in different languages.	-	-	X	X

## 7.5 Physical Assets

This specification does not define any standardized entries for *2:PhysicalAssetRequirements* and *2:PhysicalAssetActuals*. However, the planned physical assets may be provided in the *2:PhysicalAssetRequirements*, and the actual used in *2:PhysicalAssetActuals*.

## 7.6 Material

For all Material provided, the optional *2:Quantity* and *2:EngineeringUnits* as well as the *2:MaterialDefinitionID* shall be provided.

If the *2:MaterialDefinitionID* and the *Identification* is provided, the *ItemNumber* of the *Identification* shall be identical to the *2:MaterialDefinitionID*.

If the *2:MaterialLotID* and the *Identification* is provided, the *LotNumber* shall be identical to the *2:MaterialLotID*.

In Table 3, predefined key-value pairs for *2:MaterialRequirements* and *2:MaterialActuals* is provided. The table indicates, in which data structure the key-value pair is expected to be used.

**Table 3 – Predefined additional Properties for ISA95MaterialDataType**

ID	DataType of Value	Description	Engineering Units	Subparameters	In	Out
DrawingNumber	String	Number of the drawing related to an item	-	DrawingVersion Number	X	X
DrawingVersionNumber	String	Drawing version	-	-	X	X
Location	String	Location defines source or destination location of the material, depending if it is an input or an output material.	-	-	X	X
RelatedContainer	String	Defines the storage location from which the material is taken or in which it is placed.	-	-	X	X
Identification	OutputInformationDataType	Identification of the output	-	-	X	X
StartTime	0:DateTime	Output of first item	-	-	-	X
EndTime	0:DateTime	Output of last item	-	-	-	X
ProcessIrregularity	ProcessIrregularity	Defines if a process irregularity has been detected. A process irregularity might for example be the breakage of a tool, or exceeding a temperature limit on coolant.	-	-	-	X



## 8 OPC UA ObjectTypes

### 8.1 JobManagementType ObjectType Definition

The *JobManagementType* provides an *AddIn* for job management and is formally defined in Table 4.

**Table 4 – JobManagementType Definition**

Attribute	Value				
BrowseName	JobManagementType				
IsAbstract	False				
References	Node Class	BrowseName	Data Type	TypeDefinition	Other
Subtype of the 0:BaseObjectType					
0:HasProperty	Variable	0:DefaultInstance-BrowseName	0:QualifiedName	0:PropertyType	
0:HasComponent	Object	JobOrderControl		2:ISA95JobOrderReceiverObjectType	M
0:HasComponent	Object	JobOrderResults		2:ISA95JobResponseProviderObjectType	M
<b>Conformance Units</b>					
Machinery Job Management Base					

The *JobOrderControl* Object provides functionality to add job orders and control them (pausing, aborting, etc.). It provides information about the currently managed job orders of the job management. It shall be used as defined in OPC 10031-4.

The *JobOrderResults* Object provides information about the current status about running job orders. It shall be used as defined in OPC 10031-4.

In case the *JobOrderControl* provides the 2:JobOrderList and the *JobOrderResults* provides the 2:JobOrderResponseList, the following rules apply.

- For each entry in the 2:JobOrderResponseList shall be a corresponding entry in the 2:JobOrderList (having the same 2:JobOrderId).
- For each entry of the 2:JobOrderList having a 2:State different than 2:AllowedToStart or 2:NotAllowedToStart (including substates), there shall be a corresponding entry in the 2:JobOrderResponseList (having the same 2:JobOrderId).

The child *Nodes* of the *JobManagementType* have additional *Attribute* values defined in Table 5.

**Table 5 – JobManagementType Attribute values for child Nodes**

BrowsePath	Value Attribute	Description Attribute
0:DefaultInstanceBrowseName	JobManagement	The default BrowseName for instances of the type.

## 9 OPC UA DataTypes

### 9.1 JobExecutionMode

This enumeration *JobExecutionMode* describes the execution mode of the machine for the job order. The enumeration is defined in Table 6.

**Table 6 – JobExecutionMode Items**

Name	Value	Description
SimulationMode	0	Machine running in simulation mode with no item involved.
TestMode	1	Machine running in test mode with an item involved.
ProductionMode	2	Machine running in production mode.

Its representation in the *AddressSpace* is defined in Table 7.

**Table 7 – JobExecutionMode Definition**

Attribute	Value				
BrowseName	JobExecutionMode				
IsAbstract	False				
References	Node Class	BrowseName	Data Type	Type Definition	Other
Subtype of the Enumeration type defined in OPC 10000-5					
0:HasProperty	Variable	0:EnumValues	0:EnumValueType[]	0:PropertyType	
<b>Conformance Units</b>					
Machinery Job Management Planned ExecutionMode					
Machinery Job Management Result ExecutionMode					

## 9.2 JobResult

This enumeration *JobResult* describes the high-level result of an executed job order. The enumeration is defined in Table 8.

**Table 8 – JobResult Items**

Name	Value	Description
Unknown	0	Unknown state. Used when result is not known, e.g. because job order is still running.
Successful	1	Job order was executed successfully.
Unsuccessful	2	Job order was not executed successfully.

Its representation in the *AddressSpace* is defined in Table 9.

