



OpCon
MES

User Manual

OpConXml PartIII



BOSCH
Invented for life

1 Change Index

Version	Change	Author	Date
V2.04	<ul style="list-style-type: none"> revised a few parts of the documentation, which differed from the german documentation (master) 	Denis Tahsinoglou	12.08.2015
V02.03	<ul style="list-style-type: none"> Review in accordance with requirements specification 	Eiberle	23.02.2014
V02.02	<ul style="list-style-type: none"> <code>plcToolChangeStarted</code>: Answer now optionally includes the tool family. Data type of <code>checkType</code> and result of int (DINT) changed to short (INT). Optional reporting of NIO strokes of tools at the end of processing. Changeover material: type variant is optionally returned for changeover request. All status variables are of type short (INT). Data types converted to PLC data types. Processing release for add-on components. 	Eiberle	20.11.2009
V02.01	<ul style="list-style-type: none"> Cancel processing modified. Examples of processing release revised. 	Eiberle	16.04.2009
V02.00 (version 01)	<ul style="list-style-type: none"> <code>WorkPart.NextProcessNo</code> added. 	Eiberle	25.03.2009
V02.00	<ul style="list-style-type: none"> Comments of the AE-MES team entered. 	Eiberle	13.03.2009
V02.00 (beta 2 version 02)	<ul style="list-style-type: none"> <code>contentType</code> added in examples. 	Eiberle	06.03.2009
V01.01	<ul style="list-style-type: none"> New document layout Examples revised. 	Eiberle	26.02.2007
V01.00	<ul style="list-style-type: none"> <code>dataUploadRequired</code> and <code>partProcessed</code> enhanced for LLD. 	Eiberle	04.08.2006

Table of Contents

1 Change Index	2
2 About this Documentation	4
2.1 Icons used in this manual	4
2.2 Conventions	4
2.3 Intended Use.....	5
2.4 Safety Regulations and Warnings	6
2.5 Legal Framework and Copyright	7
3 Introduction	8
3.1 Brief Description.....	8
3.2 Examples	8
4 Part 3: Specification AE	9
4.1 BDE/MDE	9
4.1.1 Changing the Setup Status	9
4.1.2 Reporting Station Errors	9
4.1.3 Current Setup Status (Optional)	10
4.2 Setup Data.....	11
4.2.1 Changeover to Product Type.....	11
4.2.2 Setting Up the Tool	21
4.2.3 Changing the Material	24
4.3 Production Data	31
4.3.1 Processing Release	31
4.3.2 Starting to Process the Workpiece.....	53
4.3.3 Canceling Processing.....	54
4.3.4 Processing Complete	54
4.3.5 Changing the Workpiece.....	78
4.4 Material data	78
4.4.1 Material master data query.....	78
4.4.2 Manual Material Message	81
4.5 Other Actions	82
4.5.1 User Login.....	82
4.5.2 Logging a User Off.....	83
4.5.3 Transferring Data for Customer Labels	83
4.5.4 Posting and Querying Material Components	84
5 Appendix A: Setpoint values and constants	87
5.1 Codes for Material Locks	87
6 Index	88

2 About this Documentation

2.1 Icons used in this manual

Standard formatting is used for all content in the Opcon documentation as follows:

- To indicate hazards:

⚡ **This is a hazard warning**

- For warnings:

⚠ *This is a warning*

- For hints:

💡 *This is a hint*

- For instructions and their consequences:

▶ This is an instruction.

This is the consequence of the action.

- When using menu texts from screenshots:

▶ In the **File** menu, click on **Open**.

- When using examples of code:

```
Dies ist ein Code-Beispiel  
DINT Test
```

For file paths:

\\fe16416\opcon\$

2.2 Conventions

In this document the following conventions are used for better readability.

Fonts

Monospace Monospace For excerpts from configuration files, source code or directories.

Bold All words in **bold** signify key terms.

Italics Words that are introduced for the first time are shown in *italics*.

"Quotation marks" Definitions are enclosed in quotation marks.

Notation

. Delimiters between XML elements.

@ Separator between XML elements and XML attributes.

The following data types are used in this document:

Data type	Description
uint	32-bit integer, unsigned uint Corresponds to the data type UDINT within IndraLogic.


int	32-bit integer, signed int Corresponds to the data type <code>DINT</code> within IndraLogic.
string	String, maximum length 80 characters. Corresponds to the data type <code>STRING</code> within IndraLogic.
string(x)	Character string, max. length x characters. Corresponds to the data type <code>STRING(x)</code> within IndraLogic.
bool	Bool can contain the letter symbols <code>true</code> or <code>false</code> . These are interpreted as case-sensitive.
real	8-byte real, decimal separator is a period, e-representation is permitted and preferred. <code>real</code> corresponds to the data type <code>REAL</code> within IndraLogic.
float	4-byte real, decimal separator is a period, e-representation is permitted and preferred. <code>real</code> Corresponds to the data type <code>REAL</code> within IndraLogic.

2.3 Intended Use

Our products are checked before delivery to ensure they are in safe working order. The products must only be used for their intended purpose. If they are used for any other purpose, this could result in personal injury or damage to property.


Before using our products, please ensure the following conditions are met in order to guarantee proper utilization:

- All persons who use our products must be familiar with the relevant safety requirements and the applicable national standards.
- Reverse engineering and reutilization or modification of our software is not permitted. Hardware must be left in its original state.
- Please ensure that the products are installed, set up, and operated in accordance with the relevant documentation.
- Read through and closely observe the operating and safety instructions and warnings. If the language provided is not sufficient, please ask us for the documentation in your particular language.
- Damaged or faulty products must not be used.
- Our products must only be used for industrial purposes.
- Components must not be commissioned until it has been verified that the machine or system in which they are installed complies with the country-specific provisions, safety regulations and standards for the application.


 *Robert Bosch GmbH provides no warranty and accepts no liability or obligation to provide compensation in the event of improper use of the products.*

2.4 Safety Regulations and Warnings

Safety regulations:

-  *Ensure the operators are familiar with the safety regulations before commissioning the product.*
- Always ensure electrical systems are operated by trained and qualified personnel.
 - The products are intended only for commercial use in industrial systems.
 - Refer to the product documentation for technical data and installation requirements if these systems are not standard installations.
 - Observe all warnings in the documentation!
 - Observe the safety requirements and regulations of the country in which the product is being operated.

Warnings:

-  *Before commissioning the product, always observe the following warnings:*
- Incorrect activation of actuators may result in dangerous movements. In our software products, incorrect activation may occur as a result of software errors or incorrectly set parameters in the software, especially in PLC function modules. These errors can occur at any time. The working area of the actuators must therefore be secured against any human intervention during both commissioning and operation.
 - The functionality of our products is constantly being expanded. In individual circumstances, it may therefore be the case that elements of the documentation do not fully match the latest product version. Certain sections may therefore differ from the latest version of the product. In this case, we would ask the user to notify us of any such discrepancies.

2.5 Legal Framework and Copyright

These instructions, including all diagrams and images contained herein, are protected by copyright. Any use of these instructions that deviates from the copyright provisions is not permitted. Any reproduction, translation into other languages, or electronic and photographic archiving and amendment requires written authorization from Robert Bosch GmbH, PA-ATMO1, Stuttgart. Any violations will incur claims for damages.

Robert Bosch GmbH reserves the right to make changes in accordance with technical advances. All rights pertaining to patents or protection of utility patents are reserved by Robert Bosch GmbH. Microsoft, Excel, Word, Outlook, and Windows are registered trademarks or brands of Microsoft Corporation in the USA and/or other countries. Other products or company names listed in this document may be protected trademarks of their respective owners, or may be subject to patent protection.

3 Introduction

3.1 Brief Description

OpCon XML is a standard for the exchange of data between production equipment and OpCon MES. The events specified by OpCon DirectDataLink are implemented as the corresponding XML telegrams.

OpCon XML has been developed to cover the widest possible platform of production facilities. The elements defined here do not necessarily have to be fully supported by all platforms. Simple clients such as PLCs, for example, can support only the core parts of the specification, while other more powerful clients, such as PC-based systems, implement the entire standard.

To enable OpCon XML to be used for the broadest possible range of applications, the standard was split into three areas:

Core specification

This section defines the parts that have to be implemented by all clients. The definitions contained in this section must be adhered to at all times.


Extended elements

This section contains the definitions that can be implemented by a client. The definitions specified here can be implemented depending on the client's capability.

If a client implements a definition, however, it must be completely implemented. Implementing only parts of a definition is not permissible.


Area-specific elements


Different production areas have different requirements for communication. In this section, you can enter the area-specific definition, e.g. specific definitions that are valid for the entire AE area.

 *This document contains part III of the specification, which is valid in the AE area.*

3.2 Examples

This specification contains a number of examples which illustrate the respective definitions. The examples are not binding. In particular, optional items may be used in the examples to highlight different aspects of the definition. It is therefore permitted that the implementations may differ from the XML telegrams shown in the examples, if the implementation does not violate any of the definitions described in this document.

 *The examples in this document are for illustration purposes only. They are not binding. In particular it cannot be assumed that the sequence of the XML elements and attributes contained in the examples is fixed. This may differ in the actual installations.*

 *The examples are normally demonstrated using `contentType=3`. However, this is only an option and is not a requirement.*

4 Part 3: Specification AE

For the AE area, a series of definitions have been made in order to unify the use of OpCon XML for the different processes.

These messages cover communication with the MES in the following areas:

- BDE/MDE
- Setup data
- Production data

4.1 BDE/MDE

BDE/MDE messages are always sent per functional unit in this specification. Therefore, the attributes `workPos` and `toolPos` of the `<header>` element do not have to be sent.

4.1.1 Changing the Setup Status

The change in the setup status is transferred via the station-side event `plcOperationModeChanged`. After receiving the message, MES **does not send an acknowledgment** to the station.

All of the attributes, including optional attributes, of the `<plcOperationModeChanged>` element must be specified. Where no meaningful values can be provided, the default values must be entered.

In addition to the data specified above, the following additional data can also be sent:

Information	Required	Data type	OpCon Xml
Piece shortage	Optional	BOOL	<code>body.partMissing</code>
On	Optional	BOOL	<code>body.powerOn</code>
Jam	Optional	BOOL	<code>body.jam</code>

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2313" version="2.2" eventName="plcOperationModeChanged"
    contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <plcOperationModeChanged typeNo="1111111111"
      typeVar=""
      shift="1"
      charge=""
      specPrgNo="0"
      operationMode="1"
      modeOn="true" />
  </event>
  <body/>
</root>
```

4.1.2 Reporting Station Errors


Station errors are sent to the MES by the event `plcError`. After receiving the message, MES **does not send an acknowledgment** to the station.

All of the attributes, including optional attributes, of the `<plcError>` element must be specified. Where no meaningful values can be provided, the default values must be entered. Otherwise, no additional data is sent to the MES.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2318" version="2.2" eventName="plcError" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <plcError typeNo="1111111111"
      typeVar=""
      errorNo="40743"
      errorText="No voltage"
      errorType="1"
      errorState="0"
      operationMode="1"
      modeOn="true"
      chainNo="12"/>
  </event>
  <body/>
</root>
```

4.1.3 Current Setup Status (Optional)

 This event is optional and should only be used if this is consistently required in the line. Projects that begin with OpCon Xml V2.3 should only use this message for projects if they have to support older versions of OpCon XML support.

The current setup status is reported to the MES by the event `dataUploadRequired`. So that an explicit mapping of the event to the current setup status can be introduced, the attribute `eventSwitch` must be set to 10. After receiving the message, the MES does not send an acknowledgment to the station.

In addition to the data specified above, the following additional data must also be sent:

Information	Required	Data type	OpCon Xml	Comments
Operation mode on	X	BOOL	item.modeOn	
Current operation mode	X	INT	item.operationMode	
Current type number	X	STRING(10)	item.typeNo	
Current type variant	Optional	STRING(10)	item.typeVar	
Current shift	Optional	DINT	item.shift	
Current batch number	Optional	STRING	item.batch	
Currently running special program	Optional	DINT	item.specPrgNo	Typically only exists in PLC clients.
Batch count NOK	Optional	DINT	item.batchCountNIO	
Batch count OK	Optional	DINT	item.batchCountIO	
Shift count NOK	Optional	DINT	item.shiftCountNIO	
Shift count OK	Optional	DINT	item.shiftCountIO	
Start time of current shift	Optional	STRING(12)	item.shiftStartTime	
End time of current shift	Optional	STRING(12)	item.shiftEndTime	
Change flag for station	Optional	BOOL	item.changeFlag	
Parts jam	Optional	BOOL	item.jam	
Part missing	Optional	BOOL	item.partMissing	

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2332" version="2.2" eventName="dataUploadRequired">
```

```

        eventSwitch="10" contentType="3">
        <location lineNo="1" statNo="30" statIdx="1" fuNo="1"
            processNo="90" processName="PlasmaCleaning" application="PLC"/>
    </header>
    <event>
        <dataUploadRequired/>
    </event>
    <body>
        <items>
            <item name="modeOn" value="true" dataType="11" />
            <item name="operationMode" value="1" dataType="2" />
            <item name="typeNo" value="1111111111" dataType="8" />
            <item name="typeVar" value="0815" dataType="8" />
            <item name="shift" value="2" dataType="3" />
            <item name="batch" value="CH990" dataType="8" />
            <item name="specPrgNo" value="0" dataType="3" />
            <item name="batchCountNIO" value="3" dataType="3" />
            <item name="batchCountIO" value="300" dataType="3" />
            <item name="shiftCountNIO" value="3" dataType="3" />
            <item name="shiftCountIO" value="300" dataType="3" />
            <item name="shiftStartTime" value="12:00:00" dataType="8" />
            <item name="shiftEndTime" value="18:00:00" dataType="8" />
            <item name="changeFlag" value="false" dataType="11" />
            <item name="jam" value="false" dataType="11" />
            <item name="partMissing" value="false" dataType="11" />
        </items>
    </body>
</root>

```

4.2 Setup Data

4.2.1 Changeover to Product Type

MES offers the possibility to connect stations via the OpCon XML interface to the changeover data administration (e.g. OpCon SetupControl). The following section describes the required communication sequences and data formats for using this integration.

Data Structures and Conventions

In the changeover request, only send the type part number and its variant. The reply contains the changeover parameters, whose format is described below depending on the category.

The changeover parameters are split into the following categories:

- Material requirements
- Tool requirements
- Program requirements
- Process requirements with limits
- Workpiece information (e.g. number per panel/carrier)

Which piece of changeover information is sent to the station is application-specific and is defined by a corresponding specification of the station interface.

With these basic structures, the standard default values are reported to the station. Depending on the station, 0..n of these default values may be contained in the response. The sequence of the individual items is arbitrary.



The station or process interfaces are specified internally in the AE and do not come under the scope of this document. More information on the available interfaces can be obtained from AE/MFI-CI.

💡 *Technically it is possible to send changeover parameters as structArray to the station. However, since this also requires a corresponding evaluation logic on the station side, this must be agreed upon before using the station programming.*

4.2.1.1 Material requirements

All requirements that affect the material to be used (e.g. components) are provided as `items`. The following information can be sent:

Information	Required	Data type	OpCon Xml	Comments
Material number	Mandatory	STRING	<code>Item.ComponentX.PartNo</code>	TTNR of the material that the station has to install. This number corresponds e.g. to the field 'P' (customer part number) of the Mat Label specification.
Type part number of the material (obsolete)	Mandatory	STRING(10)	<code>item.ComponentX.TypeNo</code>	Obsolete, do not use for new interfaces. Bosch TTNR of the material that the station has to install.
Revision status of the material	Optional	STRING	<code>Item.ComponentX.Revision</code>	Revision status of the material that the station has to install.
Type variant of the material (obsolete)	Optional	STRING(10)	<code>item.ComponentX.TypeVar</code>	Obsolete, do not use for new interfaces. Variant of the Bosch TTNR of the material that the station has to install.
Name of the component	Optional	STRING	<code>item.ComponentX.Name</code>	Unique name of the material throughout the station, which the station can use to interpret and assign the material.

Example:

In this section of an MES response, the setup parameters for two materials are transferred, which the station has to use for processing:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <!-- (...) -->
  <body>
    <items>
      <item name="Component1.PartNo" value="8936971155" dataType="8"/>
      <item name="Component1.Revision" value="" dataType="8"/>
      <item name="Component1.Name" value="Con1" dataType="8"/>
      <item name="Component2.PartNo" value="4560981278" dataType="8"/>
      <item name="Component2.Revision" value="" dataType="8"/>
      <item name="Component2.Name" value="Con2" dataType="8"/>
    </items>
  </body>
</root>
```

4.2.1.2 Tool requirements

All of the specifications that affect tools are provided as `items`. The following information can be sent:

Information	Required	Data type	OpCon Xml	Comments
Tool family	X	STRING(31)	<code>item.ToolX.Family</code>	Contains the ID of the tool family
Tool name	Optional	STRING	<code>item.ToolX.Name</code>	Unique name of the tool throughout the station, which the station can use to interpret and assign the tool.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<root>
<!-- (...) -->
  <body>
    <items>
      <item name="Tool1.Family" value="1002090" dataType="8"/>
      <item name="Tool1.Name" value="Mask" dataType="8"/>
      <item name="Tool2.Family" value="786644" dataType="8"/>
      <item name="Tool2.Name" value="Downholder" dataType="8"/>
    </items>
  </body>
</root>

```

4.2.1.3 Transport medium (workpiece carrier, magazine, container)

All specifications that affect the transport media are provided as `items`. The following information can be sent:

Information	Required	Data type	OpCon Xml	Comments
Family	X	STRING(31)	<code>item.CarrierX.Family</code>	Contains the ID of the transport medium family.
Name	Optional	STRING	<code>item.CarrierX.Name</code>	Unique name of the material throughout the station that the station can use to interpret and assign the transport medium.

Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
<!-- (...) -->
  <body>
    <items>
      <item name="Carrier1.Family" value="778823" dataType="8"/>
      <item name="Carrier1.Name" value="WTESP" dataType="8"/>
    </items>
  </body>
</root>

```

4.2.1.4 Programs

All specifications that affect the programs are provided as `items`. The following information can be sent:

Information	Required	Data type	OpCon Xml	Comments
Program number	Optional	DINT	<code>item.ProgX.Numbers</code>	Program number (if programs are managed by the client using numbers and not using the name).
Subsystem	Optional	STRING	<code>item.ProgX.SubSystem</code>	Subsystem
Program name	Optional	STRING(31)	<code>item.ProgX.Name</code>	Unique name of the material throughout the station, which the station can use to interpret and assign the program

Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
<!-- (...) -->
  <body>
    <items>
      <item name="Prog1.Name" value="PRG_MILL_1111111111" dataType="8"/>
      <item name="Prog1.SubSystem" value="BL922" dataType="8"/>
      <item name="Prog2.Name" value="NC808098" dataType="8"/>
      <item name="Prog2.SubSystem" value="XT2" dataType="8"/>
    </items>
  </body>

```

</root>

4.2.1.5 Process Specifications

All process specifications including limits are provided as `items`. The following information can be sent:

Information	Required	Data type	OpCon Xml	Comments
Name	Mandatory	STRING	<code>item.ProcessSetupX.Name</code>	Name of the specified process parameter
Specification	Mandatory	DINT, REAL, STRING	<code>item.ProcessSetupX.SetValue</code>	Setpoint value (set value)
Lower limit	Optional	DINT, REAL, STRING	<code>item.ProcessSetupX.LoLim</code>	Lower tolerance limit of the parameter
Upper limit	Optional	DINT, REAL, STRING	<code>item.ProcessSetupX.UpLim</code>	Upper tolerance limit of the parameter
Tolerance type	Optional	INT	<code>item.ProcessSetupX.CheckType</code>	Tolerance type of the parameter. The content describes how the default values are to be interpreted (absolute values, relative values, only checked against upper and lower limit).
Unit	Optional	STRING	<code>item.ProcessSetupX.Unit</code>	Measurement unit of the parameter (e.g. kg, Nm,...)

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <!-- (...) -->
  <body>
    <items>
      <item name="ProcessSetup1.Name" value="IntSample" dataType="8"/>
      <item name="ProcessSetup1.SetValue" value="5" dataType="3"/>
      <item name="ProcessSetup1.LoLim" value="2" dataType="3"/>
      <item name="ProcessSetup1.UpLim" value="8" dataType="3"/>
      <item name="ProcessSetup1.CheckType" value="5" dataType="2"/>
      <item name="ProcessSetup1.Unit" value="N" dataType="8"/>
      <item name="ProcessSetup2.Name" value="RealSample" dataType="8"/>
      <item name="ProcessSetup2.SetValue" value="5.00000000" dataType="5"/>
      <item name="ProcessSetup2.LoLim" value="2.00000000" dataType="5"/>
      <item name="ProcessSetup2.UpLim" value="8.00000000" dataType="5"/>
      <item name="ProcessSetup2.CheckType" value="5" dataType="2"/>
      <item name="ProcessSetup2.Unit" value="N" dataType="8"/>
      <item name="ProcessSetup3.Name" value="StringSample" dataType="8"/>
      <item name="ProcessSetup3.SetValue" value="ABC" dataType="8"/>
      <item name="ProcessSetup3.LoLim" value="" dataType="8"/>
      <item name="ProcessSetup3.UpLim" value="" dataType="8"/>
      <item name="ProcessSetup3.CheckType" value="3" dataType="2"/>
      <item name="ProcessSetup3.Unit" value="" dataType="8"/>
    </items>
  </body>
</root>
```

4.2.1.6 Workpiece Data

All requirements concerning the workpiece are provided in the element `workPart`. The following information can be sent:

Information	Required	Data type	OpCon Xml	Comments
Units per transport medium	Optional	DINT	item.WorkPart.PartsPerCarrier	Number of workpieces on the transport medium (e.g. workpiece carrier, boot, panel,...)
Family	Optional	STRING	item.WorkPart.Family	Identifier for the class of the workpiece
Units per row	Optional	DINT	item.WorkPart.PartsPerRow	Number of workpieces within a row on the transport medium
Units per column	Optional	DINT	item.WorkPart.PartsPerCol	Number of workpieces within a column on the transport medium
Processing code	Optional	INT	item.WorkPart.WorkingCode	Classification of the workpiece(s) (e.g. series part(s), calibration part(s), etc.)

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2311" version="2.3" eventName="plcChangeOverStarted"
    contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="0"/>
    <trace/>
  </event>
  <body>
    <items>
      <item name="WorkPart.PartsPerCarrier" value="6" dataType="3"/>
      <item name="WorkPart.Family" value="AA8823108" dataType="8"/>
      <item name="WorkPart.PartsPerRow" value="2" dataType="3"/>
      <item name="WorkPart.PartsPerCol" value="3" dataType="3"/>
      <item name="WorkPart.WorkingCode" value="0" dataType="2"/>
    </items>
  </body>
</root>
```

With these basic structures, the standard default values are reported to the process. Depending on the process, 0..n these default values can be included in the response. The sequence of the individual Items is arbitrary.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2311" version="2.2" eventName="plcChangeOverStarted"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="0"/>
    <trace/>
  </event>
  <body>
    <items>
      <item name="WorkPart.PartsPerCarrier" value="6" dataType="3"/>
      <item name="WorkPart.Family" value="AA8823108" dataType="8"/>
      <item name="WorkPart.PartsPerRow" value="2" dataType="3"/>
      <item name="WorkPart.PartsPerCol" value="3" dataType="3"/>
      <item name="WorkPart.WorkingCode" value="0" dataType="2"/>
    </items>
  </body>
</root>
```

```

<item name="ProcessSetup1.Name" value="IntSample" dataType="8"/>
<item name="ProcessSetup1.SetValue" value="5" dataType="3"/>
<item name="ProcessSetup1.LoLim" value="2" dataType="3"/>
<item name="ProcessSetup1.UpLim" value="8" dataType="3"/>
<item name="ProcessSetup1.CheckType" value="5" dataType="2"/>
<item name="ProcessSetup1.Unit" value="N" dataType="8"/>
<item name="ProcessSetup2.Name" value="RealSample" dataType="8"/>
<item name="ProcessSetup2.SetValue" value="5.00000000" dataType="5"/>
<item name="ProcessSetup2.LoLim" value="2.00000000" dataType="5"/>
<item name="ProcessSetup2.UpLim" value="8.00000000" dataType="5"/>
<item name="ProcessSetup2.CheckType" value="5" dataType="2"/>
<item name="ProcessSetup2.Unit" value="N" dataType="8"/>
<item name="Prog1.Name" value="PRG_MILL_1111111111" dataType="8"/>
<item name="Prog1.SubSystem" value="BL922" dataType="8"/>
<item name="Tool1.Family" value="1002090" dataType="8"/>
<item name="Tool1.Name" value="Mask" dataType="8"/>
<item name="Tool2.Family" value="786644" dataType="8"/>
<item name="Tool2.Name" value="Downholder" dataType="8"/>
<item name="Component1.TypeNo" value="8936971155" dataType="8"/>
<item name="Component1.TypeVar" value="" dataType="8"/>
<item name="Component1.Name" value="Con1" dataType="8"/>
<item name="Component2.TypeNo" value="4560981278" dataType="8"/>
<item name="Component2.TypeVar" value="" dataType="8"/>
<item name="Component2.Name" value="Con2" dataType="8"/>
</items>
</body>
</root>

```

In the event of a **negative response** no additional data is sent to the station.

Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2311" version="2.2" eventName="plcChangeOverStarted"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="-1">MES detected an error.</result>
    <trace>
      <trace level="error" code="5543" text="Invalid type" source="SomeSource"/>
    </trace>
  </event>
</body>
</root>

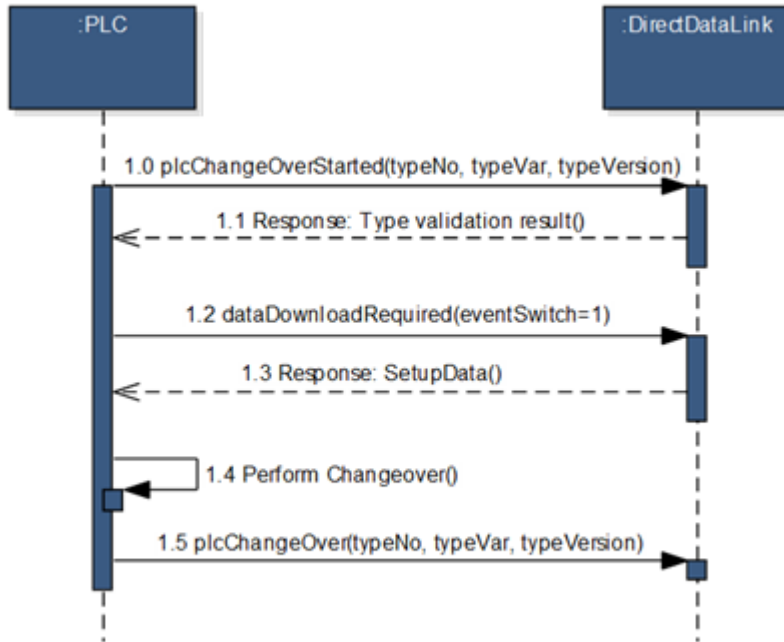
```

Scenario: BPS Change

For correct entry of the changeover times according to BPS, the client must implement the following sequence:

1. Report the start of the changeover process with `plcChangeOverStarted`.
2. Query the changeover parameters with the event `dataDownloadRequired` (`eventSwitch=1`).

3. Report the end of the changeover with `plcChangeOver`.



The individual steps are described again in detail in the following:

Reporting the start of the changeover process

When the station detects that a changeover is required, first a `plcChangeOverStarted` event must be sent. The new tool type must also be sent in the query. On the MES side, a validation takes place when the message is received (for example, does this type exist, is the type released, etc.). The result of this validation is returned to the station as a response. The changeover process can only be continued if this validation was successful. In all other cases, the process must be canceled and the cause of the fault is visualized.

In the following example, the station detects that a tool change to the new type '555.100.100.0' is required. Therefore, the event `plcChangeOverStarted` is sent.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2311" version="2.3" eventName="plcChangeOverStarted"
    contentType="3">
    <location lineNo="8030" statNo="40" statIdx="1" fuNo="1"
      processNo="100" processName="Connector handling"
      application="PLC"/>
  </header>
  <event>
    <plcChangeOverStarted typeNo="5551001000" typeVar="" />
  </event>
  <body/>
</root>
```

If no problems are detected in the review of the type within MES, the type can be fitted. Therefore a positive response is sent to the station:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2311" version="2.3" eventName="plcChangeOverStarted"
    contentType="3" timeStamp="2008-11-23T13:21:34.231+02:00">
    <location lineNo="8030" statNo="40" statIdx="1" fuNo="1"
      processNo="100" processName="Connector handling"
      application="PLC"/>
  </header>
  <event>
    <result returnCode="0" />
  </event>
</root>
```

```

    <trace/>
  </event>
</body>
</root>

```

Request changeover parameters

During the changeover process, the station must query the current changeover parameters from MES. In order to do this the event must be sent with dataDownloadRequired with evenSwitch = 1. MES returns the changeover parameters to be used to the station in the response.

Example

In the following example, the station sends the request for the changeover parameters of the product type 555.100.100.0.

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2311" version="2.3" eventName="dataDownloadRequired"
    eventSwitch="1" contentType="3">
    <location lineNo="8030" statNo="40" statIdx="1" fuNo="1"
      processNo="100" processName="Connector handling"
      application="PLC"/>
  </header>
  <event>
    <dataDownloadRequired />
  </event>
  <body>
    <items>
      <item name="TypeNo" value="5551001000" dataType="8"/>
      <item name="TypeVar" value="" dataType="8"/>
    </items>
  </body>
</root>

```

In response to this event, MES sends the type data defined for this station:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2311" version="2.3" eventName="dataDownloadRequired"
    eventSwitch="1" contentType="3"
    timeStamp="2012-12-17T12:42:57.117+02:00" >
    <location lineNo="8030" statNo="40" statIdx="1" fuNo="1"
      processNo="100" processName="Connector handling"
      application="PLC"/>
  </header>
  <event>
    <result returnCode="0"/>
    <trace/>
  </event>
  <body>
    <items>
      <item name="Component1.TypeNo" value="8936971155" dataType="8"/>
      <item name="Component1.TypeVar" value="" dataType="8"/>
      <item name="Component1.Name" value="Con1" dataType="8"/>
      <item name="Component2.TypeNo" value="4560981278" dataType="8"/>
      <item name="Component2.TypeVar" value="" dataType="8"/>
      <item name="Component2.Name" value="Con2" dataType="8"/>
    </items>
  </body>
</root>

```

Reporting the end of the changeover process

As the last action in the changeover process, the completion of the changeover must be reported to MES. The station sends the event `plcChangeOver` to MES. This event is only used for notification, therefore MES doesn't send an answer to this message.


If an error occurs during the changeover process or the changeover process is canceled, the station cannot send a `plcChangeOver` event.

Example

In the following example, the station has successfully changed over to the new product type '555.100.100.0'. As a final step, it sends the event `plcChangeOver`, in order to inform MES about the end of the changeover.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2311" version="2.3" eventName="plcChangeOver"
    contentType="3">
    <location lineNo="8030" statNo="40" statIdx="1" fuNo="1"
      processNo="100" processName="Connector handling"
      application="PLC"/>
  </header>
  <event>
    <plcChangeOver typeNo="5551001000" typeVar=""/>
  </event>
  <body/>
</root>
```

4.2.1.7 Scenario: Simplified Changeover (Obsolete)

 *The simplified changeover has been marked as obsolete and will be deleted from the specification in one of the next versions. This scenario should therefore only be used in new projects in exceptional cases.*

The changeover is covered by the events `plcChangeOverStarted` and `plcChangeOver`. `plcChangeOverStarted` is sent to MES before or immediately after the start of the changeover process. For this event a response is sent to the station. This response then contains the current changeover specifications and the information regarding whether changeover is permitted.

The event `plcChangeOver` is sent from the station immediately following successful changeover. The current information about the changed type can then be forwarded to the MES. After receiving the message, MES does not send an acknowledgment to the station.

Initiate changeover:

The changeover is initiated by sending the event `plcChangeOverStarted`. The response for the station contains the current changeover parameters for the specified product type. Beside the attributes specified in part I no extra information have to be send.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2311" version="2.3" eventName="plcChangeOverStarted"
    contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
```

```

    <event>
      <plcChangeOverStarted typeNo="1111111111" typeVar="0815"/>
    </event>
  </body>
</root>

```

In a **positive response**, in addition to the data specified above, changeover parameters relevant for the station are also sent.

Example

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2311" version="2.3" eventName="plcChangeOverStarted"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="0"/>
    <trace/>
  </event>
  <body>
    <items>
      <item name="WorkPart.PartsPerCarrier" value="6" dataType="3"/>
      <item name="WorkPart.Family" value="AA8823108" dataType="8"/>
      <item name="WorkPart.PartsPerRow" value="2" dataType="3"/>
      <item name="WorkPart.PartsPerCol" value="3" dataType="3"/>
      <item name="WorkPart.WorkingCode" value="0" dataType="2"/>
      <item name="ProcessSetup1.Name" value="IntSample" dataType="8"/>
      <item name="ProcessSetup1.SetValue" value="5" dataType="3"/>
      <item name="ProcessSetup1.LoLim" value="2" dataType="3"/>
      <item name="ProcessSetup1.UpLim" value="8" dataType="3"/>
      <item name="ProcessSetup1.CheckType" value="5" dataType="2"/>
      <item name="ProcessSetup1.Unit" value="N" dataType="8"/>
      <item name="ProcessSetup2.Name" value="RealSample" dataType="8"/>
      <item name="ProcessSetup2.SetValue" value="5.00000000" dataType="5"/>
      <item name="ProcessSetup2.LoLim" value="2.00000000" dataType="5"/>
      <item name="ProcessSetup2.UpLim" value="8.00000000" dataType="5"/>
      <item name="ProcessSetup2.CheckType" value="5" dataType="2"/>
      <item name="ProcessSetup2.Unit" value="N" dataType="8"/>
      <item name="Prog1.Name" value="PRG_MILL_1111111111" dataType="8"/>
      <item name="Prog1.SubSystem" value="BL922" dataType="8"/>
      <item name="Tool1.Family" value="1002090" dataType="8"/>
      <item name="Tool1.Name" value="Mask" dataType="8"/>
      <item name="Tool2.Family" value="786644" dataType="8"/>
      <item name="Tool2.Name" value="Downholder" dataType="8"/>
      <item name="Component1.TypeNo" value="8936971155" dataType="8"/>
      <item name="Component1.TypeVar" value="" dataType="8"/>
      <item name="Component1.Name" value="Con1" dataType="8"/>
      <item name="Component2.TypeNo" value="4560981278" dataType="8"/>
      <item name="Component2.TypeVar" value="" dataType="8"/>
      <item name="Component2.Name" value="Con2" dataType="8"/>
    </items>
  </body>
</root>

```

In a **negative response**, no additional data is sent to the station:

Example

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2311" version="2.2" eventName="plcChangeOverStarted"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">

```


```

        <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
            processNo="90" processName="PlasmaCleaning" application="PLC"/>
    </header>
    <event>
        <result returnCode="-1">MES detected an error.</result>
        <trace>
            <trace level="error" code="5543" text="Invalid type" source="SomeSource"/>
        </trace>
    </event>
</body>
</root>

```

4.2.1.8 Completing the Changeover Process

After the process has been successfully changed to the new type, the event must be `plcChangeOver` sent to MES. This is to inform the system that the station is equipped with the type specified in the event.

 *The client cannot assume that MES always contains the information about which type the process was last changed to. If this information is strictly essential for the process, this information must be secured by another means.*

Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
    <header eventId="2312" version="2.2" eventName="plcChangeOver"
        contentType="3">
        <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
            processNo="90" processName="PlasmaCleaning" application="PLC"/>
    </header>
    <event>
        <plcChangeOver typeNo="1111111111" typeVar=""/>
    </event>
</body>
</root>

```


4.2.2 Setting Up the Tool

The setup of a tool is handled by the `plcToolChangeStarted` and `plcToolChanged` events.

Once the tool change is complete, the process must notify MES of this with the `plcToolChanged` event. The tool is then assigned to the process on the MES side. MES sends an acknowledgment to the process to confirm the successful posting of the tool. Only after a positive message from MES has been received for `plcToolChanged` may the process use the tool.

4.2.2.1 Initiating Tool Setup

The MES is informed of the start of a tool changeover by the system sending `plcToolChangeStarted`. `plcToolChangeStarted` is sent to MES before or immediately after the start of the tool changeover. For this event, a response is sent to the station. The response contains information regarding whether the tool is locked and if so, which locks are in place.

 *With `plcToolChangeStarted`, the tool has not yet been posted to the requesting process. If the changeover process is canceled, no further action must be taken.*

If the system detects that the tool is locked or only partially released, a warning is returned as a response. The detailed information about the new tool is transferred in the element `tools` as a structure array. A single entry can contain the following information:

Information	Required	Data type	OpCon Xml	Comments
-------------	----------	-----------	-----------	----------

Tool identification	Mandatory	STRING	tools[]@toolID	Uniquely identifies the tool.
Status of the tool	Optional	INT	tools[]@state	Possible status: <ul style="list-style-type: none"> -1 = Unknown status 0 = Enabled 1 = Warning limit reached 2 = Locked
Current status of counter stroke	Optional	DINT	tools[]@counter	-
Tool family	Optional	STRING	tools[]@family	Contains the name of the tool family. If no family is present, an empty string ("") is entered here.