**Table 9 – JobResult Definition**

Attribute	Value				
BrowseName	JobResult				
IsAbstract	False				
References	Node Class	BrowseName	Data Type	Type Definition	Other
Subtype of the Enumeration type defined in OPC 10000-5					
0:HasProperty	Variable	0:EnumValues	0:EnumValueType[]	0:PropertyType	
<b>Conformance Units</b>					
Machinery Job Management Result JobResult					

## 9.3 OutputInformationDataType

This structure contains identification information. The structure is defined in Table 10.

**Table 10 – OutputInformationDataType Structure**

Name	Type	Description	Optional
OutputInformationDataType	structure		
ItemNumber	String	ItemNumber defines an Identifier to identify the type of the item (Material Identifier).	False
OutputInfo	OutputInfoType	Bitmask indicating which of the optional fields are used for identification. If none is selected, only ItemNumber is used. Each selected optional field shall provide a value.	False
OrderNumber	String	OrderNumber defines an Identifier to identify the order. Shall be provided if defined in OutputInfo.	True
LotNumber	String	LotNumber defines an Identifier to identify the production-group of the item (Lot Identifier). Shall be provided if defined in OutputInfo.	True
SerialNumber	String	SerialNumber defines an Identifier to identify the one entity of the item (Product Identifier). Shall be provided if defined in OutputInfo.	True

Its representation in the *AddressSpace* is defined in Table 11.

**Table 11 – OutputInformationDataType Definition**

Attribute	Value				
BrowseName	OutputInformationDataType				
IsAbstract	False				
References	Node Class	BrowseName	DataType	TypeDefinition	Other
Subtype of the 0:Structure defined in OPC 10000-5					
<b>Conformance Units</b>					
Machinery Job Management Planned Base					
Machinery Job Management Result Base					

## 9.4 OutputInfoType

This *OutputInfoType* is a bitmask that defines the usage of a provided output. It is defined in Table 12.

**Table 12 – OutputInfoType Items**

Name	Bit No.	Description
OrderNumber	0	Identifier identifies the order (order identifier).
LotNumber	1	Identifier identifies the production group of the item (lot identifier).
SerialNumber	2	Identifier identifies the one entity of the item (product identifier).

Its representation in the *AddressSpace* is defined in Table 13.

**Table 13 – OutputInfoType Definition**

Attribute	Value				
BrowseName	OutputInfoType				
IsAbstract	False				
References	Node Class	BrowseName	DataType	TypeDefinition	Other
Subtype of the Byte type defined in OPC 10000-5					
0:HasProperty	Variable	0:OptionSetValues	0:LocalizedText[]	0:PropertyType	
<b>Conformance Units</b>					
Machinery Job Management Planned Base					
Machinery Job Management Result Base					

## 9.5 BOMComponentInformationDataType

This structure contains information about a component of the bill of material. The structure is defined in Table 14.

**Table 14 – BOMComponentInformationDataType Structure**

Name	Type	Description
BOMComponentInformationDataType	structure	
Identification	OutputInformationDataType	Identification of the output.
Quantity	0:Double	Quantity defines the amount of material. This quantity can be specified in different ways, e.g. weight or number.
EngineeringUnits	0:EUIInformation	The engineering unit of the quantity.

Its representation in the *AddressSpace* is defined in Table 15.

**Table 15 – BOMComponentInformationDataType Definition**

Attribute	Value				
BrowseName	BOMComponentInformationDataType				
IsAbstract	False				
References	Node Class	BrowseName	DataType	TypeDefinition	Other
Subtype of the 0:Structure defined in OPC 10000-5					
Conformance Units					
Machinery Job Management Result BOM					

## 9.6 BOMInformationDataType

This structure contains information about one item of the bill of material. The structure is defined in Table 16.

**Table 16 – BOMInformationDataType Structure**

Name	Type	Description
BOMInformationDataType	structure	
Identification	OutputInformationDataType	Identification of the output.
ComponentInformation	BOMComponentInformationDataType[]	Contains information about components.

Its representation in the *AddressSpace* is defined in Table 17.

**Table 17 – BOMInformationDataType Definition**

Attribute	Value				
BrowseName	BOMInformationDataType				
IsAbstract	False				
References	Node Class	BrowseName	DataType	TypeDefinition	Other
Subtype of the 0:Structure defined in OPC 10000-5					
Conformance Units					
Machinery Job Management Result BOM					

## 9.7 OutputPerformanceInfoDataType

This structure contains an individual performance information. The structure is defined in Table 18.

**Table 18 – OutputPerformanceInfoDataType Structure**

Name	Type	Description	Optional
OutputPerformanceInfoDataType	structure		
Identification	OutputInformationDataType	Identification of the output.	False
StartTime	0:DateTime	Output of first item from order.	True
EndTime	0:DateTime	Output of last item from order.	True
Parameters	2:ISA95ParameterDataType[]	Parameters specific to the performance like pressure or temperature.	False

Its representation in the *AddressSpace* is defined in Table 19.

**Table 19 – OutputPerformanceInfoDataType Definition**

Attribute	Value				
BrowseName	OutputPerformanceInfoDataType				
IsAbstract	False				
References	Node Class	BrowseName	DataType	TypeDefinition	Other
Subtype of the 0:Structure defined in OPC 10000-5					
Conformance Units					
Machinery Job Management Result PerformanceInfo					

## 9.8 ProcessIrregularity

This enumeration *ProcessIrregularity* describes if a process irregularity took place. The enumeration is defined in Table 6.

**Table 20 – ProcessIrregularity Items**

Name	Value	Description
CapabilityUnavailable	0	The machine is not able to give a statement about process irregularities.
Detected	1	A process irregularity has been detected.
NotDetected	2	There was no process irregularity detected.
NotYetDetermined	3	A statement about the process irregularity is to be expected.

Its representation in the *AddressSpace* is defined in Table 7.

**Table 21 – ProcessIrregularity Definition**

Attribute	Value				
BrowseName	ProcessIrregularity				
IsAbstract	False				
References	Node Class	BrowseName	Data Type	Type Definition	Other
Subtype of the Enumeration type defined in OPC 10000-5					
0:HasProperty	Variable	0:EnumValues	0:EnumValueType[]	0:PropertyType	
<b>Conformance Units</b>					
Machinery Job Management Result ProcessIrregularity					

## 10 Profiles and ConformanceUnits

### 10.1 Conformance Units

This chapter defines the corresponding *Conformance Units* for the OPC UA Information Model for Machinery – Job Management.

**Table 22 – Conformance Units for Machinery – Job Management**

Category	Title	Description
Server	Machinery Job Management Base	Supports the JobManagementType and at least one instance used as AddIn under an MachineryBuildingBlocks folder.
Server	Machinery Job Management Planned Base	Supports for a job order (input) the following predefined parameters: RunsPlanned for JobOrderParameters; Identification for MaterialRequirements Supports the DataTypes OutputInformationDataType and OutputInfoType If a predefined parameter of this specification is used, it is used as defined.
Server	Machinery Job Management Planned JobName	Supports for a job order (input) the following predefined parameters: JobName for JobOrderParameters
Server	Machinery Job Management Planned OrderNumbers	Supports for a job order (input) the following predefined parameters: OrderNumbers for JobOrderParameters
Server	Machinery Job Management Planned Customers	Supports for a job order (input) the following predefined parameters: Customers for JobOrderParameters
Server	Machinery Job Management Planned CustomerOrderNumbers	Supports for a job order (input) the following predefined parameters: CustomerOrderNumbers for JobOrderParameters
Server	Machinery Job Management Planned PlannedProductionTime	Supports for a job order (input) the following predefined parameters: PlannedProductionTime for JobOrderParameters
Server	Machinery Job Management Planned PlannedSetupTime	Supports for a job order (input) the following predefined parameters: PlannedSetupTime for JobOrderParameters
Server	Machinery Job Management Planned PlannedTimePerRun	Supports for a job order (input) the following predefined parameters: PlannedTimePerRun for JobOrderParameters
Server	Machinery Job Management Planned PlannedQuantityPerRun	Supports for a job order (input) the following predefined parameters: PlannedQuantityPerRun for JobOrderParameters

Server	Machinery Job Management Planned PlannedOrderQuantity	Supports for a job order (input) the following predefined parameters: PlannedOrderQuantity for JobOrderParameters
Server	Machinery Job Management Planned PlannedDuration	Supports for a job order (input) the following predefined parameters: PlannedDuration for JobOrderParameters
Server	Machinery Job Management Planned JobAnnotation	Supports for a job order (input) the following predefined parameters: JobAnnotation for JobOrderParameters
Server	Machinery Job Management Planned ComponentName	Supports for a job order (input) the following predefined parameters: ComponentName for EquipmentRequirements
Server	Machinery Job Management Planned DrawingNumber	Supports for a job order (input) the following predefined parameters: DrawingNumber for MaterialRequirements
Server	Machinery Job Management Planned DrawingVersionNumber	Supports for a job order (input) the following predefined parameters: DrawingVersionNumber for MaterialRequirements
Server	Machinery Job Management Planned Location	Supports for a job order (input) the following predefined parameters: Location for MaterialRequirements
Server	Machinery Job Management Planned RelatedContainer	Supports for a job order (input) the following predefined parameters: RelatedContainer for MaterialRequirements
Server	Machinery Job Management Planned ExecutionMode	Supports for a job order (input) the following predefined parameters: JobExecutionMode for JobOrderParameters; Supports the DataType JobExecutionMode
Server	Machinery Job Management ReasonForStateChange	Supports for a job order (input) and job result (actual) the following predefined parameters: ReasonForStateChange for JobOrderParameters
Server	Machinery Job Management Result Base	Supports for a job result (actual) the following predefined parameters: Identification for MaterialActuals Supports the DataTypes OutputInformationDataType and OutputInfoType If a predefined parameter of this specification is used, it is used as defined.
Server	Machinery Job Management Result JobName	Supports for a job result (actual) the following predefined parameters: JobName for JobResponseData
Server	Machinery Job Management Result OrderNumbers	Supports for a job result (actual) the following predefined parameters: OrderNumbers for JobResponseData
Server	Machinery Job Management Result Customers	Supports for a job result (actual) the following predefined parameters: Customers for JobResponseData
Server	Machinery Job Management Result CustomerOrderNumbers	Supports for a job result (actual) the following predefined parameters: CustomerOrderNumbers for JobResponseData
Server	Machinery Job Management Result RunsCompleted	Supports for a job result (actual) the following predefined parameters: RunsCompleted for JobResponseData
Server	Machinery Job Management Result RunsStarted	Supports for a job result (actual) the following predefined parameters: RunsStarted for JobResponseData
Server	Machinery Job Management Result ActualQuantityCurrentRun	Supports for a job result (actual) the following predefined parameters: ActualQuantityCurrentRun for JobResponseData
Server	Machinery Job Management Result ActualProductionTime	Supports for a job result (actual) the following predefined parameters: ActualProductionTime for JobResponseData
Server	Machinery Job Management Result ActualUnitSetupTime	Supports for a job result (actual) the following predefined parameters: ActualUnitSetupTime for JobResponseData
Server	Machinery Job Management Result ActualUnitDelayTime	Supports for a job result (actual) the following predefined parameters: ActualUnitDelayTime for JobResponseData
Server	Machinery Job Management Result ProducedQuantity	Supports for a job result (actual) the following predefined parameters: ProducedQuantity for JobResponseData
Server	Machinery Job Management Result GoodQuantity	Supports for a job result (actual) the following predefined parameters: GoodQuantity for JobResponseData
Server	Machinery Job Management Result EstimatedRemainingTime	Supports for a job result (actual) the following predefined parameters: EstimatedRemainingTime for JobResponseData
Server	Machinery Job Management Result ComponentName	Supports for a job result (actual) the following predefined parameters: ComponentName for EquipmentActuals
Server	Machinery Job Management Result DrawingNumber	Supports for a job result (actual) the following predefined parameters: DrawingNumber for MaterialActuals
Server	Machinery Job Management Result DrawingVersionNumber	Supports for a job result (actual) the following predefined parameters: DrawingVersionNumber for MaterialActuals
Server	Machinery Job Management Result Location	Supports for a job result (actual) the following predefined parameters: Location for MaterialActuals
Server	Machinery Job Management Result RelatedContainer	Supports for a job result (actual) the following predefined parameters: RelatedContainer for MaterialActuals
Server	Machinery Job Management Result StartTime	Supports for a job result (actual) the following predefined parameters: StartTime for MaterialActuals