Example:

This example shows a request from a process for a tool change. The process is currently equipped with the tool 085G9ZA00931_02V01. This is to be replaced by the tool 085G9ZA00931_02V02.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="23119" version="2.2" eventName="plcToolChangeStarted"
    contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <plcToolChangeStarted identifier="085G9ZA00931_02V02"
      oldIdentifier="085G9ZA00931_02V01"/>
  </event>
</body>
</root>
```

MES reports the tool with the ID 085G9ZA00931_02V02 as not locked.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="23119" version="2.2" eventName="plcToolChangeStarted"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="0"/>
    <trace/>
  </event>
  <body>
    <structArrays>
      <array name="tools">
        <structDef>
          <item name="toolID" dataType="8"/>
          <item name="state" dataType="2"/>
          <item name="counter" dataType="3"/>
          <item name="family" dataType="8"/>
        </structDef>
        <values>
          <item toolID="085G9ZA00931_02V02" state="0" counter="119999"
            family="085G9ZA00931_02V" />
        </values>
      </array>
    </structArrays>
  </body>
</root>
```

If MES detects that the tool is locked, then the following response is sent:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="23119" version="2.2" eventName="plcToolChangeStarted"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="-2">MES detected a warning.</result>
    <trace>
      <trace level="warning" code="12334" text="Blocking level reached"
        source="ToolControl"/>
    </trace>
  </event>
  <body>
    <structArrays>
      <array name="tools">
        <structDef>
          <item name="toolID" dataType="8"/>
          <item name="state" dataType="2"/>
          <item name="counter" dataType="3"/>
          <item name="family" dataType="8"/>
        </structDef>
        <values>
          <item toolID="085G9ZA00931_02V02" state="2" counter="125001"
            family="085G9ZA00931_02V"/>
        </values>
      </array>
    </structArrays>
  </body>
</root>
```

If the tool reaches its warning limit, this warning is also reported to the station:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="23119" version="2.2" eventName="plcToolChangeStarted"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="-2">MES detected a warning.</result>
    <trace>
      <trace level="warning" code="12333" text="Warning level reached"
        source="ToolControl"/>
    </trace>
  </event>
  <body>
    <structArrays>
      <array name="tools">
        <structDef>
          <item name="toolID" dataType="8"/>
          <item name="state" dataType="2"/>
          <item name="counter" dataType="3"/>
          <item name="family" dataType="8"/>
        </structDef>
        <values>
          <item toolID="085G9ZA00931_02V02" state="1" counter="120001"
            family="085G9ZA00931_02V"/>
        </values>
      </array>
    </structArrays>
  </body>
</root>
```

4.2.2.2 Conclusion of the Tool Setup

MES is informed of the conclusion of the (successful) change of a tool by transmission of the `plcToolChanged` event. Only after receipt of this event the tool is assigned to the station in the computer system. MES sends an acknowledgment to the station after processing the event.

Aside from the information defined in Part I, there is no additional data sent with the event. The response also includes no additional data.

Example:

The station uses the following message to indicate successful changeover of the new tool with ID 1002090546. Since this is a tool change, the old tool is known and is also indicated to MES:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="23120" version="2.2" eventName="plcToolChanged"
    contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <plcToolChanged identifier="085G9ZA00931_02V02"
      oldIdentifier="085G9ZA00931_02V01"/>
  </event>
</body>
</root>
```

MES responds indicating the successful posting of the tool.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="23120" version="2.2" eventName="plcToolChanged"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="0"/>
    <trace/>
  </event>
</body>
</root>
```

4.2.3 Changing the Material

The change of materials (components) is covered by the events `plcMaterialChangeStarted` and `plcMaterialChanged`.

When the material has been successfully changed, the station must report this to the MES using the event `plcMaterialChanged`. MES sends an acknowledgment to the station to confirm the successful posting of the material. The station cannot use the material until after a positive message from the MES for `plcMaterialChanged` is received.

4.2.3.1 Initiating the Material Change

The MES is informed of the start of a material change by the system sending `plcMaterialChangeStarted`. `plcMaterialChangeStarted` is sent to MES before or immediately after the start of the material change. For this event, a response is sent to the station. This response contains the information regarding whether the material is blocked and, if so, which locks are in place.

💡 *With `plcMaterialChangeStarted` the material is not yet posted to the requesting station. If the changeover process is canceled, no further action must be taken.*

If it is detected that the material is locked or only partially released, a warning is returned as a response. The detailed information about the new material is transferred in the element `matData` as a structure array.

A single entry can contain the following information:

Information	Required	Data type	OpCon Xml	Comments
Material ID	Mandatory	STRING	<code>matData[]@matID</code>	Uniquely identifies the material.
Status of the material	Optional	INT	<code>matData[]@state</code>	Possible status: <ul style="list-style-type: none"> -1 = Unknown status 0 = Enabled 1 = Warning limit reached (e.g. if MSD time is exceeded) 2 = Locked
Material number	Optional	STRING	<code>matData[]@partNo</code>	TTNR of the material that the station has to install. This number corresponds to e.g. the field 'P' (Customer Part Number) of the Mat Label specification.
Type part number of the material (obsolete)	Optional	STRING(10)	<code>matData[]@typeNo</code>	Obsolete, do not use for new interfaces. Bosch type part number
Type variant of the material (obsolete)	Optional	STRING(10)	<code>matData[]@typeVar</code>	Obsolete, do not use for new interfaces. Type variant/version of the material
Revision status of the material	Optional	STRING	<code>matData[]@revision</code>	Version of the material that the station has to install
Batch	Optional	STRING	<code>matData[]@batch</code>	Manufacturer batch
Manufacturer / supplier	Optional	STRING	<code>matData[]@vendor</code>	-
Residual quantity	Optional	REAL	<code>matData[]@qty</code>	Residual amount of material
Master Mat ID	Optional	STRING	<code>matData[]@masterMatID</code>	
ERP ID of the material	Optional	STRING	<code>matData[]@erpID</code>	
Time stamp of material entry	Optional	STRING	<code>matData[]@rcvTimeStamp</code>	Specifies when the material was posted in the incoming goods station.
ID of the incoming goods station	Optional	STRING	<code>matData[]@rcvStation</code>	
Operator of the incoming goods station	Optional	STRING	<code>matData[]@operator</code>	
Version number of the stored Mat Label	Optional	DINT	<code>matData[]@labelVersion</code>	
Manufacturer part number	Optional	STRING	<code>matData[]@manufacturerTypeNo</code>	
Order number	Optional	STRING	<code>matData[]@orderingNo</code>	
Additional data for the material	Optional	STRING	<code>matData[]@partAddInfo</code>	
Manufacturing time stamp	Optional	STRING	<code>matData[]@productionDate</code>	
Expiry date	Optional	STRING	<code>matData[]@expirationDate</code>	

roHS Info	Optional	STRING	matData[]@roHS	
MSD level	Optional	STRING	matData[]@msLevel	
Order number	Optional	STRING	matData[]@purchaseOrderNo	
Delivery note number	Optional	STRING	matData[]@shippingNoteNo	
Supplier	Optional	STRING	matData[]@supplierID	
Package ID	Optional	STRING	matData[]@packageID	
Unit of measurement	Optional	STRING	matData[]@quantityUnit	For example meters, kilograms, units, etc.
Number of batches	Optional	DINT	matData[]@batchCounter	Specifies how many batches are grouped together to form this material unit
Batch ID 1	Optional	STRING	matData[]@batch1	Batch ID of material
Batch ID2	Optional	STRING	matData[]@batch2	Optional additional batch
Supplier data	Optional	STRING	matData[]@supplierData	
No. of SPUs	Optional	DINT	matData[]@numberOfSPU	
Residual quantity	Optional	LREAL	matData[]@remainingQty	
Time stamp of the first opening	Optional	STRING	matData[]@firstOpenedTimestamp	Specifies when the material has been fully consumed.
Consumed time stamp	Optional	STRING	matData[]@usedUpTimestamp	Specifies when the material has been fully consumed.
Current setup location	Optional	STRING	matData[]@currentLocation	
Current setup	Optional	STRING	matData[]@currentSetup	Only relevant in connection with component insertion machines.
Position within the machine	Optional	STRING	matData[]@stationInfo	Only relevant in connection with component insertion machines.
Subsequently set expiry date	Optional	STRING	matData[]@expirationDateOverride	
Material to which this material has been spliced.	Optional	STRING	matData[]@splicedPrefixMatID	Only relevant in connection with component insertion machines.
Original unit of measure	Optional	LREAL	matData[]@QuantityRawUnit	
Format of the Mat Label	Optional	STRING	matData[]@labelFormat	

💡 Not every client has the option to process structArrays of any width. Therefore, in the definition of the materials to be reported, it is important to check whether the relevant client can process this quantity.

💡 *matData* can have more than one entry. In this case, another material has been added to the requested material, e.g. for component rolls by splicing the remaining roll. Because these materials can each have their own status, all the information is reported. The order of the entries within *matData* then corresponds to the order in which they appear on the reels or spools, if the reel or spool permits an order.

💡 The (remaining) quantity of a material may only be specified in the basic unit. The following basic units are currently allowed:

Length: meters (MTR)

Surface area: meters squared

Weight: kilograms (KGM)

Number: units (NAR)

Volume: liters (LTR)

💡 The complete list of units to be supported is documented in the Mat Label specification or UNECE Recommendation 20.

Optionally, the response may still contain the lock reasons for the individual materials. The information about the lock reason of the new material is transferred in the element `matBlocks` as a structure array. A single entry can contain the following information:

Information	Required	Data type	OpCon Xml	Comments
Material ID	X	STRING	<code>matData[]@matID</code>	Uniquely identifies the material.
Block code	Optional	DINT	<code>matBlocks[]@reasonID</code>	Describes the reason for the lock.
Lock description	Optional	STRING	<code>matBlock[]@reasonText</code>	Descriptive text of the lock reason.

Example:

Request from a station for a material change. The process is currently equipped with the material 98076152430192834465001. This is to be replaced with the material 98076152430192834473468.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="23117" version="2.2" eventName="plcMaterialChangeStarted"
    contentType="3">
    <location lineNo="1" statNo="10" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="5" processName="SMT" application="PanaCIM"/>
  </header>
  <event>
    <plcMaterialChangeStarted identifier="98076152430192834473468"
      oldIdentifier="98076152430192834465001"/>
  </event>
</body>
</root>
```

MES reports the material with the ID 98076152430192834473468 as not locked.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="23117" version="2.2" eventName="plcMaterialChangeStarted"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="10" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="5" processName="SMT" application="PanaCIM"/>
  </header>
  <event>
    <result returnCode="0"/>
    <trace/>
  </event>
  <body>
    <structArrays>
      <array name="matData">
        <structDef>
          <item name="matID" dataType="8"/>
          <item name="state" dataType="2"/>
          <item name="typeNo" dataType="8"/>
          <item name="typeVar" dataType="8"/>
          <item name="qty" dataType="5"/>
        </structDef>
        <values>
          <item matID="98076152430192834473468" state="0"
            typeNo="1236532775" typeVar="12" qty="2500.00"/>
        </values>
      </array>
      <array name="matBlocks">
        <structDef>
          <item name="matID" dataType="8"/>
          <item name="reasonID" dataType="3"/>
          <item name="reasonText" dataType="8"/>
        </structDef>
      </array>
    </structArrays>
  </body>
</root>
```

```

        <values/>
    </array>
</structArrays>
</body>
</root>

```

Additional material still remains on the component roll. In this case, the data of the residual material is also transferred:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="23117" version="2.2" eventName="plcMaterialChangeStarted"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="10" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="5" processName="SMT" application="PanaCIM"/>
  </header>
  <event>
    <result returnCode="0"/>
    <trace/>
  </event>
  <body>
    <structArrays>
      <array name="matData">
        <structDef>
          <item name="matID" dataType="8"/>
          <item name="state" dataType="2"/>
          <item name="typeNo" dataType="8"/>
          <item name="typeVar" dataType="8"/>
          <item name="qty" dataType="5"/>
        </structDef>
        <values>
          <item matID="98076152430192800000111" state="0"
            typeNo="1236532775" typeVar="12" qty="150.00"/>
          <item matID="98076152430192834473468" state="0"
            typeNo="1236532775" typeVar="12" qty="2500.00"/>
        </values>
      </array>
      <array name="matBlocks">
        <structDef>
          <item name="matID" dataType="8"/>
          <item name="reasonID" dataType="3"/>
          <item name="reasonText" dataType="8"/>
        </structDef>
        <values/>
      </array>
    </structArrays>
  </body>
</root>

```

MES reports the material with the ID 98076152430192834473468 as locked.

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="23117" version="2.2" eventName="plcMaterialChangeStarted"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="10" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="5" processName="SMT" application="PanaCIM"/>
  </header>
  <event>
    <result returnCode="-2">MES detected a warning.</result>
    <trace>
      <trace level="warning" code="100234" text="Material blocked."
        source="MatControl"/>
    </trace>
  </event>
  <body>
    <structArrays>

```

```

<array name="matData">
  <structDef>
    <item name="matID" dataType="8"/>
    <item name="state" dataType="2"/>
    <item name="typeNo" dataType="8"/>
    <item name="typeVar" dataType="8"/>
    <item name="qty" dataType="5"/>
  </structDef>
  <values>
    <item matID="98076152430192834473468" state="2"
      typeNo="1236532775" typeVar="12" qty="2500.00"/>
  </values>
</array>
<array name="matBlocks">
  <structDef>
    <item name="matID" dataType="8"/>
    <item name="reasonID" dataType="3"/>
    <item name="reasonText" dataType="8"/>
  </structDef>
  <values>
    <item matID="98076152430192834473468" reasonID="17"
      reasonText="Some reason"/>
  </values>
</array>
</structArrays>
</body>
</root>

```

Additional material still remains on the component roll. The scanned material is locked, the residual material is not locked:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="23117" version="2.2" eventName="plcMaterialChangeStarted"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="10" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="5" processName="SMT" application="PanaCIM"/>
  </header>
  <event>
    <result returnCode="-2">MES detected a warning.</result>
    <trace>
      <trace level="warning" code="100234" text="Material blocked."
        source="MatControl"/>
    </trace>
  </event>
  <body>
    <structArrays>
      <array name="matData">
        <structDef>
          <item name="matID" dataType="8"/>
          <item name="state" dataType="2"/>
          <item name="typeNo" dataType="8"/>
          <item name="typeVar" dataType="8"/>
          <item name="qty" dataType="5"/>
        </structDef>
        <values>
          <item matID="98076152430192800000111" state="0"
            typeNo="1236532775" typeVar="12" qty="150.00"/>
          <item matID="98076152430192834473468" state="2"
            typeNo="1236532775" typeVar="12" qty="2500.00"/>
        </values>
      </array>
      <array name="matBlocks">
        <structDef>
          <item name="matID" dataType="8"/>
          <item name="reasonID" dataType="3"/>

```

```

        <item name="reasonText" dataType="8"/>
    </structDef>
    <values>
        <item matID="98076152430192834473468" reasonID="17"
            reasonText="Some reason" />
    </values>
</array>
</structArrays>
</body>
</root>

```

MES reports the material with the ID 98076152430192834473468 as locked (two locks are in place).

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
    <header eventId="23117" version="2.2" eventName="plcMaterialChangeStarted"
        timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
        <location lineNo="1" statNo="10" statIdx="1" fuNo="1" workPos="1" toolPos="1"
            processNo="5" processName="SMT" application="PanaCIM"/>
    </header>
    <event>
        <result returnCode="-2">MES detected a warning.</result>
        <trace>
            <trace level="warning" code="100234" text="Material blocked."
                source="MatControl"/>
        </trace>
    </event>
    <body>
        <structArrays>
            <array name="matData">
                <structDef>
                    <item name="matID" dataType="8"/>
                    <item name="state" dataType="2"/>
                    <item name="typeNo" dataType="8"/>
                    <item name="typeVar" dataType="8"/>
                    <item name="qty" dataType="5"/>
                </structDef>
                <values>
                    <item matID="98076152430192834473468" state="2"
                        typeNo="1236532775" typeVar="12" qty="2500.00"/>
                </values>
            </array>
            <array name="matBlocks">
                <structDef>
                    <item name="matID" dataType="8"/>
                    <item name="reasonID" dataType="3"/>
                    <item name="reasonText" dataType="8"/>
                </structDef>
                <values>
                    <item matID="98076152430192834473468" reasonID="17"
                        reasonText="Some reason"/>
                    <item matID="98076152430192834473468" reasonID="678"
                        reasonText="Some other reason"/>
                </values>
            </array>
        </structArrays>
    </body>
</root>

```

4.2.3.2 Completing the Material Change

The completion of the successful material change is reported to MES by sending the event `plcMaterialChanged`. In terms of data, the material is not technically assigned to the station until this event is received. After processing the event, MES sends an acknowledgment to the station.

Apart from the information defined in part I, no additional data is sent with the event. The response also does not contain any additional data.

Example:

The station uses the following message to report the successful setup of the new material with the ID 98076152430192834465001. Because this is a material change, the old material is known and is reported to MES:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="23118" version="2.3" eventName="plcMaterialChanged"
    contentType="3">
    <location lineNo="1" statNo="10" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="5" processName="SMT" application="PanaCIM"/>
  </header>
  <event>
    <plcMaterialChanged identifier="980761524301928344 65001"
      oldIdentifier="98076152430192834473468"/>
  </event>
  <body/>
</root>
```

MES reports the successful posting of the material:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="23118" version="2.3" eventName="plcMaterialChanged"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="10" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="5" processName="SMT" application="PanaCIM"/>
  </header>
  <event>
    <result returnCode="0"/>
    <trace/>
  </event>
  <body/>
</root>
```

4.3 Production Data

During production, stations must exchange various production-related (workpiece-related) data with the MES using the following events:

- partReceived
- partProcessingStarted
- partProcessingPaused
- partProcessingAborted
- partProcessed
- partDisplaced

4.3.1 Processing Release

The processing release from MES is determined during event `partReceived`. This event is usually triggered by processes when a workpiece is fed into the station. The processing release can be composed of several partial releases, which are sent to `partReceived` the process in the response.

If the correct setup of the tools and materials requires verification while the processing release is being determined, the IDs of the tools and materials must be included in the request.

4.3.1.1 Verifying the Tool Setup

For the tools that have been changed over, the following information must be sent:

Information	Required	Data type	OpCon Xml	Comments
Changed tool	Optional	STRING[]	tools[]	Array with the tool IDs that are currently set up or in use.