Server	Machinery Job Management Result EndTime	Supports for a job result (actual) the following predefined parameters: EndTime for MaterialActuals
Server	Machinery Job Management Result ProcessIrregularity	Supports for a job result (actual) the following predefined parameters: ProcessIrregularity for MaterialActuals Supports the DataType ProcessIrregularity
Server	Machinery Job Management Result BOM	Supports for a job result (actual) the following predefined parameters: AsBuiltBOM for JobResponseData Supports the DataTypes BOMInformationDataType, BOMComponentInformationDataType
Server	Machinery Job Management Result PerformanceInfo	Supports for a job result (actual) the following predefined parameters: OutputPerformanceInfo for JobResponseData Supports the DataType OutputPerformanceInfoDataType
Server	Machinery Job Management Result ExecutionMode	Supports for a job result (actual) the following predefined parameters: JobExecutionMode for JobResponseData; Supports the DataType JobExecutionMode
Server	Machinery Job Management Result JobResult	Supports for a job result (actual) the following predefined parameters: JobResult for JobResponseData; Supports the DataType JobResult
Server	Machinery Job Management Minimum String Length	Supports for all received JobOrderParameters in all its internal data structures defined in OPC 10031-4 or this specification a string length (in chars) of at least 64. That includes the length of the string part of any LocalizedText.
Server	Machinery Job Management Localization	Supports for all received JobOrderParameters in all its internal data structures using arrays of LocalizedText to provide text in different languages at least two entries.

## 10.2 Profiles

### 10.2.1 Profile list

Table 23 lists all Profiles defined in this document and defines their URIs.

**Table 23 – Profile URIs for Machinery – Job Management**

Profile	URI
Machinery Job Management Base Server Facet	<a href="http://opcfoundation.org/UA-Profile/Machinery/Jobs/Server/Base">http://opcfoundation.org/UA-Profile/Machinery/Jobs/Server/Base</a>

### 10.2.2 Server Facets

#### 10.2.2.1 Overview

The following sections specify the *Facets* available for *Servers* that implement the Machinery – Job Management companion specification. Each section defines and describes a *Facet* or *Profile*.

#### 10.2.2.2 Machinery Job Management Base Server Facet

Table 24 defines a *Facet* with the base functionality of job management. Note, that it does not include the eventing mechanism. This mechanism is already defined by 2:ISA-95 Job Order Status Events Server Facet.

**Table 24 – Machinery Job Management Base Server Facet**

Group	Conformance Unit / Profile Title	Mandatory / Optional
Profile	2:ISA-95 Job Order Receiver Server V2 Facet	
Profile	2:ISA-95 Job Response Provider Server V2 Facet	
Machinery/Jobs	Machinery Job Management Base	M
Machinery/Jobs	Machinery Job Management Minimum String Length	M
Machinery/Jobs	Machinery Job Management Planned Base	O
Machinery/Jobs	Machinery Job Management Planned ExecutionMode	O
Machinery/Jobs	Machinery Job Management ReasonForStateChange	O
Machinery/Jobs	Machinery Job Management Result Base	O
Machinery/Jobs	Machinery Job Management Result BOM	O
Machinery/Jobs	Machinery Job Management Result PerformanceInfo	O
Machinery/Jobs	Machinery Job Management Result ExecutionMode	O
Machinery/Jobs	Machinery Job Management Result JobResult	O
Machinery/Jobs	Machinery Job Management Localization	O
Machinery/Jobs	Machinery Job Management Planned JobName	O
Machinery/Jobs	Machinery Job Management Planned OrderNumbers	O
Machinery/Jobs	Machinery Job Management Planned Customers	O
Machinery/Jobs	Machinery Job Management Planned CustomerOrderNumbers	O
Machinery/Jobs	Machinery Job Management Planned PlannedProductionTime	O
Machinery/Jobs	Machinery Job Management Planned PlannedSetupTime	O
Machinery/Jobs	Machinery Job Management Planned PlannedTimePerRun	O
Machinery/Jobs	Machinery Job Management Planned PlannedQuantityPerRun	O
Machinery/Jobs	Machinery Job Management Planned PlannedOrderQuantity	O
Machinery/Jobs	Machinery Job Management Planned PlannedDuration	O
Machinery/Jobs	Machinery Job Management Planned JobAnnotation	O
Machinery/Jobs	Machinery Job Management Planned ComponentName	O
Machinery/Jobs	Machinery Job Management Planned DrawingNumber	O
Machinery/Jobs	Machinery Job Management Planned DrawingVersionNumber	O
Machinery/Jobs	Machinery Job Management Planned Location	O
Machinery/Jobs	Machinery Job Management Planned RelatedContainer	O
Machinery/Jobs	Machinery Job Management Result JobName	O
Machinery/Jobs	Machinery Job Management Result OrderNumbers	O
Machinery/Jobs	Machinery Job Management Result Customers	O
Machinery/Jobs	Machinery Job Management Result CustomerOrderNumbers	O
Machinery/Jobs	Machinery Job Management Result RunsCompleted	O
Machinery/Jobs	Machinery Job Management Result RunsStarted	O
Machinery/Jobs	Machinery Job Management Result ActualQuantityCurrentRun	O
Machinery/Jobs	Machinery Job Management Result ActualProductionTime	O
Machinery/Jobs	Machinery Job Management Result ActualUnitSetupTime	O
Machinery/Jobs	Machinery Job Management Result ActualUnitDelayTime	O
Machinery/Jobs	Machinery Job Management Result ProducedQuantity	O
Machinery/Jobs	Machinery Job Management Result GoodQuantity	O
Machinery/Jobs	Machinery Job Management Result EstimatedRemainingTime	O
Machinery/Jobs	Machinery Job Management Result ComponentName	O
Machinery/Jobs	Machinery Job Management Result DrawingNumber	O
Machinery/Jobs	Machinery Job Management Result DrawingVersionNumber	O
Machinery/Jobs	Machinery Job Management Result Location	O
Machinery/Jobs	Machinery Job Management Result RelatedContainer	O
Machinery/Jobs	Machinery Job Management Result StartTime	O
Machinery/Jobs	Machinery Job Management Result EndTime	O
Machinery/Jobs	Machinery Job Management Result ProcessIrregularity	O

### 10.2.3 Client Facets

#### 10.2.3.1 Overview

This specification does not define any client facets or profiles. Note that OPC 10031-4 already defines client facets that can be used.



## 11 Namespaces

### 11.1 Namespace Metadata

Table 25 defines the namespace metadata for this document. 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 OPC 10000-5 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 OPC 10000-5.

The version information is also provided as part of the *ModelTableEntry* in the *UANodeSet XML* file. The *UANodeSet XML* schema is defined in OPC 10000-6.

**Table 25 – NamespaceMetadata Object for this Document**

Attribute	Value	
BrowseName	http://opcfoundation.org/UA/Machinery/Jobs/	
Property	DataType	Value
NamespaceUri	String	http://opcfoundation.org/UA/Machinery/Jobs/
NamespaceVersion	String	1.0.1
NamespacePublicationDate	DateTime	2024-05-01
IsNamespaceSubset	Boolean	False
StaticNodeIdsTypes	IdType []	0
StaticNumericNodeIdsRange	NumericRange []	
StaticStringNodeIdsPattern	String	

Note: The *IsNamespaceSubset Property* is set to False as the *UANodeSet XML* file contains the complete Namespace. *Servers* only exposing a subset of the Namespace need to change the value to True.

### 11.2 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 26 provides a list of namespaces that may be used in a Machinery – Job Management OPC UA *Server*.

**Table 26 – Namespaces used in a Machinery Job Management Server**

NamespaceURI	Description
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.
Local Server URI	Namespace for nodes defined in the local server. This namespace shall have namespace index 1.
http://opcfoundation.org/UA/ISA95-JOBCONTROL_V2/	Namespace for <i>NodeIds</i> and <i>BrowseNames</i> defined in OPC 10031-4. The namespace index is <i>Server</i> specific.
http://opcfoundation.org/UA/Machinery/	Namespace for <i>NodeIds</i> and <i>BrowseNames</i> defined in OPC UA for Machinery (OPC 40001-1). The namespace index is <i>Server</i> specific.
http://opcfoundation.org/UA/Machinery/Jobs/	Namespace for <i>NodeIds</i> and <i>BrowseNames</i> defined in this document. The namespace index is <i>Server</i> specific.
Vendor specific types	A <i>Server</i> may provide vendor-specific types like types derived from <i>ObjectTypes</i> defined in this document in a vendor-specific namespace.
Vendor specific instances	A <i>Server</i> provides vendor-specific instances of the standard types or vendor-specific instances of vendor-specific types in a vendor-specific namespace. It is recommended to separate vendor specific types and vendor specific instances into two or more namespaces.

Table 27 provides a list of namespaces and their indices used for *BrowseNames* in this document. The default namespace of this document is not listed since all *BrowseNames* without prefix use this default namespace.