From an MES perspective, tool carriers (WPC = Work Part Carriers) are also a kind of tool. If workpiece carriers are to be monitored, the ID of the workpiece carrier must also be included in the tools[] array.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2321" version="2.3" eventName="partReceived"
    contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partReceived identifier="DMC22934" typeNo="1111111111" typeVar="" />
  </event>
  <body>
    <arrays>
      <array name="tools" dataType="8">
        <item value="1002090001"/>
        <item value="786644001"/>
      </array>
    </arrays>
  </body>
</root>
```

4.3.1.2 Verifying the Material Setup

For the changed materials, the following information must be sent:

Information	Required	Data type	OpCon Xml	Comments
Materials set up	Optional	STRING[]	components []	Array with the matlIDs of the materials that are currently set up or in use.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2321" version="2.3" eventName="partReceived"
    contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partReceived identifier="DMC22934" typeNo="1111111111" typeVar="" />
  </event>
  <body>
    <arrays>
      <array name="components" dataType="8">
        <item value="98076152430192834465001"/>
        <item value="32890234122342318827001"/>
      </array>
    </arrays>
  </body>
</root>
```


4.3.1.3 Verifying the Setup

The processing release may include a check of the setup(s). To do this, the processing station does not have to send any additional information, because the setup is uniquely identified using the information in the message header.

The response contains a Struct-Array, in which the locked locations are listed. It is possible that a location may have been locked for several reasons. In this case, more than one lock is also returned to the station.

In addition, the response may also contain locks that do not directly correspond with their own location. This scenario can be used, for example, if the processing request and the end of the processing are not represented by the same location (for example, processing request takes place at processing position 1, and reporting the end of processing takes place at processing position 3). If you have already ensured in the processing request that no relevant processing position within the station is locked, MES can return all locks of the station or the functional unit to the station. If a lock is detected, the setup status (item.LocationState) is set to locked. The existing locks are also sent to the station.

The station can now validate again whether the locations contained in the response (e.g. processing positions) are used, or whether the processing can also be performed without these locations. If the processing can be performed without the use of the locked locations, the station can agree to lock the station on the MES side. Otherwise, the station is not permitted to perform processing.

4.3.1.4 Verifying Uniquely Identifiable Components (Additional Components)

If a process installs a uniquely identifiable component ("A-component"), this must also be taken into account in the processing request. The process can only be performed if a release exists for the product (main part) and the component.

Information	Required	Data type	OpCon Xml	Comments
ID of the insert component	X	STRING	item. InsertComponentID	Unique ID of the additional component

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2321" version="2.3" eventName="partReceived"
    contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partReceived identifier="DMC22934" typeNo="111111111" typeVar="" />
  </event>
  <body>
    <items>
      <item name="InsertComponentID" value="IAC27110701" dataType="8"/>
    </items>
  </body>
</root>
```

These basic structures are used to report the new workpiece. Depending on the process, 0..n these default values can be included in the response. The sequence of the individual Items is optional, but the numbering of the individual standard structures must always begin with 1.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2321" version="2.3" eventName="partReceived"
    contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
```

```

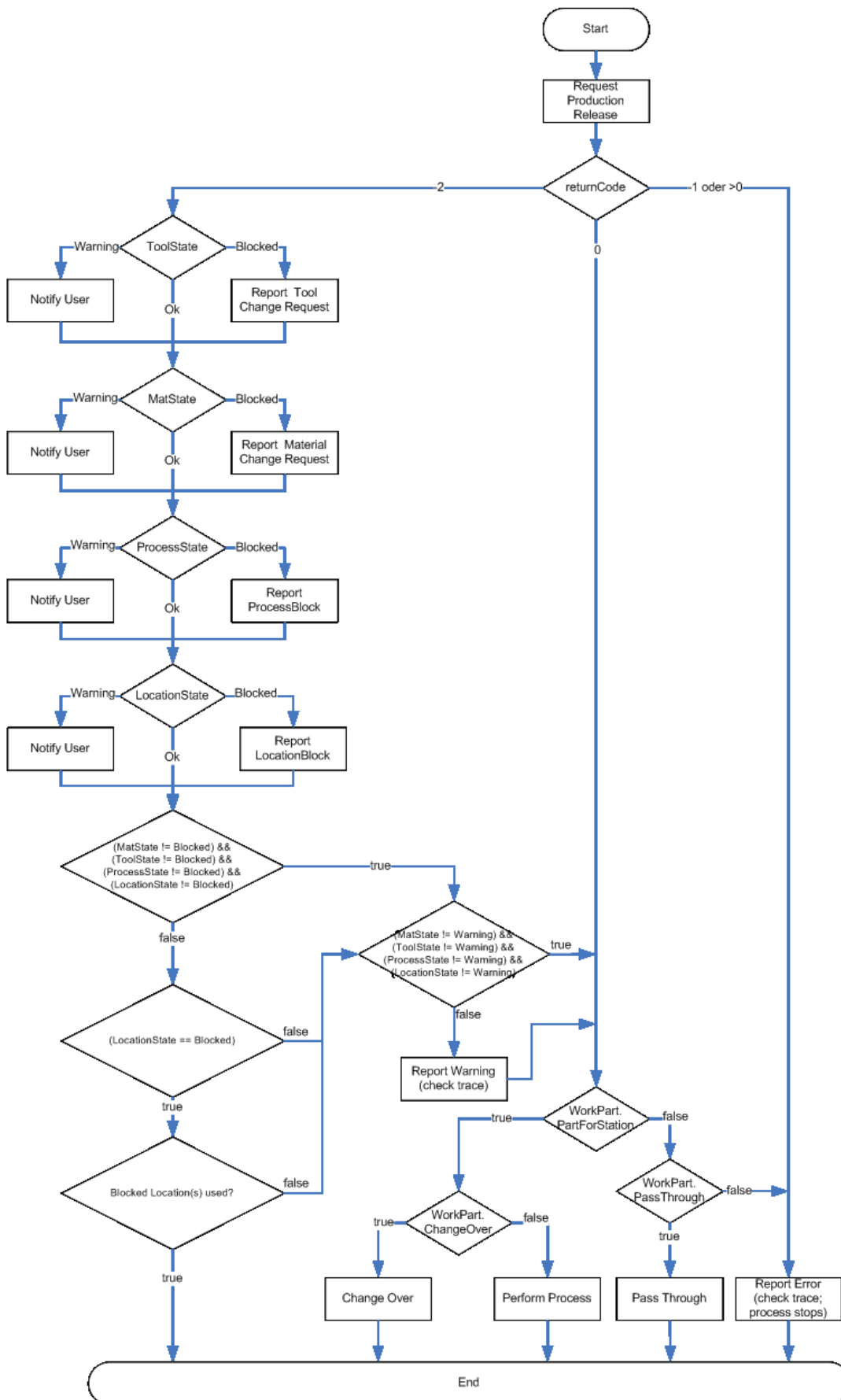
        processNo="90" processName="PlasmaCleaning" application="PLC"/>
</header>
<event>
  <partReceived identifier="DMC22934" typeNo="111111111" typeVar=""/>
</event>
<body>
  <arrays>
    <array name="components" dataType="8">
      <item value="98076152430192834465001"/>
      <item value="32890234122342318827001"/>
    </array>
    <array name="tools" dataType="8">
      <item value="1002090001"/>
      <item value="786644001"/>
    </array>
  </arrays>
  <items>
    <item name="InsertComponentID" value="IAC27110701" dataType="8"/>
  </items>
</body>
</root>

```



Which of the options described above must be sent from a specific process is not included in this document and must therefore be defined specifically for each process.

The processing release is sent in the response message. How the individual process interprets the answer, is shown in the following flow chart:



If the request can be processed by MES in principle (`returnCode` ≤ 0), the following information is sent in the response:

Information	Required	Data type	OpCon Xml	Comments
Workpiece release	Optional	BOOL	workPart@partForStation	Defines whether the workpiece status (e.g. production progress, condition, processing counter, etc.) meets the release conditions.
Workpiece ID	Optional	STRING	workPart@identifier	ID of the workpiece
Changeover required	Optional	BOOL	workPart@changeOver	Determines whether the process must be changed from an MES perspective: <ul style="list-style-type: none"> • true = Changeover required • false = No changeover required
Processing code	Optional	INT	workPart@workingCode	Classification of the workpieces(s) (e.g., series part(s), calibration part(s), etc.)
Model number	Optional	STRING(10)	workPart@typeNo	(Bosch) model part number of the workpiece
Model variant	Optional	STRING(10)	workPart@typeVar	Model variant (revision, index) of the workpiece
Batch number	Optional	STRING	workPart@batch	ID of the batch to which the workpiece belongs.
Pass-through permitted	Optional	BOOL	workPart@passThrough	This information is only relevant if partForStation=false. passThrough defines whether in this case the workpiece may pass through, or whether it must stop at the station and the station must notify the operator. <ul style="list-style-type: none"> • true = Workpiece may pass through • false = Workpiece must stop at the station
Next process	Optional	DINT	workPart@nextProcessNo	Contains the number of the next process that can process the workpiece.
SPC test(s) to be performed	Optional	DINT	workPart@spcTests	Bit-coded information regarding whether, and which, SPC tests are to be performed. A set bit means that this test must be performed. It is possible for multiple bits to be set at the same time.
Status of the workpiece	Optional	INT	workPart@partState	Status of the workpiece. The values applicable for the status can be taken from the annex of OpCon XML part I.
Tool release	Optional	INT	item.ToolState	Shows the status of the tools. Possible status: <ul style="list-style-type: none"> • -1 = Unknown status • 0 = Enabled • 1 = Warning limit reached • 2 = Locked If more several tools were used, the highest status will be set out of the single status, i.e. if one status is set to locked, item.ToolState will be locked, too.

Material release	Optional	INT	item. MaterialState	Shows the status of the materials. Possible status: <ul style="list-style-type: none"> -1 = Unknown status 0 = Enabled 1 = Warning limit reached (e.g. for MSD time monitoring) 2 = Locked If multiple materials are used, the status is set to the highest value of the individual statuses = As soon as a material is locked, item.MaterialState is also set to locked.
Process release	Optional	INT	item. ProcessState	Shows the status of the process. Possible status: <ul style="list-style-type: none"> -1 = Unknown status 0 = Enabled 2 = Locked
Station release	Optional	INT	item. LocationState	Shows the status of the equipment. Possible statuses: <ul style="list-style-type: none"> -1 = Unknown Status 0 = Enabled 2 = Locked
ID of the insert component	Optional	STRING	insert Component@ identifier	Unique ID of the additional component.
Additional component release	Optional	BOOL	insert Component@ partForStation	Defines whether the status of the additional component (e.g. production progress, status, processing counter, etc.) meets the release conditions.
Processing code of the additional component	Optional	INT	insert Component@ workingCode	Classification of the workpiece(s) (e.g. series part(s), calibration part(s), etc.)
Model number of the additional component	Optional	STRING(10)	insert Component@ typeNo	-
Model variant of the additional component	Optional	STRING(10)	insert Component@ typeVar	-
Batch of the additional component	Optional	STRING	insert Component@ batch	-
Next process for the additional component	Optional	DINT	insert Component@ nextProcesNo	Contains the number of the next process that is permitted to process the additional component.



If a process is locked (*ProcessState* == 2), the whole station must automatically be locked. Even if a station contains several processes, by definition the whole station must be locked and the workpiece is not permitted to enter.

4.3.1.5 Tools

Detailed information about the tools can be transferred in the element `tools`. The following information may be contained in the element `tools` :

Information	Required	Data type	OpCon Xml	Comments
Tool ID	X	STRING	<code>tools[]@toolID</code>	-
Tool status	Optional	INT	<code>tools[]@state</code>	Possible status: -1 = Unknown status 0 = Enabled 1 = Warning limit reached 2 = Locked

4.3.1.6 Materials

Detailed information about the sent materials can be transferred in the element `components`. The following information may be contained in the element `components` :

Information	Required	Data type	OpCon Xml	Comments
Mat ID	X	STRING	<code>components[]@matID</code>	-
Material status	Optional	INT	<code>components[]@state</code>	Possible status: -1 = Unknown status 0 = Enabled 1 = Warning limit reached (e.g. if MSD time is exceeded) 2 = Locked

If you want to transfer detailed information about the locked locations, the element `locationBlocks` must be used. The following information can be included in the element `locationBlocks`:

Equipment

Information	Required	Data type	OpCon Xml	Comments
Line number	Optional	DINT	<code>locationBlocks[]@lineNo</code>	-
Station number	Optional	DINT	<code>locationBlocks[]@statNo</code>	
Station index	Optional	DINT	<code>locationBlocks[]@statIdx</code>	
Functional unit	Optional	DINT	<code>locationBlocks[]@fuNo</code>	
Processing position	Optional	DINT	<code>locationBlocks[]@workPos</code>	
Tool position	Optional	DINT	<code>locationBlocks[]@toolPos</code>	
Block code	Optional	DINT	<code>locationBlocks[]@code</code>	
Additional information	Optional	STRING	<code>locationBlocks[]@info</code>	
Originator of the lock	Optional	STRING	<code>locationBlocks[]@source</code>	
Originator-specific ID	Optional	STRING	<code>locationBlocks[]@sourceToken</code>	

Example:

The following examples show the different responses to different outcomes of determining the processing request. For the request it is assumed that you want to check both the tools and the set-up materials.

The first example shows the default case (workpiece cannot be processed).

The subsequent examples show the different cases, such as tool locked, incorrect process. In these examples, the differences to the default case are highlighted in yellow.

Workpiece can be processed:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2321" version="2.3" eventName="partReceived"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="0"/>
    <trace/>
  </event>
  <body>
    <structs>
      <workPart identifier="DMC22934" partForStation="true" changeOver="false"
        workingCode="0" typeNo="111111111" typeVar="" batch="847320"
        nextProcessNo="90" partState="4" spcTests="0"/>
      <insertComponent identifier="IAC27110701" partForStation="true"
        workingCode="0" typeNo="222222222" typeVar="" batch="KXY90011"
        nextProcessNo="90" partState="4" spcTests="0"/>
    </structs>
    <items>
      <item name="ToolState" value="0" dataType="2"/>
      <item name="MaterialState" value="0" dataType="2"/>
      <item name="ProcessState" value="0" dataType="2"/>
      <item name="LocationState" value="0" dataType="2"/>
    </items>
    <structArrays>
      <array name="tools">
        <structDef>
          <item name="toolID" dataType="8"/>
          <item name="state" dataType="2"/>
        </structDef>
        <values>
          <item toolID="1002090001" state="0"/>
          <item toolID="786644001" state="0"/>
        </values>
      </array>
      <array name="components">
        <structDef>
          <item name="matID" dataType="8"/>
          <item name="state" dataType="2"/>
        </structDef>
        <values>
          <item matID="98076152430192834465001" state="0"/>
          <item matID="32890234122342318827001" state="0"/>
        </values>
      </array>
      <array name="locationBlocks">
        <structDef>
          <item name="lineNo" dataType="3" />
          <item name="statNo" dataType="3" />
          <item name="statIdx" dataType="3" />
          <item name="fuNo" dataType="3" />
          <item name="workPos" dataType="3" />
          <item name="toolPos" dataType="3" />
          <item name="code" dataType="3" />
          <item name="info" dataType="8" />
          <item name="source" dataType="8" />
          <item name="sourceToken" dataType="8" />
        </structDef>
        <values/>
      </array>
    </structArrays>
  </body>
</root>
```

```
</root>
```

Workpiece locked (incorrect process):

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2321" version="2.3" eventName="partReceived"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="0"/>
    <trace/>
  </event>
  <body>
    <structs>
      <workPart identifier="DMC22934" partForStation="false" changeOver="false"
        workingCode="0" typeNo="111111111" typeVar="" batch="847320"
        nextProcessNo="70" partState="4" spcTests="0"/>
      <insertComponent identifier="IAC27110701" partForStation="true"
        workingCode="0" typeNo="222222222" typeVar="" batch="KXY90011"
        nextProcessNo="90" partState="4" spcTests="0"/>
    </structs>
    <items>
      <item name="ToolState" value="0" dataType="2"/>
      <item name="MaterialState" value="0" dataType="2"/>
      <item name="ProcessState" value="0" dataType="2"/>
      <item name="LocationState" value="0" dataType="2"/>
    </items>
    <structArrays>
      <array name="tools">
        <structDef>
          <item name="toolID" dataType="8"/>
          <item name="state" dataType="2"/>
        </structDef>
        <values>
          <item toolID="1002090001" state="0"/>
          <item toolID="786644001" state="0"/>
        </values>
      </array>
      <array name="components">
        <structDef>
          <item name="matID" dataType="8"/>
          <item name="state" dataType="2"/>
        </structDef>
        <values>
          <item matID="98076152430192834465001" state="0"/>
          <item matID="32890234122342318827001" state="0"/>
        </values>
      </array>
      <array name="locationBlocks">
        <structDef>
          <item name="lineNo" dataType="3" />
          <item name="statNo" dataType="3" />
          <item name="statIdx" dataType="3" />
          <item name="fuNo" dataType="3" />
          <item name="workPos" dataType="3" />
          <item name="toolPos" dataType="3" />
          <item name="code" dataType="3" />
          <item name="info" dataType="8" />
          <item name="source" dataType="8" />
          <item name="sourceToken" dataType="8" />
        </structDef>
      </array>
    </body>
  </root>
```



```

        <values/>
    </array>
</structArrays>
</body>
</root>

```

Workpiece is locked (internal MES processing error):

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2321" version="2.3" eventName="partReceived"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="-1">MES detected an error.</result>
    <trace>
      <trace level="error" code="700020012" text="Communication failure with
service vMDT (maybe down)." source="LineControl"/>
    </trace>
  </event>
  <body>
    <structs>
      <workPart identifier="DMC22934" partForStation="false" changeOver="false"
        workingCode="0" typeNo="111111111" typeVar="" batch="847320"
        partState="4" spcTests="0" />
      <insertComponent identifier=" IAC27110701" partForStation="true"
        workingCode="0" typeNo="222222222" typeVar="" batch="KXY90011"
        nextProcessNo="90"
        partState="4" spcTests="0"/>
    </structs>
    <items>
      <item name="ToolState" value="0" dataType="2"/>
      <item name="MaterialState" value="0" dataType="2"/>
      <item name="ProcessState" value="0" dataType="2"/>
      <item name="LocationState" value="0" dataType="2"/>
    </items>
    <structArrays>
      <array name="tools">
        <structDef>
          <item name="toolID" dataType="8"/>
          <item name="state" dataType="2"/>
        </structDef>
        <values>
          <item toolID="1002090001" state="0"/>
          <item toolID="786644001" state="0"/>
        </values>
      </array>
      <array name="components">
        <structDef>
          <item name="matID" dataType="8"/>
          <item name="state" dataType="2"/>
        </structDef>
        <values>
          <item matID="98076152430192834465001" state="0"/>
          <item matID="32890234122342318827001" state="0"/>
        </values>
      </array>
      <array name="locationBlocks">
        <structDef>
          <item name="lineNo" dataType="3" />
          <item name="statNo" dataType="3" />
          <item name="statIdx" dataType="3" />

```