**Table 27 – Namespaces used in this document**

NamespaceURI	Namespace Index	Example
http://opcfoundation.org/UA/	0	0:EngineeringUnits
http://opcfoundation.org/UA/ISA95-JOBCONTROL_V2/	2	2:ISA95ParameterDataType
http://opcfoundation.org/UA/Machinery/	3	3:MachineryBuildingBlocks

## **Annex A (normative)**

### **Machinery – Job Management Namespace and mappings**

#### **A.1 NodeSet and supplementary files for Machinery – Job Management Information Model**

The Machinery – Job Management *Information Model* is identified by the following URI:

<http://opcfoundation.org/UA/Machinery/Jobs/>

Documentation for the NamespaceUri can be found [here](#).

The *NodeSet* associated with this version of specification can be found here:

<https://reference.opcfoundation.org/nodesets/?u=http://opcfoundation.org/UA/Machinery/Jobs/&v=1.0.0&i=1>

The *NodeSet* associated with the latest version of the specification can be found here:

<https://reference.opcfoundation.org/nodesets/?u=http://opcfoundation.org/UA/Machinery/Jobs/&i=1>

Supplementary files for the Machinery – Job Management *Information Model* can be found here:

<https://reference.opcfoundation.org/nodesets/?u=http://opcfoundation.org/UA/Machinery/Jobs/&v=1.0.0&i=2>

The files associated with the latest version of the specification can be found here:

<https://reference.opcfoundation.org/nodesets/?u=http://opcfoundation.org/UA/Machinery/Jobs/&i=2>

---

## Annex B (informative)

### Examples

#### B.1 Overview

This annex gives examples on how the model can be used.

#### B.2 Example of a job order producing glass

In Table 28, an example JobResponseData is provided.

**Table 28 – Example of a job order result (Glass)**

Field of JobResponseData	Value
JobResponseID	2
JobOrderID	77773
StartTime	17.10.2022 17:17
EndTime	17.10.2022 17:19
JobState	COMPLETED
MaterialActuals[]	
Entry1	
MaterialUse	Material produced
Quantity	1
EngineeringUnits	pcs
Properties[]	
Identification	
ItemNumber	ISO_4-16-4_LE Solar
OutputInfo	SerialNumbers
OrderNumber	Auftr_ISO_123
LotNumber	AuftragsPOS_02
SerialNumber	ISO_12345
StartTime	17.10.2022 17:17
EndTime	17.10.2022 17:19
Param_1234	1450 [Width, mm]
Param_1235	1950 [Length, mm]
Param_1236	24,02 [Thickness, mm]
Location	ISO_Line_3_Exit or Rack_1234
JobResponseData[]	
AsBuiltBOM[]	
Entry1	
Identification	

<b>ItemNumber</b>	ISO_4-16-4_LE Solar
<b>OutputInfo</b>	SerialNumber
<b>SerialNumber</b>	ISO_12345
<b>ComponentInformation[]</b>	
<b>Entry1</b>	
<b>Identification</b>	
<b>ItemNumber</b>	1004 (4mm Float)
<b>OutputInfo</b>	SerialNumber
<b>SerialNumber</b>	Pane_123456
<b>Quantity</b>	1
<b>EngineeringUnits</b>	pcs
<b>Entry2</b>	
<b>Identification</b>	
<b>ItemNumber</b>	Alu_16mm
<b>OutputInfo</b>	SerialNumber
<b>SerialNumber</b>	Frame_3456456
<b>Quantity</b>	1
<b>EngineeringUnits</b>	pcs
<b>Entry3</b>	
<b>Identification</b>	
<b>ItemNumber</b>	3004 (4mm LowE)
<b>OutputInfo</b>	SerialNumber
<b>SerialNumber</b>	Pane_256663
<b>Quantity</b>	1
<b>EngineeringUnits</b>	pcs
<b>Entry4</b>	
<b>Identification</b>	
<b>ItemNumber</b>	Diacol Base Component
<b>OutputInfo</b>	SerialNumber
<b>SerialNumber</b>	DiaB1_256663 (Barrel Nbr)
<b>Quantity</b>	0,025
<b>EngineeringUnits</b>	dm3
<b>Entry5</b>	
<b>Identification</b>	
<b>ItemNumber</b>	Diacol Hardener Component
<b>OutputInfo</b>	SerialNumber
<b>SerialNumber</b>	DiaH1_2563 (Barrel Nbr)
<b>Quantity</b>	0,005

<b>EngineeringUnits</b>	dm3
<b>Entry6</b>	
<b>Identification</b>	
<b>ItemNumber</b>	Argon
<b>OutputInfo</b>	- (only ItemNumber)
<b>Quantity</b>	5,3
<b>EngineeringUnits</b>	dm3
<b>OutputPerformanceInfo[]</b>	
<b>Entry1</b>	
<b>Identification</b>	
<b>ItemNumber</b>	ISO_4-16-4_LE Solar
<b>OutputInfo</b>	SerialNumber
<b>SerialNumber</b>	ISO_12345
<b>StartTime</b>	17.10.2022 17:17
<b>EndTime</b>	17.10.2022 17:19
<b>Parameters[]</b>	
<b>Entry1</b>	
<b>ID</b>	ISO_Value_1234
<b>Description</b>	Gas Filling Level
<b>Value</b>	95
<b>EngineeringUnits</b>	%
<b>Entry2</b>	
<b>ID</b>	ISO_Value_1100
<b>Description</b>	Washing Water Temp
<b>Value</b>	62
<b>EngineeringUnits</b>	%
<b>Entry3</b>	
<b>ID</b>	ISO_Value_1255
<b>Description</b>	Corner Position Pressing
<b>Value</b>	20,4; 20,6 ;20,5; 20,3
<b>EngineeringUnits</b>	mm
<b>Entry4</b>	
<b>ID</b>	ISO_Value_1001
<b>Description</b>	Override average
<b>Value</b>	95,2
<b>EngineeringUnits</b>	%

### B.3 Example of a job breaking a tree down into a shelf floor, table legs and chips

First, the job is sent from the MES system to the woodworking machine. This can be either done by calling the `2:Store` or the `2:StoreAndStart Method` on the `2:JobOrderReceiverObject` defined in OPC 10031-4. In both cases, the job order information is transferred as instance of `ISA95JobOrderDataType`. An example of such a data structure is given in Table 29. In Figure 3, the planned job order is sketched. From a tree, 1 shelf floor and 4 table legs should be created. In addition a bag of chips is created. In addition a bag of chips is created.

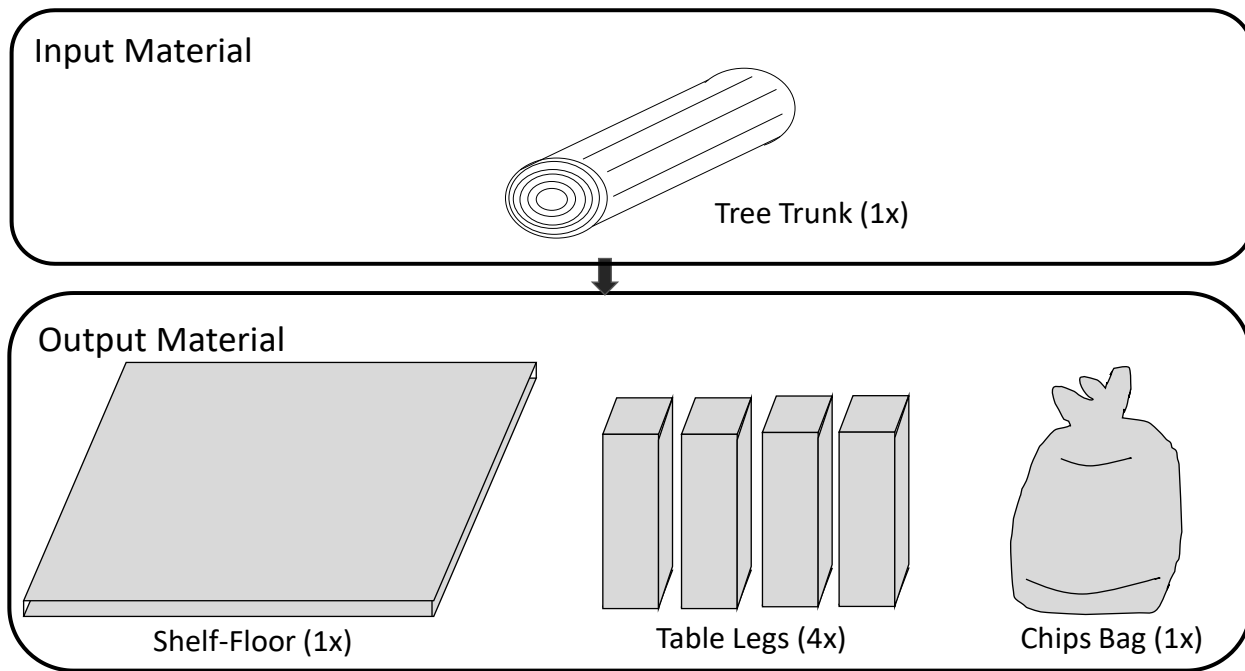


Figure 3 – Example of a planned job order

Table 29 – Example of a job order

Field of ISA95JobOrderDataType	Value
JobOrderID	12345
OrderNumbers	Order_Forest_Utilize_01
StartTime (planned)	27.01.2023 10:17
EndTime (planned)	27.01.2023 10:19
MaterialRequirements[]	
Entry1 (Tree as planned Input)	
MaterialUse	Material consumed
Quantity	1
EngineeringUnits	pcs
Properties[]	
Identification	
ItemNumber	TreeTrunk
OutputInfo	- (only ItemNumber)
Param_771	200 [diameter, cm]

<b>Location</b>	Cutting_line_input_1
<b>Entry2 (shelf floor as planned output)</b>	
<b>MaterialUse</b>	Material produced
<b>Quantity</b>	1
<b>EngineeringUnits</b>	pcs
<b>Properties[]</b>	
<b>Identification</b>	
<b>ItemNumber</b>	Shelf_Floor_0010
<b>OutputInfo</b>	- (only ItemNumber)
<b>OrderNumber</b>	Order_Forest_Utilize_01
<b>LotNumber</b>	Forest_Spruce_ShelfFloor_1245
<b>SerialNumber</b>	FSSF_1234568
<b>Param_1234</b>	2500 [Width, mm]
<b>Param_1235</b>	1500 [Length, mm]
<b>Param_1236</b>	15,8 [Thickness, mm]
<b>Location</b>	Cutting_line_output_1
<b>Entry3 (table leg (4x) as planned output)</b>	
<b>MaterialUse</b>	Material produced
<b>Quantity</b>	4
<b>EngineeringUnits</b>	pcs
<b>Properties[]</b>	
<b>Identification</b>	
<b>ItemNumber</b>	Table_Leg_012
<b>OutputInfo</b>	- (only ItemNumber)
<b>OrderNumber</b>	Order_Forest_Utilize_01
<b>LotNumber</b>	Forest_Spruce_TableLeg_124
<b>StartTime</b>	27.01.2023 10:17
<b>EndTime</b>	27.01.2023 10:19
<b>Param_1234</b>	125 [Width, mm]
<b>Param_1235</b>	855 [Length, mm]
<b>Param_1236</b>	125 [Thickness, mm]
<b>Location</b>	Cutting_line_output_2
<b>Entry4 (chips planned output)</b>	
<b>MaterialUse</b>	Material produced
<b>Quantity</b>	1,75
<b>EngineeringUnits</b>	m^3
<b>Properties[]</b>	
<b>Identification</b>	