```

        <item name="fuNo" dataType="3" />
        <item name="workPos" dataType="3" />
        <item name="toolPos" dataType="3" />
        <item name="code" dataType="3" />
        <item name="info" dataType="8" />
        <item name="source" dataType="8" />
        <item name="sourceToken" dataType="8" />
    </structDef>
    <values/>
</array>
</structArrays>
</body>
</root>

```

Warning limit for tool has been reached:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2321" version="2.3" eventName="partReceived"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="-2">MES detected a warning.</result>
    <trace>
      <trace level="warning" code="800020005" text="Warning level reached"
        source="LineControl"/>
    </trace>
  </event>
  <body>
    <structs>
      <workPart identifier="DMC22934" partForStation="true" changeOver="false"
        workingCode="0" typeNo="111111111" typeVar="" batch="847320"
        nextProcessNo="90" partState="4" spcTests="0"/>
      <insertComponent identifier="IAC27110701" partForStation="true"
        workingCode="0" typeNo="222222222" typeVar="" batch="KXY90011"
        nextProcessNo="90" partState="4" spcTests="0"/>
    </structs>
    <items>
      <item name="ToolState" value="1" dataType="2"/>
      <item name="MaterialState" value="0" dataType="2"/>
      <item name="ProcessState" value="0" dataType="2"/>
      <item name="LocationState" value="0" dataType="2"/>
    </items>
    <structArrays>
      <array name="tools">
        <structDef>
          <item name="toolID" dataType="8"/>
          <item name="state" dataType="2"/>
        </structDef>
        <values>
          <item toolID="1002090001" state="1"/>
          <item toolID="786644001" state="0"/>
        </values>
      </array>
      <array name="components">
        <structDef>
          <item name="matID" dataType="8"/>
          <item name="state" dataType="2"/>
        </structDef>
        <values>
          <item matID="98076152430192834465001" state="0"/>
        </values>
      </array>
    </structArrays>
  </body>
</root>

```

```

        <item matID="32890234122342318827001" state="0"/>
    </values>
</array>
<array name="locationBlocks">
    <structDef>
        <item name="lineNo" dataType="3" />
        <item name="statNo" dataType="3" />
        <item name="statIdx" dataType="3" />
        <item name="fuNo" dataType="3" />
        <item name="workPos" dataType="3" />
        <item name="toolPos" dataType="3" />
        <item name="code" dataType="3" />
        <item name="info" dataType="8" />
        <item name="source" dataType="8" />
        <item name="sourceToken" dataType="8" />
    </structDef>
    <values/>
</array>
</structArrays>
</body>
</root>

```

Tools locked (internal MES processing error):

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
    <header eventId="2321" version="2.3" eventName="partReceived"
        timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
        <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
            processNo="90" processName="PlasmaCleaning" application="PLC"/>
    </header>
    <event>
        <result returnCode="-1">MES detected an error.</result>
        <trace>
            <trace level="error" code="700020012" text="Communication failure with
                service ToolControl (maybe down)." source="LineControl"/>
        </trace>
    </event>
    <body>
        <structs>
            <workPart identifier="DMC22934" partForStation="true" changeOver="false"
                workingCode="0" typeNo="111111111" typeVar="" batch="847320"
                nextProcessNo="90" partState="4" spcTests="0"/>
            <insertComponent identifier=" IAC27110701" partForStation="true"
                workingCode="0" typeNo="222222222" typeVar="" batch="KXY90011"
                nextProcessNo="90" partState="4" spcTests="0"/>
        </structs>
        <items>
            <item name="ToolState" value="-1" dataType="2"/>
            <item name="MaterialState" value="0" dataType="2"/>
            <item name="ProcessState" value="0" dataType="2"/>
            <item name="LocationState" value="0" dataType="2"/>
        </items>
        <structArrays>
            <array name="tools">
                <structDef>
                    <item name="toolID" dataType="8"/>
                    <item name="state" dataType="2"/>
                </structDef>
                <values>
                    <item toolID="1002090001" state="-1"/>
                    <item toolID="786644001" state="-1"/>
                </values>
            </array>
        </structArrays>
    </body>
</root>

```

```

    </values>
  </array>
  <array name="components">
    <structDef>
      <item name="matID" dataType="8"/>
      <item name="state" dataType="2"/>
    </structDef>
    <values>
      <item matID="98076152430192834465001" state="0" />
      <item matID="32890234122342318827001" state="0" />
    </values>
  </array>
  <array name="locationBlocks">
    <structDef>
      <item name="lineNo" dataType="3" />
      <item name="statNo" dataType="3" />
      <item name="statIdx" dataType="3" />
      <item name="fuNo" dataType="3" />
      <item name="workPos" dataType="3" />
      <item name="toolPos" dataType="3" />
      <item name="code" dataType="3" />
      <item name="info" dataType="8" />
      <item name="source" dataType="8" />
      <item name="sourceToken" dataType="8" />
    </structDef>
    <values/>
  </array>
</structArrays>
</body>
</root>

```

Tools locked (lock limit exceeded):

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2321" version="2.3" eventName="partReceived"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="-2">MES detected a warning.</result>
    <trace>
      <trace level="warning" code="800020006" text="Blocking level reached"
        source="LineControl"/>
    </trace>
  </event>
  <body>
    <structs>
      <workPart identifier="DMC22934" partForStation="true" changeOver="false"
        workingCode="0" typeNo="1111111111" typeVar="" batch="847320"
        nextProcessNo="90" partState="4" spcTests="0"/>
      <insertComponent identifier="IAC27110701" partForStation="true"
        workingCode="0" typeNo="2222222222" typeVar="" batch="KXY90011"
        nextProcessNo="90" partState="4" spcTests="0"/>
    </structs>
    <items>
      <item name="ToolState" value="2" dataType="2"/>
      <item name="MaterialState" value="0" dataType="2"/>
      <item name="ProcessState" value="0" dataType="2"/>
      <item name="LocationState" value="0" dataType="2"/>
    </items>
    <structArrays>

```

```

<array name="tools">
  <structDef>
    <item name="toolID" dataType="8"/>
    <item name="state" dataType="2"/>
  </structDef>
  <values>
    <item toolID="1002090001" state="2"/>
    <item toolID="786644001" state="0"/>
  </values>
</array>
<array name="components">
  <structDef>
    <item name="matID" dataType="8"/>
    <item name="state" dataType="2"/>
  </structDef>
  <values>
    <item matID="98076152430192834465001" state="0"/>
    <item matID="32890234122342318827001" state="0"/>
  </values>
</array>
<array name="locationBlocks">
  <structDef>
    <item name="lineNo" dataType="3" />
    <item name="statNo" dataType="3" />
    <item name="statIdx" dataType="3" />
    <item name="fuNo" dataType="3" />
    <item name="workPos" dataType="3" />
    <item name="toolPos" dataType="3" />
    <item name="code" dataType="3" />
    <item name="info" dataType="8" />
    <item name="source" dataType="8" />
    <item name="sourceToken" dataType="8" />
  </structDef>
  <values/>
</array>
</structArrays>
</body>
</root>

```

Warning limit for tool has been reached and material is locked:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2321" version="2.3" eventName="partReceived"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="-2">MES detected a warning.</result>
    <trace>
      <trace level="warning" code="800020007" text="Multiple blocking reasons
        detected." source="LineControl"/>
    </trace>
  </event>
  <body>
    <structs>
      <workPart identifier="DMC22934" partForStation="true" changeOver="false"
        workingCode="0" typeNo="1111111111" typeVar="" batch="847320"
        nextProcessNo="90" partState="4" spcTests="0"/>
      <insertComponent identifier="IAC27110701" partForStation="true"
        workingCode="0" typeNo="2222222222" typeVar="" batch="KXY90011"
        nextProcessNo="90" partState="4" spcTests="0"/>
    </structs>
  </body>
</root>

```

```

</structs>
<items>
  <item name="ToolState" value="1" dataType="2"/>
  <item name="MaterialState" value="2" dataType="2"/>
  <item name="ProcessState" value="0" dataType="2"/>
  <item name="LocationState" value="0" dataType="2"/>
</items>
<structArrays>
  <array name="tools">
    <structDef>
      <item name="toolID" dataType="8"/>
      <item name="state" dataType="2"/>
    </structDef>
    <values>
      <item toolID="1002090001" state="1"/>
      <item toolID="786644001" state="0"/>
    </values>
  </array>
  <array name="components">
    <structDef>
      <item name="matID" dataType="8"/>
      <item name="state" dataType="2"/>
    </structDef>
    <values>
      <item matID="98076152430192834465001" state="2" />
      <item matID="32890234122342318827001" state="0" />
    </values>
  </array>
  <array name="locationBlocks">
    <structDef>
      <item name="lineNo" dataType="3" />
      <item name="statNo" dataType="3" />
      <item name="statIdx" dataType="3" />
      <item name="fuNo" dataType="3" />
      <item name="workPos" dataType="3" />
      <item name="toolPos" dataType="3" />
      <item name="code" dataType="3" />
      <item name="info" dataType="8" />
      <item name="source" dataType="8" />
      <item name="sourceToken" dataType="8" />
    </structDef>
    <values/>
  </array>
</structArrays>
</body>
</root>

```

Workpiece must not be processed (additional component locked, incorrect process):

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2321" version="2.3" eventName="partReceived"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="0"/>
    <trace/>
  </event>
  <body>
    <structs>
      <workPart identifier="DMC22934" partForStation="true" changeOver="false"

```

```

        workingCode="0" typeNo="111111111" typeVar="" batch="847320"
        nextProcessNo="90" partState="4" spcTests="0"/>
<insertComponent identifier=" IAC27110701" partForStation="false"
        workingCode="0" typeNo="222222222" typeVar="" batch="KXY90011"
        nextProcessNo="220" partState="4" spcTests="0"/>
</structs>
<items>
  <item name="ToolState" value="0" dataType="2"/>
  <item name="MaterialState" value="0" dataType="2"/>
  <item name="ProcessState" value="0" dataType="2"/>
  <item name="LocationState" value="0" dataType="2"/>
</items>
<structArrays>
  <array name="tools">
    <structDef>
      <item name="toolID" dataType="8"/>
      <item name="state" dataType="2"/>
    </structDef>
    <values>
      <item toolID="1002090001" state="0"/>
      <item toolID="786644001" state="0"/>
    </values>
  </array>
  <array name="components">
    <structDef>
      <item name="matID" dataType="8"/>
      <item name="state" dataType="2"/>
    </structDef>
    <values>
      <item matID="98076152430192834465001" state="0"/>
      <item matID="32890234122342318827001" state="0"/>
    </values>
  </array>
  <array name="locationBlocks">
    <structDef>
      <item name="lineNo" dataType="3" />
      <item name="statNo" dataType="3" />
      <item name="statIdx" dataType="3" />
      <item name="fuNo" dataType="3" />
      <item name="workPos" dataType="3" />
      <item name="toolPos" dataType="3" />
      <item name="code" dataType="3" />
      <item name="info" dataType="8" />
      <item name="source" dataType="8" />
      <item name="sourceToken" dataType="8" />
    </structDef>
    <values/>
  </array>
</structArrays>
</body>
</root>

```

Setup was locked:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2321" version="2.3" eventName="partReceived"
    timeStamp="2012-12-07T13:28:38.8611+02:00" contentType="3">
    <location lineNo="8030" statNo="40" statIdx="1" fuNo="1" workPos="1"
      toolPos="0" processNo="100" processName="Connector Gel Dispensing"
      application="PLC"/>
  </header>
</event>

```

```
<result returnCode="-2">MES detected a warning.</result>
<trace>
  <trace level="warning" code="800103520" text="Blocking level reached."
    source="LineControl"/>
</trace>
</event>
<body>
  <items>
    <item name="ToolState" value="0" dataType="2"/>
    <item name="MaterialState" value="0" dataType="2"/>
    <item name="ProcessState" value="0" dataType="2"/>
    <item name="LocationState" value="2" dataType="2"/>
  </items>
  <structs>
    <workPart identifier="20121206_133312" partForStation="true"
      changeOver="false" workingCode="0" typeNo="111111111"
      typeVar="" batch="847320" nextProcessNo="90"
      partState="4" spcTests="0"/>
  </structs>
  <structArrays>
    <array name="tools">
      <structDef>
        <item name="toolID" dataType="8"/>
        <item name="state" dataType="2"/>
      </structDef>
      <values>
        <item toolID="1002090001" state="0"/>
        <item toolID="786644001" state="0"/>
      </values>
    </array>
    <array name="components">
      <structDef>
        <item name="matID" dataType="8"/>
        <item name="state" dataType="2"/>
      </structDef>
      <values>
        <item matID="98076152430192834465001" state="0" />
        <item matID="32890234122342318827001" state="0" />
      </values>
    </array>
    <array name="locationBlocks">
      <structDef>
        <item name="lineNo" dataType="3" />
        <item name="statNo" dataType="3" />
        <item name="statIdx" dataType="3" />
        <item name="fuNo" dataType="3" />
        <item name="workPos" dataType="3" />
        <item name="toolPos" dataType="3" />
        <item name="code" dataType="3" />
        <item name="info" dataType="8" />
        <item name="source" dataType="8" />
        <item name="sourceToken" dataType="8" />
      </structDef>
      <values>
        <item lineNo="8030" statNo="40" statIdx="1" fuNo="1" workPos="1"
          toolPos="0" code="123" info="Quality issues"
          source="OIS.NET Portal" sourceToken="MES\Gatekeeper" />
        <item lineNo="8030" statNo="40" statIdx="1" fuNo="1" workPos="1"
          toolPos="0" code="345" info="Special test required"
          source="OIS.NET Portal" sourceToken="MES\Gatekeeper" />
      </values>
    </array>
  </structArrays>
</body>
</root>
```



4.3.1.7 Group Processing Request

Group processing requests are used when n - workpieces need to be processed at once. Typical examples of groups are, for example, panels and carriers - i.e. a set of identifiable workpieces that pass through the process chain as a unit. A group processing request determines the release for the group itself, as well as the status for each individual workpiece within the group. As for individual workpieces, for a processing request, the event `partReceived` must be used. For identification, the ID of the group must be used within the event (e.g. panel edge bar code). The assignment of which individual workpieces are on the transport medium must be known in MES before group requests are created.

While the request for this type of group processing request looks identical to the request for individual processing, the responses are different. In the response to a group processing request, as well as the information already described above, additional information about the individual workpieces is also returned within the transport medium using the array `WorkItems` (e.g. ID, position in the transport medium, processing release).


The following information can currently be transferred using the structure `WorkItems` :

Information	Required	Data type	OpCon Xml	Comments
Position in the transport medium	X	DINT	<code>workItem[]@pos</code>	The position in the transport medium is specified by a sequential number from 1 ... n .
Individual ID	Optional	STRING(80)	<code>workItem[]@id</code>	Identification of the individual workpiece (if available)
Processing release	Optional	BOOL	<code>workItem[]@partForStation</code>	Processing release for the individual workpiece
Product status	Optional	INT	<code>workItem@state</code>	Status of the individual product

 *In future, this structure can also contain other attributes (including the attributes named above - max 10 attributes). The attributes listed above are currently defined by the standard.*

The order in the `workItem` StructArray does not have to correspond to the order of the individual workpieces on the transport medium. For the unique allocation of the processing release, the contents of the attribute `workItem[]@pos` must be used.

Apart from the additional check of the individual releases, in this case the rules described above also apply for client-side processing and interpretation of the confirmation from the server.

 *In general, the processing release is not directly linked to the individual status of the workpieces contained. It is therefore possible that an individual workpiece may not be processed, but the group itself can be. A process must therefore always first check the individual status of the workpiece to be processed, before processing takes place. Conversely, it is also true that if the processing release for the group is not granted, none of the individual workpieces in the group may be processed, regardless of the status of the individual workpiece.*

Example:

Processing release:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2321" version="2.3" eventName="partReceived"
    contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partReceived identifier="DMCY00100" typeNo="1111111111" typeVar=""/>
  </event>
  <body>
    <arrays>
```

```

        <array name="components" dataType="8">
            <item value="98076152430192834465001" />
            <item value="32890234122342318827001" />
        </array>
        <array name="tools" dataType="8">
            <item value="1002090001" />
            <item value="786644001" />
        </array>
    </arrays>
</body>
</root>

```

Processing release: Default case (all workpieces can be processed)

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
    <header eventId="2321" version="2.3" eventName="partReceived"
        timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
        <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
            processNo="90" processName="PlasmaCleaning" application="PLC"/>
    </header>
    <event>
        <result returnCode="0"/>
        <trace/>
    </event>
    <body>
        <structs>
            <workPart identifier="DMCY00100" partForStation="true" changeOver="false"
                workingCode="0" typeNo="111111111" typeVar="" batch="847320"
                nextProcessNo="90" partState="4" spcTests="0"/>
        </structs>
        <items>
            <item name="ToolState" value="0" dataType="2"/>
            <item name="MaterialState" value="0" dataType="2"/>
            <item name="ProcessState" value="0" dataType="2"/>
            <item name="LocationState" value="0" dataType="2"/>
        </items>
        <structArrays>
            <array name="tools">
                <structDef>
                    <item name="toolID" dataType="8"/>
                    <item name="state" dataType="2"/>
                </structDef>
                <values>
                    <item toolID="1002090001" state="0"/>
                    <item toolID="786644001" state="0"/>
                </values>
            </array>
            <array name="components">
                <structDef>
                    <item name="matID" dataType="8"/>
                    <item name="state" dataType="2"/>
                </structDef>
                <values>
                    <item matID="98076152430192834465001" state="0"/>
                    <item matID="32890234122342318827001" state="0"/>
                </values>
            </array>
            <array name="locationBlocks">
                <structDef>
                    <item name="lineNo" dataType="3" />
                    <item name="statNo" dataType="3" />
                    <item name="statIdx" dataType="3" />
                    <item name="fuNo" dataType="3" />
                    <item name="workPos" dataType="3" />
                </structDef>
            </array>
        </structArrays>
    </body>
</root>

```

```

        <item name="toolPos" dataType="3" />
        <item name="code" dataType="3" />
        <item name="info" dataType="8" />
        <item name="source" dataType="8" />
        <item name="sourceToken" dataType="8" />
    </structDef>
    <values/>
</array>
<array name="workItems">
    <structDef>
        <item name="pos" dataType="3"/>
        <item name="partForStation" dataType="11"/>
        <item name="state" dataType="2"/>
    </structDef>
    <values>
        <item pos="1" partForStation="true" state="4"/>
        <item pos="2" partForStation="true" state="4"/>
    </values>
</array>
</structArrays>
</body>
</root>

```

Processing release: Group is locked (incorrect process)

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
    <header eventId="2321" version="2.3" eventName="partReceived"
        timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
        <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
            processNo="90" processName="PlasmaCleaning" application="PLC"/>
    </header>
    <event>
        <result returnCode="0"/>
        <trace/>
    </event>
    <body>
        <structs>
            <workPart identifier="DMCY00100" partForStation="false" changeOver="false"
                workingCode="0" typeNo="111111111" typeVar="" batch="847320"
                nextProcessNo="90" partState="4" spcTests="0"/>
        </structs>
        <items>
            <item name="ToolState" value="0" dataType="2"/>
            <item name="MaterialState" value="0" dataType="2"/>
            <item name="ProcessState" value="0" dataType="2"/>
            <item name="LocationState" value="0" dataType="2"/>
        </items>
        <structArrays>
            <array name="tools">
                <structDef>
                    <item name="toolID" dataType="8"/>
                    <item name="state" dataType="2"/>
                </structDef>
                <values>
                    <item toolID="1002090001" state="0"/>
                    <item toolID="786644001" state="0"/>
                </values>
            </array>
            <array name="components">
                <structDef>
                    <item name="matID" dataType="8"/>
                    <item name="state" dataType="2"/>
                </structDef>
            </array>
        </structArrays>
    </body>
</root>

```

```

        <values>
          <item matID="98076152430192834465001" state="0" />
          <item matID="32890234122342318827001" state="0" />
        </values>
      </array>
      <array name="locationBlocks">
        <structDef>
          <item name="lineNo" dataType="3" />
          <item name="statNo" dataType="3" />
          <item name="statIdx" dataType="3" />
          <item name="fuNo" dataType="3" />
          <item name="workPos" dataType="3" />
          <item name="toolPos" dataType="3" />
          <item name="code" dataType="3" />
          <item name="info" dataType="8" />
          <item name="source" dataType="8" />
          <item name="sourceToken" dataType="8" />
        </structDef>
        <values/>
      </array>
      <array name="workItems">
        <structDef>
          <item name="pos" dataType="3"/>
          <item name="partForStation" dataType="11"/>
          <item name="state" dataType="2"/>
        </structDef>
        <values>
          <item pos="1" partForStation="true" state="4"/>
          <item pos="2" partForStation="true" state="4"/>
        </values>
      </array>
    </structArrays>
  </body>
</root>

```

Processing release: Individual workpiece locked, group can be processed

In this case, all workpieces belonging to the group, apart from workpieces with the attribute `partForStation = false`, can be processed.