<b>ItemNumber</b>	SpruceChips_012
<b>OutputInfo</b>	- (only ItemNumber)
<b>Param_333</b>	Spruce [Woodtype]
<b>Location</b>	BagFiller_output_1

After the job order is received, it needs to be in state “AllowedToStart” before it can be executed. In case of the 2:StoreAndStart Method, it is already in the state, otherwise the 2:Start Method needs to be called. Once it is in the “AllowedToStart” state, the woodworking machine is internally starting the job order once it is ready (and depending on priorities of other job orders which are “AllowedToStart”. As soon as the job order has started (state of the job order is “Running”), the machine can provide results of the execution of the job order. The job order may get interrupted, or even aborted. Or it completes by switching into the “Ended” state. In Table 30, the final job order results, provided in an instance of 2:ISA95JobResponseDataType, are shown.

**Table 30 – Example of a job order result**

Field of JobResponseData	Value
<b>JobResponseID</b>	4
<b>JobOrderID</b>	12345
<b>StartTime</b>	27.01.2023 10:20
<b>EndTime</b>	27.01.2023 10:25
<b>JobState</b>	COMPLETED
<b>MaterialActuals[]</b>	
<b>Entry1 (shelf floor as actual output)</b>	
<b>MaterialUse</b>	Material produced
<b>Quantity</b>	1
<b>EngineeringUnits</b>	pcs
<b>Properties[]</b>	
<b>Identification</b>	
<b>ItemNumber</b>	Shelf_Floor_0010
<b>OutputInfo</b>	- (only ItemNumber)
<b>OrderNumber</b>	Order_Forest_Utilize_01
<b>LotNumber</b>	Forest_Spruce_ShelfFloor_1245
<b>SerialNumber</b>	FSSF_1234568
<b>StartTime</b>	27.01.2023 10:20
<b>EndTime</b>	27.01.2023 10:22
<b>Param_1234</b>	2500 [Width, mm]
<b>Param_1235</b>	1500 [Length, mm]
<b>Param_1236</b>	15,8 [Thickness, mm]
<b>Location</b>	Cutting_line_output_1
<b>Entry2 (table leg (4x) as actual output)</b>	
<b>MaterialUse</b>	Material produced
<b>Quantity</b>	4

<b>EngineeringUnits</b>	pcs
<b>Properties[]</b>	
<b>Identification</b>	
<b>ItemNumber</b>	Table_Leg_012
<b>OutputInfo</b>	LotNumber
<b>OrderNumber</b>	Order_Forest_Utilize_01
<b>LotNumber</b>	Forest_Spruce_TableLeg_124
<b>StartTime</b>	27.01.2023 10:22
<b>EndTime</b>	27.01.2023 10:25
<b>Param_1234</b>	125 [Width, mm]
<b>Param_1235</b>	855 [Length, mm]
<b>Param_1236</b>	125 [Thickness, mm]
<b>Location</b>	Cutting_line_output_2
<b>Entry3 (chips as actual output)</b>	
<b>MaterialUse</b>	Material produced
<b>Quantity</b>	1,75
<b>EngineeringUnits</b>	m^3
<b>Properties[]</b>	
<b>Identification</b>	
<b>ItemNumber</b>	SpruceChips_012
<b>OutputInfo</b>	- (only ItemNumber)
<b>SerialNumber</b>	Bag_12345
<b>StartTime</b>	27.01.2023 10:20
<b>EndTime</b>	27.01.2023 10:25
<b>Param_333</b>	Spruce [Woodtype]
<b>Location</b>	BagFiller_output_1
<b>JobResponseData[]</b>	
<b>AsBuiltBOM[]</b>	
<b>Entry1</b>	
<b>Identification</b>	
<b>ItemNumber</b>	Shelf_Floor_0010
<b>OutputInfo</b>	LotNumber
<b>LotNumber</b>	Forest_Spruce_ShelfFloor_1245
<b>ComponentInformation[]</b>	
<b>Entry1</b>	
<b>Identification</b>	
<b>ItemNumber</b>	TreeTrunk
<b>OutputInfo</b>	SerialNumber

<b>SerialNumber</b>	TreeTrunk_Sepp123
<b>Quantity</b>	1
<b>EngineeringUnits</b>	pcs
<b>Entry2</b>	
<b>Identification</b>	
<b>ItemNumber</b>	Table_Leg_012
<b>OutputInfo</b>	LotNumber
<b>LotNumber</b>	Forest_Spruce_TableLeg_124
<b>ComponentInformation[]</b>	
<b>Entry1</b>	
<b>Identification</b>	
<b>ItemNumber</b>	TreeTrunk
<b>OutputInfo</b>	SerialNumber
<b>SerialNumber</b>	TreeTrunk_Sepp123
<b>Quantity</b>	1
<b>EngineeringUnits</b>	pcs