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2321" version="2.3" eventName="partReceived"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="0"/>
    <trace/>
  </event>
  <body>
    <structs>
      <workPart identifier="DMCY00100" partForStation="true" changeOver="false"
        workingCode="0" typeNo="111111111" typeVar="" batch="847320"
        nextProcessNo="90" partState="4" spcTests="0"/>
    </structs>
    <items>
      <item name="ToolState" value="0" dataType="2"/>
      <item name="MaterialState" value="0" dataType="2"/>
      <item name="ProcessState" value="0" dataType="2"/>
      <item name="LocationState" value="0" dataType="2"/>
    </items>
  </body>
</root>

```

```

<array name="tools">
  <structDef>
    <item name="toolID" dataType="8"/>
    <item name="state" dataType="2"/>
  </structDef>
  <values>
    <item toolID="1002090001" state="0"/>
    <item toolID="786644001" state="0"/>
  </values>
</array>
<array name="components">
  <structDef>
    <item name="matID" dataType="8"/>
    <item name="state" dataType="2"/>
  </structDef>
  <values>
    <item matID="98076152430192834465001" state="0"/>
    <item matID="32890234122342318827001" state="0"/>
  </values>
</array>
<array name="locationBlocks">
  <structDef>
    <item name="lineNo" dataType="3" />
    <item name="statNo" dataType="3" />
    <item name="statIdx" dataType="3" />
    <item name="fuNo" dataType="3" />
    <item name="workPos" dataType="3" />
    <item name="toolPos" dataType="3" />
    <item name="code" dataType="3" />
    <item name="info" dataType="8" />
    <item name="source" dataType="8" />
    <item name="sourceToken" dataType="8" />
  </structDef>
  <values/>
</array>
<array name="workItems">
  <structDef>
    <item name="pos" dataType="3"/>
    <item name="partForStation" dataType="11"/>
    <item name="state" dataType="2"/>
  </structDef>
  <values>
    <item pos="1" partForStation="true" state="4"/>
    <item pos="2" partForStation="false" state="5"/>
  </values>
</array>
</structArrays>
</body>
</root>

```

4.3.2 Starting to Process the Workpiece

When processing of the workpiece begins, the process must send the event `partProcessingStarted` to MES. After receiving the message, MES does not send an acknowledgment to the process.

Within this event, in addition to the data specified above, the process can also send the following additional data:

Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>

```

```

<header eventId="2323" version="2.3" eventName="partProcessingStarted"
        contentType="3">
  <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
            processNo="90" processName="PlasmaCleaning" application="PLC"/>
</header>
<event>
  <partProcessingStarted identifier="DMC22934"/>
</event>
<body/>
</root>

```

4.3.3 Canceling Processing

If processing of the workpiece is canceled, the system must check if the station can further evaluate the processing result. If this is not the case (e.g. on OpCon-Clients: Emergency stop, mode changes, etc.) then the process must send `partProcessingAborted` to MES.

MES does not send an acknowledgment to the station after receiving the message.

If an evaluation can still take place, a `partProcessed` with the `result=Aborted` (including processing result) must be sent.

Within this event, in addition to the data specified in Part I, the following additional data can also be sent:

Information	Required	Data type	OpCon Xml	Comments
Reason for cancelation	Optional	STRING	<code>body.abortText</code>	-
Processing basic data	Optional	udt	<code>resHead</code>	The definitions from part II apply for the required data of <code>resHead</code> .

Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2325" version="2.3" eventName="partProcessingAborted"
        contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
            processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partProcessingAborted identifier="DMC22934"/>
  </event>
  <body>
    <structs>
      <resHead result="3" typeNo="1111111111" typeVar="0815" workingCode="0"
              nioBits="16"/>
    </structs>
  </body>
</root>

```

4.3.4 Processing Complete

The processing end of a process must be communicated to MES by sending the event `partProcessed`. In addition to the basic processing data and the processing result (e.g. process result, NOK bits), the following information can also be transferred within the event:

- Trace data for the installed components
- Process failures

- Trace data for process values (e.g. measurement values)
- SPC test results
- Special information

💡 *In the transfer of processing results, the complete location must always be entered. The use of '0' for a part of the location (e.g. tool position) is not permitted.*

After the workpiece has been processed by the process, the event `partProcessed` must be sent by the station. The result data can also be transferred to the line computer. MES normally sends an acknowledgment to the station after receiving the message. However, if data that requires archiving is processed, MES can send a response to the station.

For stations that can report their processing results in segments (e.g. several processing steps are performed and one result must be written for each processing step), these partial processing results can also be sent to MES with a `dataUploadRequired` event. Details of this are described in .

4.3.4.1 Individual Request

4.3.4.1.1 Basic Data

The basic data is transferred in the `resHead` element. This data includes the general statements about the end of processing (for example, whether the overall processing was considered successful by the process).

💡 *The `resHead` element is described in part II.*

The use of the optional attributes of the `resHead` element must be defined specifically for each plant. The use of the two elements `resHead.workCycleCount` and `resHead.cycleTimePrev` is not foreseen in the standard case.

4.3.4.1.2 Processing Result

When reporting the end of processing, each process must also send a processing result. This consists of an overall evaluation of the process execution (e.g. "process executed successfully" or "process could not be executed successfully", etc.) and details of any process errors that have occurred. For an individual workpiece, this information is sent within the element `resHead` in the attributes `result` and `nioBits` . Part II describes the exact use of these fields.

Example:

The simplest feedback of the end of processing contains only the element `resHead` for a single workpiece. This enables the process to be included in the production process within MES.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.3" eventName="partProcessed"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partProcessed identifier="DMC22934"/>
  </event>
  <body>
    <structs>
      <resHead result="1" typeNo="111111111" typeVar="0815"
        workingCode="0" nioBits="0"/>
    </structs>
  </body>
</root>
```

4.3.4.1.3 Process failures

If a workpiece was not processed successfully, the cause of the error must be reported to the MES. In addition to the option described above of reporting the cause of the error using resHead.nioBits, the Array errors can also be used to report the list of error causes.

Information	Required	Data type	OpCon Xml	Comments
Cause of error	Mandatory	STRING	errors[]@name	Name of the cause of error
Error type	Optional	INT	errors[]@errType	Classification of the error; the following values are allowed. Currently a '2' must always be sent.
Error number	Optional	STRING(20)	errors[]@errNumber	Alphanumeric error code
Error details	Optional	STRING	errors[]@errInfo	More detailed description of the error, or environment in which the error occurred.

Example

The simplest feedback of the end of processing only contains the element resHead for an individual workpiece. This enables the process to be included in the production process within MES.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.3" eventName="partProcessed"
    contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partProcessed identifier="DMC22934"/>
  </event>
  <body>
    <structs>
      <resHead result="2" typeNo="111111111" typeVar="0815"
        workingCode="0" nioBits="0"/>
    </structs>
    <structArrays>
      <array name="errors">
        <structDef>
          <item name="name" dataType="8"/>
          <item name="errType" dataType="2"/>
          <item name="errNumber" dataType="8"/>
          <item name="errInfo" dataType="8"/>
        </structDef>
        <values>
          <item name="" errType="2" errNumber="40298" errInfo=""/>
        </values>
      </array>
    </structArrays>
  </body>
</root>
```

4.3.4.1.4 Trace Data of the Installed Materials

If a process installs materials, at the end of processing, these must be reported to MES for documentation in the trace system.

💡 If a process installs materials and does not report these at the end of processing, these materials will also not be posted to the workpiece and cannot be traced.

The material data is transferred in the element `matData` as a structure array. A single entry can contain the following information:

Information	Required	Data type	OpCon Xml	Comments
ID of the material	Optional	STRING	<code>matData[]@matID</code>	The material identifier is typically the MatID (see MatLabel specification).
Remaining quantity or consumed quantity of the material	Optional	REAL	<code>matData[]@qty</code>	In addition to discrete components such as resistors and capacitors, it must also be possible to enter volume materials (e.g. adhesives). In this case, the consumed quantity is typically reported. Note that the interpretation of the field (remaining quantity or quantity consumed) must be defined between the station and MES on a project-specific basis.
ID of an A component	Optional	STRING(80)	<code>matData[]@identifier</code>	A-components must always be uniquely identified. This ID is different from the ID of the material, which identifies the batch or the packaging unit.
Model part number of the material	Optional	STRING(10)	<code>matData[]@typeNo</code>	Type part number under which the material is managed at Bosch.
Material batch	Optional	STRING	<code>matData[]@batch</code>	Manufacturer's material batch number.
Manufacturer of the material	Optional	STRING	<code>matData[]@vendor</code>	The batch number is unique per manufacturer, but is not valid across manufacturers. Since the model part number can also be used for various materials in some cases, the manufacturer must be transferred to enable unique identification without a Mat ID.
Material index number	Optional	DINT	<code>matData[]@idx</code>	Required to report large quantities of component data (e.g. at component insertion machines).

💡 The (remaining) quantity of a material may always only be specified in the basic unit. The following basic units are currently allowed:

Length: meters (MTR)

Surface area: square meters

Weight: kilograms (KGM)

Quantity: units (NAR)

Volumes: liters (LTR)

💡 In the default case, the material must always be identified using the Mat ID (`matData[]@matID`). For plants that do not have this information, or if no `Mat-DB` is available in the plant, the material must be identified using the triple properties `{type part number (matData[]@typeNo), Batch(matData[]@batch), Manufacturer matData[]@vendor }`. In this scenario, not all functions of the MES may be available.

💡 Not all information has to be sent on all occasions. If material is to be posted, only the ID of the material must be sent, using either `MAT-ID` or `{TTNR;Charge;Hersteller}` (see note above). If structure arrays are used, a maximum of 10 items of information about the material can be transferred. As such, you need to make a sensible selection here.

Example:

In order to enter the installed material, the respective material IDs must be sent to the MES. In this example, two materials were installed. The details about the material are already stored in MES. The materials can be defined either as user-defined arrays or as structure arrays.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.3" eventName="partProcessed" contentType="3">
    <location lineNo="1" statNo="10" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="5" processName="SMT" application="PanaCIM"/>
  </header>
  <event>
    <partProcessed identifier="DMC22934"/>
  </event>
  <body>
    <structs>
      <resHead result="1" typeNo="111111111" typeVar="0815" workingCode="0"
nioBits="0"/>
    </structs>
    <structArrays>
      <array name="matData">
        <structDef>
          <item name="idx" dataType="3"/>
          <item name="matID" dataType="8"/>
          <item name="qty" dataType="5"/>
        </structDef>
        <values>
          <item idx="1" matID="884566S123456789055" qty="2889.00"/>
          <item idx="2" matID="303479S234234234777" qty="1334.00"/>
        </values>
      </array>
    </structArrays>
  </body>
</root>
```

4.3.4.1.5 Process Data

Process data contains measured values recorded as the process is implemented, and which are to be further processed or saved in the trace system.

💡 *Through the use of modern measuring and test technology, very large quantities of data can be produced as a process is implemented. However, it is not always practical to further process all data in the higher-level system (e.g. MES). We therefore recommend you carefully pre-select the data so the process is effectively monitored.*

The process data is transferred in the elements `processRealData`, `processIntData` and `processStringData` as structure arrays. A single entry can contain the following information:

processRealData:

Information	Required	Data type	OpCon Xml	Comments
Position in use (groups)	Optional	DINT	processRealData[]@pos	Only relevant for group requests (see below).
Name of the measuring point	X	STRING	processRealData[]@name	The name of the measuring point must be unique within the process.
Measured value	Optional	REAL	processRealData[]@value	-
Test result	Optional	INT	processRealData[]@result	-
Lower limit value	Optional	REAL	processRealData[]@LoLim	-

Upper limit value	Optional	REAL	processRealData[]@upLim	-
Set value	Optional	REAL	processRealData[]@setValue	-
Test type	Optional	INT	processRealData[]@checkType	-
Unit of measured value	Optional	STRING	processRealData[]@unit	-
PAA relevance class	Optional	DINT	processRealData[]@paaRel	Relevance class for PAA: <ul style="list-style-type: none"> 0 = PAA not relevant 1...9 = PAA relevance class

processIntData:

Information	Required	Data type	OpCon Xml	Comments
Position in use (groups)	Optional	DINT	processIntData[]@pos	Only relevant for group requests (see below).
Name of the measuring point	X	STRING	processIntData[]@name	The name of the measuring point must be unique within the process.
Measured value	Optional	REAL	processIntData[]@value	-
Test result	Optional	INT	processIntData[]@result	-
Lower limit value	Optional	REAL	processIntData[]@LoLim	-
Upper limit value	Optional	REAL	processIntData[]@upLim	-
Set value	Optional	REAL	processIntData[]@setValue	-
Test type	Optional	INT	processIntData[]@checkType	-
Unit of measured value	Optional	STRING	processIntData[]@unit	-
PAA relevance class	Optional	DINT	processIntData[]@paaRel	Relevance class for PAA: <ul style="list-style-type: none"> 0 = PAA not relevant 1...9 = PAA relevance class

processStrData:

Information	Required	Data type	OpCon Xml	Comments
Position in use (groups)	Optional	DINT	processStrData[]@pos	Only relevant for group requests (see below).
Name of the measuring point	X	STRING	processStrData[]@name	The name of the measuring point must be unique within the process.
Measured value	Optional	REAL	processStrData[]@value	-
Test result	Optional	INT	processStrData[]@result	-
Set value	Optional	REAL	processStrData[]@setValue	-
Test type	Optional	INT	processStrData[]@checkType	-
Unit of measured value	Optional	STRING	processStrData[]@unit	-

PAA relevance class	Optional	DINT	processStrData [] @ paaRel	Relevance class for PAA: <ul style="list-style-type: none"> 0 = PAA not relevant 1...9 = PAA relevance class
------------------------	-----------------	------	-------------------------------	--

Example:

Based on its data types, the process data must be divided into the relevant processXXXXataarrays. In this example, two REAL and one INT values are transferred.

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.3" eventName="partProcessed"
    contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partProcessed identifier="DMC22934"/>
  </event>
  <body>
    <structs>
      <resHead result="1" typeNo="1111111111" typeVar="" workingCode="0"
nioBits="0"/>
    </structs>
    <structArrays>
      <array name="processRealData">
        <structDef>
          <item name="name" dataType="8"/>
          <item name="value" dataType="5"/>
          <item name="result" dataType="2"/>
          <item name="loLim" dataType="5"/>
          <item name="upLim" dataType="5"/>
          <item name="setValue" dataType="5"/>
          <item name="checkType" dataType="2"/>
          <item name="unit" dataType="8"/>
          <item name="paaRel" dataType="3"/>
        </structDef>
        <values>
          <item name="Torque" value="67.489998" result="1" loLim="60.000000"
            upLim="70.000000" setValue="65.000000" checkType="6" unit="Nm"
            paaRel="5"/>
          <item name="Angle" value="14.750000" result="1" loLim="13.500000"
            upLim="15.500000" setValue="15.000000" checkType="6" unit="Deg"
            paaRel="0"/>
        </values>
      </array>
      <array name="processIntData">
        <structDef>
          <item name="name" dataType="8"/>
          <item name="value" dataType="3"/>
          <item name="result" dataType="2"/>
          <item name="loLim" dataType="3"/>
          <item name="upLim" dataType="3"/>
          <item name="setValue" dataType="3"/>
          <item name="checkType" dataType="2"/>
          <item name="unit" dataType="8"/>
          <item name="paaRel" dataType="3"/>
        </structDef>
        <values>
          <item name="Force" value="153" result="1" loLim="150" upLim="155"
            setValue="151" checkType="6" unit="N" paaRel="0"/>
        </values>
      </array>
    </structArrays>
  </body>
</root>

```

```

    </structArrays>
  </body>
</root>

```

4.3.4.1.6 Tools

If tools to be monitored by MES are used within the process, these tools must be sent within the `partProcessed` event. The tool data is transferred in the element `toolData` as a structure array. A single entry can contain the following information:

Information	Required	Data type	OpCon Xml	Comments
Identifying the tool	X	STRING	<code>toolData[]@toolID</code>	-
Number of uses / consumption	Optional	DINT	<code>toolData[]@consumption</code>	The number always refers to the current use, it does not have to be counted consecutively. If a stamp is used twice per tool, then for workpiece 1 <code>consumption = 2</code> is transferred, and also for workpiece 2 <code>consumption = 2</code> .
Number of NOK strokes	Optional	DINT	<code>toolData[]@consumptionNio</code>	Number of NOK strokes (also included in the number of completed strokes)

From an MES perspective, tool carriers (WPC = Work Part Carriers) are also tools. If workpiece carriers are to be monitored, the ID of the workpiece carrier must be included in the `toolData[]` array.

Example:

The two tools `1002090001` and `786644001` are reported with their respective strokes required for the execution of the process.

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.3" eventName="partProcessed" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partProcessed identifier="DMC22934"/>
  </event>
  <body>
    <structs>
      <resHead result="1" typeNo="111111111" typeVar="" workingCode="0"
nioBits="0"/>
    </structs>
    <structArrays>
      <array name="toolData">
        <structDef>
          <item name="toolID" dataType="8"/>
          <item name="consumption" dataType="3"/>
          <item name="consumptionNio" dataType="3"/>
        </structDef>
        <values>
          <item toolID="1002090001" consumption="3" consumptionNio="0"/>
          <item toolID="786644001" consumption="3" consumptionNio="0"/>
        </values>
      </array>
    </structArrays>
  </body>
</root>

```

If during processing with the tool 1002090001 a faulty stroke is performed, the total number of uses increases by the number of faulty uses:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.3" eventName="partProcessed" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partProcessed identifier="DMC22934"/>
  </event>
  <body>
    <structs>
      <resHead result="1" typeNo="111111111" typeVar="" workingCode="0"
nioBits="0"/>
    </structs>
    <structArrays>
      <array name="toolData">
        <structDef>
          <item name="toolID" dataType="8"/>
          <item name="consumption" dataType="3"/>
          <item name="consumptionNio" dataType="3"/>
        </structDef>
        <values>
          <item toolID="1002090001" consumption="4" consumptionNio="1"/>
          <item toolID="786644001" consumption="3" consumptionNio="0"/>
        </values>
      </array>
    </structArrays>
  </body>
</root>
```

4.3.4.1.7 SPC test results

For the transmission of SPC test results, the array spcResults must be used.

Information	Required	Data type	OpCon Xml	Comments
SPC test program number	Mandatory	DINT	spcResults[] @testNo	The SPC test program number corresponds to the numerical value of the bit set in spcTests (see processing release).
SPC test result	Mandatory	INT	spcResults[] @testResult	This value reports the result of the SPC test to the station. If the test could not be carried out directly on the station, but that part was rejected for the test, this can also be reported in the result. Only the following values are valid as a result: 0 -> Test is reported back later 1 -> Test passed 2 -> Test not passed
Note whether the workpiece is to be locked	Optional	BOOL	spcResults[] @blockParts	Reserved for future extensions in asynchronous results reporting. Not currently evaluated.
ID of the assigned SPC session	Optional	STRING	spcResults[] @sessionID	Only necessary for the asynchronous feedback of test results.

In the processing request, the station received the request to perform SPC tests 1 and 8. The tests were performed, and test 1 was successful, but test 8 failed. In the subsequent partProcessed request, the corresponding results are sent to MES:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.3" eventName="partProcessed" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partProcessed identifier="DMC22934"/>
  </event>
  <body>
    <structs
      <resHead result="1" typeNo="111111111" typeVar="" workingCode="0"
nioBits="0"/>
    </structs>
    <structArrays>
      <array name="spcResults">
        <structDef>
          <item name="testNo" dataType="3"/>
          <item name="testResult" dataType="2"/>
        </structDef>
        <values>
          <item testNo="1" testResult="1" />
          <item testNo="8" testResult="2" />
        </values>
      </array>
      <array name="toolData">
        <structDef>
          <item name="toolID" dataType="8"/>
          <item name="consumption" dataType="3"/>
          <item name="consumptionNio" dataType="3"/>
        </structDef>
        <values>
          <item toolID="1002090001" consumption="3" consumptionNio="0"/>
          <item toolID="786644001" consumption="3" consumptionNio="0"/>
        </values>
      </array>
    </structArrays>
  </body>
</root>
```

4.3.4.1.8 Special Information

Besides the data described above, there is also a series of information which cannot be transmitted using the elements described above. Such information is sent as individual variables within the `items` element.

The names for these individual values are constructed according to the following syntax:

- [Category][Sequential number][.][Designator]

where the individual elements mean the following:

- [Category]: Name of the category (e.g. "Prog" for program information)
- [Sequential number]: 1-based sequential number in each category

- [.]: Separator
- [Designator]: Name of the information (e.g. "Version to designate a version)

4.3.4.1.8.1 Category: Programs

This category describes the programs used by the process (e.g. test, AOI, or NC programs).

Information	Mandatory	Data Type	OpCon Xml	Comment
Name of the program	X	STRING	item.ProgX.Name	e.g. AOI program
Version of the program	optional	STRING	item.ProgX.Version	-

4.3.4.1.8.2 Category: Container

This category describes a box, a magazine or general container in which the workpieces are packed.

Information	Required	Data type	OpCon Xml	Comments
ID of the container	X	STRING	item.PackageX.ID	E.g. box number

Example:

Individual request:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.3" eventName="partProcessed"
    contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partProcessed identifier="DMC22934"/>
  </event>
  <body>
    <structs>
      <resHead result="3" typeNo="111111111" typeVar="" workingCode="0"
        nioBits="16"/>
    </structs>
    <items>
      <item name="Prog1.Name" value="PRG_MILL_111111111" dataType="8"/>
      <item name="Prog1.SubSystem" value="BL922" dataType="8"/>
      <item name="Prog2.Name" value="NC808098" dataType="8"/>
      <item name="Prog2.SubSystem" value="XT2" dataType="8"/>
    </items>
  </body>
</root>
```

4.3.4.2 Group Request

If the processing result is sent for an entire group, the same determinations are initially given as for **individual requests** (see page 55). The following differences and/or extensions must be taken into consideration in a group request:

4.3.4.2.1 Basic Data

The content of the `resHead` element refers to the complete group (e.g. harness or boot). The result is that a group can only contain workpieces whose basic data (type part number, lot, order number, etc.) is identical.

4.3.4.2.2 Processing Result

The process must send a processing result for both the group and the individual workpieces. The processing result for the group is sent in the element `resHead` in the attributes `result` and `nioBits`.

The processing results for the individual workpieces are sent in the element `results` as a structure array. A single entry can contain the following information:

Information	Required	Data type	OpCon Xml	Comments
Position within the group	X	DINT	<code>results[]@pos</code>	Describes the position in the group (e.g. in use).
Processing result of the individual workpiece	X	INT	<code>results[]@result</code>	The processing result is transferred as numeric. The allowed values and their description correspond to those of <code>resHead.result</code> .
Process error for individual workpiece	X	DINT	<code>results[]@nioBits</code>	The process errors for the individual workpiece are transferred with bit encoding. For details, refer to <code>resHead.nioBits</code> .
ID of the individual workpiece	Optional	STRING	<code>result[]@identifier</code>	The ID must be specified if there is not yet a connection between the group ID and the position numbers of the individual workpieces.

Example:

In this example, the successful processing of six uses is reported to MES. All six individual switch processes were performed successfully.

In the first process (e.g. laser engraving), the IDs of the individual workpieces must also be included:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.3" eventName="partProcessed"
    contentType="3">
    <location lineNo="1" statNo="10" statIdx="1" fuNo="1" toolPos="1" workPos="1"
      processNo="10" processName="DMC Laser" application="IPC" />
  </header>
  <event>
    <partProcessed identifier="DMC22934"/>
  </event>
  <body>
    <structs>
      <resHead result="1" typeNo="1111111111" typeVar="" workingCode="0"
        nioBits="0"/>
    </structs>
    <structArrays>
      <array name="results">
        <structDef>
          <item name="pos" dataType="3"/>
          <item name="result" dataType="2"/>
          <item name="nioBits" dataType="3"/>
          <item name="identifier" dataType="8"/>
        </structDef>
        <values>
          <item pos="1" result="1" nioBits="0" identifier="DMC2293401"/>
          <item pos="2" result="1" nioBits="0" identifier="DMC2293402"/>
          <item pos="3" result="1" nioBits="0" identifier="DMC2293403"/>
          <item pos="4" result="1" nioBits="0" identifier="DMC2293404"/>
          <item pos="5" result="1" nioBits="0" identifier="DMC2293405"/>
        </values>
      </array>
    </structArrays>
  </body>
</root>
```

```

        <item pos="6" result="1" nioBits="0" identifier="DMC2293406"/>
    </values>
</array>
</structArrays>
</body>
</root>

```

All subsequent processes can then report their processing results without specifying the ID of the individual workpiece.

Processing result as a structure array:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.3" eventName="partProcessed"
    contentType="3">
    <location lineNo="1" statNo="10" statIdx="1" fuNo="1" toolPos="1" workPos="1"
      processNo="5" processName="SMT" application="PanaCIM" />
  </header>
  <event>
    <partProcessed identifier="DMC22934"/>
  </event>
  <body>
    <structs>
      <resHead result="1" typeNo="1111111111" typeVar="" workingCode="0"
        nioBits="0"/>
    </structs>
    <structArrays>
      <array name="results">
        <structDef>
          <item name="pos" dataType="3"/>
          <item name="result" dataType="2"/>
          <item name="nioBits" dataType="3"/>
        </structDef>
        <values>
          <item pos="1" result="1" nioBits="0"/>
          <item pos="2" result="1" nioBits="0"/>
          <item pos="3" result="1" nioBits="0"/>
          <item pos="4" result="1" nioBits="0"/>
          <item pos="5" result="1" nioBits="0"/>
          <item pos="6" result="1" nioBits="0"/>
        </values>
      </array>
    </structArrays>
  </body>
</root>

```

Individual switch processes can be reported as faulty without the whole use being faulty. In this example, the individual switch process to Position 3 failed:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.3" eventName="partProcessed"
    contentType="3">
    <location lineNo="1" statNo="10" statIdx="1" fuNo="1" toolPos="1" workPos="1"
      processNo="5" processName="SMT" application="PanaCIM" />
  </header>
  <event>
    <partProcessed identifier="DMC22934"/>
  </event>
  <body>
    <structs>

```

```

        <resHead result="1" typeNo="111111111" typeVar="" workingCode="0"
            nioBits="0"/>
    </structs>
    <structArrays>
        <array name="results">
            <structDef>
                <item name="pos" dataType="3"/>
                <item name="result" dataType="2"/>
                <item name="nioBits" dataType="3"/>
            </structDef>
            <values>
                <item pos="1" result="1" nioBits="0" />
                <item pos="2" result="1" nioBits="0" />
                <item pos="3" result="2" nioBits="1025" />
                <item pos="4" result="1" nioBits="0" />
                <item pos="5" result="1" nioBits="0" />
                <item pos="6" result="1" nioBits="0" />
            </values>
        </array>
    </structArrays>
</body>
</root>

```

4.3.4.2.3 Trace Data of the Installed Materials

As for the individual requests, the installed materials are transferred using the element `matData` . If no further information is sent, the material is assigned to all individual workpieces. If, on the other hand, you want to perform an explicit allocation, in addition to `matData` , the element `comp` must also be transferred as a structure array. A single entry can contain the following information:

Information	Required	Data type	OpCon Xml	Comments
Position within the group	X	DINT	<code>comp[]@pos</code>	Describes the position in the group (e.g. in use).
Material reference	X	DINT	<code>comp[]@ref</code>	Refers to the entry within <code>matData</code> , for which the following is valid: <code>matData[]@idx == comp[]@ref</code>
ID of the component	Optional	STRING	<code>comp[]@refDes</code>	ID for the component on the individual workpiece (e.g. R100 or CON1)

Example:

In this example, the installed materials are used six times and then reported to MES. All six individual switch processes were processed successfully.

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
    <header eventId="2326" version="2.3" eventName="partProcessed" contentType="3">
        <location lineNo="1" statNo="10" statIdx="1" fuNo="1" toolPos="1" workPos="1"
            processNo="5" processName="SMT" application="PanaCIM" />
    </header>
    <event>
        <partProcessed identifier="DMC22934"/>
    </event>
    <body>
        <structArrays>
            <array name="matData">
                <structDef>
                    <item name="idx" dataType="3"/>
                    <item name="matID" dataType="8"/>
                    <item name="qty" dataType="5"/>
                </structDef>
                <values>

```

```

        <item idx="1" matID="884566S123456789055" qty="2889.00"/>
        <item idx="1" matID="884566S123456789055" qty="1334.00"/>
    </values>
</array>
<array name="comp">
    <structDef>
        <item name="pos" dataType="3"/>
        <item name="ref" dataType="3"/>
        <item name="refDes" dataType="8"/>
    </structDef>
    <values>
        <item pos="1" ref="1" refDes="R100"/>
        <item pos="1" ref="1" refDes="R101"/>
        <item pos="1" ref="1" refDes="R102"/>
        <item pos="1" ref="2" refDes="C10"/>
        <item pos="1" ref="2" refDes="C20"/>
        <item pos="2" ref="1" refDes="R100"/>
        <item pos="2" ref="1" refDes="R101"/>
        <item pos="2" ref="1" refDes="R102"/>
        <item pos="2" ref="2" refDes="C10"/>
        <item pos="2" ref="2" refDes="C20"/>
        <item pos="3" ref="1" refDes="R100"/>
        <item pos="3" ref="1" refDes="R101"/>
        <item pos="3" ref="1" refDes="R102"/>
        <item pos="3" ref="2" refDes="C10"/>
        <item pos="3" ref="2" refDes="C20"/>
        <item pos="4" ref="1" refDes="R100"/>
        <item pos="4" ref="1" refDes="R101"/>
        <item pos="4" ref="1" refDes="R102"/>
        <item pos="4" ref="2" refDes="C10"/>
        <item pos="4" ref="2" refDes="C20"/>
        <item pos="5" ref="1" refDes="R100"/>
        <item pos="5" ref="1" refDes="R101"/>
        <item pos="5" ref="1" refDes="R102"/>
        <item pos="5" ref="2" refDes="C10"/>
        <item pos="5" ref="2" refDes="C20"/>
        <item pos="6" ref="1" refDes="R100"/>
        <item pos="6" ref="1" refDes="R101"/>
        <item pos="6" ref="1" refDes="R102"/>
        <item pos="6" ref="2" refDes="C10"/>
        <item pos="6" ref="2" refDes="C20"/>
    </values>
</array>
<array name="results">
    <structDef>
        <item name="pos" dataType="3"/>
        <item name="result" dataType="2"/>
        <item name="nioBits" dataType="3"/>
    </structDef>
    <values>
        <item pos="1" result="1" nioBits="0" />
        <item pos="2" result="1" nioBits="0" />
        <item pos="3" result="1" nioBits="0" />
        <item pos="4" result="1" nioBits="0" />
        <item pos="5" result="1" nioBits="0" />
        <item pos="6" result="1" nioBits="0" />
    </values>
</array>
</structArrays>
</body>
</root>

```

4.3.4.2.4 Process Data

In a group request, process data can be sent that is the same for all individual workpieces within the group, as well as data that is specific for individual workpieces.

Process data that applies for the whole group, is transferred in the elements `panelRealData`, `panelIntData` and `panelStrData` as a structure array or user-defined array. A single entry can contain the following information:

panelRealData:

Information	Required	Data type	OpCon Xml	Comments
Position in utility (group)	optional	DINT	<code>panelRealData[]@pos</code>	Only relevant for group-requests (see below)
Name of the measuring point	X	STRING	<code>panelRealData[]@name</code>	The name of the measuring point must be unique within the process.
Measured value	Optional	REAL	<code>panelRealData[]@value</code>	-
Test result	Optional	INT	<code>panelRealData[]@result</code>	-
Lower limit value	Optional	REAL	<code>panelRealData[]@loLim</code>	-
Upper limit value	Optional	REAL	<code>panelRealData[]@upLim</code>	-
Set value	Optional	REAL	<code>panelRealData[]@setValue</code>	-
Test type	Optional	INT	<code>panelRealData[]@checkType</code>	-
Unit of measured value	Optional	STRING	<code>panelRealData[]@unit</code>	-
PAA relevance class	Optional	DINT	<code>panelRealData[]@paaRel</code>	Relevance class for PAA: 0 = PAA not relevant 1...9 = PAA relevance class

panelIntData:

Information	Required	Data type	OpCon Xml	Comments
Position in utility (group)	optional	DINT	<code>panelIntData[]@pos</code>	Only relevant for group-requests (see below)
Name of the measuring point	X	STRING	<code>panelIntData[]@name</code>	The name of the measuring point must be unique within the process.
Measured value	Optional	REAL	<code>panelIntData[]@value</code>	-
Test result	Optional	INT	<code>panelIntData[]@result</code>	-
Lower limit value	Optional	REAL	<code>panelIntData[]@loLim</code>	-
Upper limit value	Optional	REAL	<code>panelIntData[]@upLim</code>	-
Set value	Optional	REAL	<code>panelIntData[]@setValue</code>	-
Test type	Optional	INT	<code>panelIntData[]@checkType</code>	-
Unit of measured value	Optional	STRING	<code>panelIntData[]@unit</code>	-
PAA relevance class	Optional	DINT	<code>panelIntData[]@paaRel</code>	Relevance class for PAA: 0 = PAA not relevant 1...9 = PAA relevance class

panelStrData:

Information	Required	Data type	OpCon Xml	Comments
Position in utility (group)	optional	DINT	panelStrData [] @ pos	Only relevant for group-requests (see below)
Name of the measuring point	X	STRING	panelStrData [] @ name	The name of the measuring point must be unique within the process.
Measured value	Optional	REAL	panelStrData [] @ value	-
Test result	Optional	INT	panelStrData [] @ result	-
Set value	Optional	REAL	panelStrData [] @ setValue	-
Test type	Optional	INT	panelStrData [] @ checkType	-
Unit of measured value	Optional	STRING	panelStrData [] @ unit	-
PAA relevance class	Optional	DINT	panelStrData [] @ paaRel	Relevance class for PAA: 0 = PAA not relevant 1...9 = PAA relevance class

For process data that is workpiece-specific, within processData the attribute pos must strictly be set for all entries.

Example:

In the following example, a global process parameter Temperature is sent and for the readings, the relevant measured values Torque, Angle and Force.

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.3" eventName="partProcessed" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partProcessed identifier="DMC22934"/>
  </event>
  <body>
    <structs>
      <resHead result="1" typeNo="111111111" typeVar="" workingCode="0"
nioBits="0"/>
    </structs>
    <structArrays>
      <array name="panelRealData">
        <structDef>
          <item name="name" dataType="8"/>
          <item name="value" dataType="5"/>
          <item name="result" dataType="2"/>
          <item name="loLim" dataType="5"/>
          <item name="upLim" dataType="5"/>
          <item name="setValue" dataType="5"/>
          <item name="checkType" dataType="2"/>
          <item name="unit" dataType="8"/>
          <item name="paaRel" dataType="3"/>
        </structDef>
        <values>
          <item name="Temperature" value="68.345000" result="1" loLim="60.000000"
upLim="70.000000"
          setValue="65.000000" checkType="6" unit="C" paaRel="5"/>
        </values>
      </array>
      <array name="processRealData">
        <structDef>
          <item name="pos" dataType="3"/>
          <item name="name" dataType="8"/>

```

```

        <item name="value" dataType="5"/>
        <item name="result" dataType="2"/>
        <item name="loLim" dataType="5"/>
        <item name="upLim" dataType="5"/>
        <item name="setValue" dataType="5"/>
        <item name="checkType" dataType="2"/>
        <item name="unit" dataType="8"/>
        <item name="paaRel" dataType="3"/>
    </structDef>
    <values>
        <item pos="1" name="Torque" value="66.834782" result="1"
loLim="60.000000"
                upLim="70.000000" setValue="65.000000" checkType="6" unit="Nm"
paaRel="5"/>
        <item pos="1" name="Angle" value="14.839000" result="1" loLim="13.500000"
                upLim="15.500000" setValue="15.000000" checkType="6" unit="Deg"
paaRel="0"/>
        <item pos="2" name="Torque" value="67.489998" result="1"
loLim="60.000000"
                upLim="70.000000" setValue="65.000000" checkType="6" unit="Nm"
paaRel="5"/>
        <item pos="2" name="Angle" value="14.750000" result="1" loLim="13.500000"
                upLim="15.500000" setValue="15.000000" checkType="6" unit="Deg"
paaRel="0"/>
        <item pos="3" name="Torque" value="67.521398" result="1"
loLim="60.000000"
                upLim="70.000000" setValue="65.000000" checkType="6" unit="Nm"
paaRel="5"/>
        <item pos="3" name="Angle" value="13.999999" result="1" loLim="13.500000"
                upLim="15.500000" setValue="15.000000" checkType="6" unit="Deg"
paaRel="0"/>
        <item pos="4" name="Torque" value="69.432981" result="1"
loLim="60.000000"
                upLim="70.000000" setValue="65.000000" checkType="6" unit="Nm"
paaRel="5"/>
        <item pos="4" name="Angle" value="15.124875" result="1" loLim="13.500000"
                upLim="15.500000" setValue="15.000000" checkType="6" unit="Deg"
paaRel="0"/>
        <item pos="5" name="Torque" value="68.932709" result="1"
loLim="60.000000"
                upLim="70.000000" setValue="65.000000" checkType="6" unit="Nm"
paaRel="5"/>
        <item pos="5" name="Angle" value="14.143490" result="1" loLim="13.500000"
                upLim="15.500000" setValue="15.000000" checkType="6" unit="Deg"
paaRel="0"/>
        <item pos="6" name="Torque" value="61.907878" result="1"
loLim="60.000000"
                upLim="70.000000" setValue="65.000000" checkType="6" unit="Nm"
paaRel="5"/>
        <item pos="6" name="Angle" value="15.389098" result="1" loLim="13.500000"
                upLim="15.500000" setValue="15.000000" checkType="6" unit="Deg"
paaRel="0"/>
    </values>
</array>
<array name="processIntData">
    <structDef>
        <item name="pos" dataType="3"/>
        <item name="name" dataType="8"/>
        <item name="value" dataType="3"/>
        <item name="result" dataType="2"/>
        <item name="loLim" dataType="3"/>
        <item name="upLim" dataType="3"/>
        <item name="setValue" dataType="3"/>
        <item name="checkType" dataType="2"/>
        <item name="unit" dataType="8"/>
        <item name="paaRel" dataType="3"/>
    </structDef>

```

```

</structDef>
<values>
  <item pos="1" name="Force" value="153" result="1" loLim="150" upLim="155"
    setValue="151" checkType="6" unit="N" paaRel="0"/>
  <item pos="2" name="Force" value="151" result="1" loLim="150" upLim="155"
    setValue="151" checkType="6" unit="N" paaRel="0"/>
  <item pos="3" name="Force" value="153" result="1" loLim="150" upLim="155"
    setValue="151" checkType="6" unit="N" paaRel="0"/>
  <item pos="4" name="Force" value="154" result="1" loLim="150" upLim="155"
    setValue="151" checkType="6" unit="N" paaRel="0"/>
  <item pos="5" name="Force" value="153" result="1" loLim="150" upLim="155"
    setValue="151" checkType="6" unit="N" paaRel="0"/>
  <item pos="6" name="Force" value="153" result="1" loLim="150" upLim="155"
    setValue="151" checkType="6" unit="N" paaRel="0"/>
</values>
</array>
<array name="results">
  <structDef>
    <item name="pos" dataType="3"/>
    <item name="result" dataType="2"/>
    <item name="nioBits" dataType="3"/>
  </structDef>
  <values>
    <item pos="1" result="1" nioBits="0" />
    <item pos="2" result="1" nioBits="0" />
    <item pos="3" result="1" nioBits="0" />
    <item pos="4" result="1" nioBits="0" />
    <item pos="5" result="1" nioBits="0" />
    <item pos="6" result="1" nioBits="0" />
  </values>
</array>
</structArrays>
</body>
</root>

```

Example

In the following example, all explicitly defined elements of the processing end are once again included in one message. This is therefore the maximum extension of a `partProcessed` message with methods of the AE standard.

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.2" eventName="partProcessed" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partProcessed identifier="DMC22934"/>
  </event>
  <body>
    <structs>
      <resHead result="1" typeNo="1111111111" typeVar="" workingCode="0"
        nioBits="0"/>
    </structs>
    <items>
      <item name="Prog1.Name" value="PRG_MILL_1111111111" dataType="8"/>
      <item name="Prog1.SubSystem" value="BL922" dataType="8"/>
      <item name="Prog2.Name" value="NC808098" dataType="8"/>
      <item name="Prog2.SubSystem" value="XT2" dataType="8"/>
    </items>
    <structArrays>
      <array name="toolData">
        <structDef>

```



```
<item name="toolID" dataType="8"/>
<item name="consumption" dataType="3"/>
<item name="consumptionNio" dataType="3"/>
</structDef>
<values>
  <item toolID="1002090001" consumption="3" consumptionNio="0"/>
  <item toolID="786644001" consumption="3" consumptionNio="0"/>
</values>
</array>
<array name="matData">
  <structDef>
    <item name="idx" dataType="3"/>
    <item name="matID" dataType="8"/>
    <item name="qty" dataType="5"/>
  </structDef>
  <values>
    <item idx="1" matID="884566S123456789055" qty="2889.00"/>
    <item idx="2" matID="303479S234234234777" qty="1334.00"/>
  </values>
</array>
<array name="comp">
  <structDef>
    <item name="pos" dataType="3"/>
    <item name="ref" dataType="3"/>
    <item name="refDes" dataType="8"/>
  </structDef>
  <values>
    <item pos="1" ref="1" refDes="R100"/>
    <item pos="1" ref="1" refDes="R101"/>
    <item pos="1" ref="1" refDes="R102"/>
    <item pos="1" ref="2" refDes="C10"/>
    <item pos="1" ref="2" refDes="C20"/>
    <item pos="2" ref="1" refDes="R100"/>
    <item pos="2" ref="1" refDes="R101"/>
    <item pos="2" ref="1" refDes="R102"/>
    <item pos="2" ref="2" refDes="C10"/>
    <item pos="2" ref="2" refDes="C20"/>
    <item pos="3" ref="1" refDes="R100"/>
    <item pos="3" ref="1" refDes="R101"/>
    <item pos="3" ref="1" refDes="R102"/>
    <item pos="3" ref="2" refDes="C10"/>
    <item pos="3" ref="2" refDes="C20"/>
    <item pos="4" ref="1" refDes="R100"/>
    <item pos="4" ref="1" refDes="R101"/>
    <item pos="4" ref="1" refDes="R102"/>
    <item pos="4" ref="2" refDes="C10"/>
    <item pos="4" ref="2" refDes="C20"/>
    <item pos="5" ref="1" refDes="R100"/>
    <item pos="5" ref="1" refDes="R101"/>
    <item pos="5" ref="1" refDes="R102"/>
    <item pos="5" ref="2" refDes="C10"/>
    <item pos="5" ref="2" refDes="C20"/>
    <item pos="6" ref="1" refDes="R100"/>
    <item pos="6" ref="1" refDes="R101"/>
    <item pos="6" ref="1" refDes="R102"/>
    <item pos="6" ref="2" refDes="C10"/>
    <item pos="6" ref="2" refDes="C20"/>
  </values>
</array>
<array name="panelRealData">
  <structDef>
    <item name="name" dataType="8"/>
    <item name="value" dataType="5"/>
    <item name="result" dataType="2"/>
    <item name="loLim" dataType="5"/>
    <item name="upLim" dataType="5"/>
```

```

        <item name="setValue" dataType="5"/>
        <item name="checkType" dataType="2"/>
        <item name="unit" dataType="8"/>
        <item name="paaRel" dataType="3"/>
    </structDef>
    <values>
        <item name="Temperature" value="68.345000" result="1" loLim="60.000000"
upLim="70.000000"
            setValue="65.000000" checkType="6" unit="C" paaRel="5"/>
    </values>
</array>
<array name="processRealData">
    <structDef>
        <item name="pos" dataType="3"/>
        <item name="name" dataType="8"/>
        <item name="value" dataType="5"/>
        <item name="result" dataType="2"/>
        <item name="loLim" dataType="5"/>
        <item name="upLim" dataType="5"/>
        <item name="setValue" dataType="5"/>
        <item name="checkType" dataType="2"/>
        <item name="unit" dataType="8"/>
        <item name="paaRel" dataType="3"/>
    </structDef>
    <values>
        <item pos="1" name="Torque" value="66.834782" result="1"
loLim="60.000000"
            upLim="70.000000" setValue="65.000000" checkType="6" unit="Nm"
paaRel="5"/>
        <item pos="1" name="Angle" value="14.839000" result="1" loLim="13.500000"
            upLim="15.500000" setValue="15.000000" checkType="6" unit="Deg"
paaRel="0"/>
        <item pos="2" name="Torque" value="67.489998" result="1"
loLim="60.000000"
            upLim="70.000000" setValue="65.000000" checkType="6" unit="Nm"
paaRel="5"/>
        <item pos="2" name="Angle" value="14.750000" result="1" loLim="13.500000"
            upLim="15.500000" setValue="15.000000" checkType="6" unit="Deg"
paaRel="0"/>
        <item pos="3" name="Torque" value="67.521398" result="1"
loLim="60.000000"
            upLim="70.000000" setValue="65.000000" checkType="6" unit="Nm"
paaRel="5"/>
        <item pos="3" name="Angle" value="13.999999" result="1" loLim="13.500000"
            upLim="15.500000" setValue="15.000000" checkType="6" unit="Deg"
paaRel="0"/>
        <item pos="4" name="Torque" value="69.432981" result="1"
loLim="60.000000"
            upLim="70.000000" setValue="65.000000" checkType="6" unit="Nm"
paaRel="5"/>
        <item pos="4" name="Angle" value="15.124875" result="1" loLim="13.500000"
            upLim="15.500000" setValue="15.000000" checkType="6" unit="Deg"
paaRel="0"/>
        <item pos="5" name="Torque" value="68.932709" result="1"
loLim="60.000000"
            upLim="70.000000" setValue="65.000000" checkType="6" unit="Nm"
paaRel="5"/>
        <item pos="5" name="Angle" value="14.143490" result="1" loLim="13.500000"
            upLim="15.500000" setValue="15.000000" checkType="6" unit="Deg"
paaRel="0"/>
        <item pos="6" name="Torque" value="61.907878" result="1"
loLim="60.000000"
            upLim="70.000000" setValue="65.000000" checkType="6" unit="Nm"
paaRel="5"/>
        <item pos="6" name="Angle" value="15.389098" result="1" loLim="13.500000"

```

```

        upLim="15.500000" setValue="15.000000" checkType="6" unit="Deg"
paaRel="0"/>
</values>
</array>
<array name="processIntData">
  <structDef>
    <item name="pos" dataType="3"/>
    <item name="name" dataType="8"/>
    <item name="value" dataType="3"/>
    <item name="result" dataType="2"/>
    <item name="loLim" dataType="3"/>
    <item name="upLim" dataType="3"/>
    <item name="setValue" dataType="3"/>
    <item name="checkType" dataType="2"/>
    <item name="unit" dataType="8"/>
    <item name="paaRel" dataType="3"/>
  </structDef>
  <values>
    <item pos="1" name="Force" value="153" result="1" loLim="150" upLim="155"
      setValue="151" checkType="6" unit="N" paaRel="0"/>
    <item pos="2" name="Force" value="151" result="1" loLim="150" upLim="155"
      setValue="151" checkType="6" unit="N" paaRel="0"/>
    <item pos="3" name="Force" value="153" result="1" loLim="150" upLim="155"
      setValue="151" checkType="6" unit="N" paaRel="0"/>
    <item pos="4" name="Force" value="154" result="1" loLim="150" upLim="155"
      setValue="151" checkType="6" unit="N" paaRel="0"/>
    <item pos="5" name="Force" value="153" result="1" loLim="150" upLim="155"
      setValue="151" checkType="6" unit="N" paaRel="0"/>
    <item pos="6" name="Force" value="153" result="1" loLim="150" upLim="155"
      setValue="151" checkType="6" unit="N" paaRel="0"/>
  </values>
</array>
<array name="results">
  <structDef>
    <item name="pos" dataType="3"/>
    <item name="result" dataType="2"/>
    <item name="nioBits" dataType="3"/>
  </structDef>
  <values>
    <item pos="1" result="1" nioBits="0" />
    <item pos="2" result="1" nioBits="0" />
    <item pos="3" result="1" nioBits="0" />
    <item pos="4" result="1" nioBits="0" />
    <item pos="5" result="1" nioBits="0" />
    <item pos="6" result="1" nioBits="0" />
  </values>
</array>
<array name="errors">
  <structDef>
    <item name="name" dataType="8"/>
    <item name="errType" dataType="2"/>
    <item name="errNumber" dataType="8"/>
    <item name="errInfo" dataType="8"/>
  </structDef>
  <values>
  </values>
</array>
<array name="spcResults">
  <structDef>
    <item name="testNo" dataType="3"/>
    <item name="testResult" dataType="3"/>
    <item name="blockParts" dataType="11"/>
    <item name="sessionID" dataType="8"/>
  </structDef>
  <values>
  </values>

```

```

        </array>
    </structArrays>
</body>
</root>

```

4.3.4.2.4.1 Response from MES

The response to be sent from MES is process-specific and must be described in the relevant specifications of the default processes.

Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.3" eventName="partProcessed"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="0"/>
    <trace/>
  </event>
</body>
</root>

```

If tool information is also sent in the request, the response can also contain the current remaining strokes for the relevant tools.

Example:

The two tools 1002090001 and 786644001 , including usage counters, have been sent to MES. The remaining strokes are returned in the response:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.3" eventName="partProcessed"
    contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partProcessed identifier="DMC22934"/>
  </event>
  <body>
    <structs>
      <resHead result="1" typeNo="111111111" typeVar="" workingCode="0"
        nioBits="0"/>
    </structs>
    <structArrays>
      <array name="toolData">
        <structDef>
          <item name="toolID" dataType="8"/>
          <item name="consumption" dataType="3"/>
          <item name="consumptionNio" dataType="3"/>
        </structDef>
        <values>
          <item toolID="1002090001" consumption="3" consumptionNio="0"/>
          <item toolID="786644001" consumption="3" consumptionNio="0"/>
        </values>
      </array>
    </structArrays>
  </body>

```

```
</body>
</root>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="2326" version="2.2" eventName="partProcessed"
    timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <result returnCode="0"/>
    <trace/>
  </event>
  <body>
    <structArrays>
      <array name="tools">
        <structDef>
          <item name="toolID" dataType="8"/>
          <item name="state" dataType="2"/>
          <item name="counter" dataType="3"/>
        </structDef>
        <values>
          <item toolID="1002090001" state="0" counter="119999"/>
          <item toolID="786644001" state="0" counter="65434"/>
        </values>
      </array>
    </structArrays>
  </body>
</root>
```

4.3.4.2.4.1.1 Incremental sending of the processing result

In some processes it may be useful to send the process results incrementally. In these cases, the following sequence must be used:

- The first 1..n-1 partial results must be sent to MES via a `dataUploadRequired` event (`eventSwitch=30`).
- The last (n-th) partial result must be reported via a `partProcessed` event.

So that all partial results can be processed correctly, the full location (all parts including `ToolPos` must be `<>0`) must be sent in the messages. The individual messages must have a different `ToolPosition`, including the `partProcessed` message.

4.3.4.2.4.2 Incremental Sending of the Processing Result

In some processes it may be useful to send the process results incrementally. In these cases, the following sequence must be used:

- The first 1...(n-1) partial results must be sent to MES via a `dataUploadRequired` event (`eventSwitch = 30`).
- The last (n-th) partial result must be reported via a `partProcessed` event.

So that all partial results can be processed correctly, the full location (all parts including `ToolPos` must be `<>0`) must be sent in the message. The individual messages must have a different `ToolPosition`, including the `partProcessed` message.

4.3.5 Changing the Workpiece

If a workpiece is changed during line processing, the event `partDisplaced` must be sent by the station. After receiving the message, MES does not normally send an acknowledgment to the station. In special cases, however, it is permitted for MES to send a response to the station.

Within this event, in addition to the data specified in Part I, the following additional data can also be sent by the station:

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="88" version="2.3" eventName="partDisplaced" contentType="3">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processNo="90" processName="PlasmaCleaning" application="PLC"/>
  </header>
  <event>
    <partDisplaced oldidentifier="DMC22934" identifier="DMC22934"/>
  </event>
  <body/>
</root>
```

4.4 Material data

4.4.1 Material master data query

If you want to explicitly query the master data for a material, the event `dataDownloadRequired` with `eventSwitch = 16` must be used. Only the Mat ID is transferred in the request, and the material master data is transferred in the response, in the same way as `plcMaterialChangeStarted`.

Query:

Information	Required	Data type	OpCon Xml	Comments
Material ID	Mandatory	STRING	<code>item.MatID</code>	Mat ID of the material

Response

Information	Required	Data type	OpCon Xml	Comments
Material ID	Mandatory	STRING	<code>matData[]@matID</code>	Uniquely identifies the material
Status of the material	Optional	INT	<code>matData[]@state</code>	Possible statuses: -1: Unknown status 0: Released 1: Warning limit reached (e.g. in MSD time monitoring) 2: Locked
Material number	Optional	STRING	<code>matData[]@patNo</code>	TTNR of the material that the station has to install. This number corresponds e.g. to the field 'P' (Customer Part Number) of the Mat Label specification.
Model part number of the material	Optional	STRING(10)	<code>matData[]@typeNo</code>	Obsolete, do not use for new interfaces. Model variant/version of the material
Model variant of the material (obsolete)	Optional	STRING(10)	<code>matData[]@typeVar</code>	Obsolete, do not use for new interfaces. Model variant/version of the material

Revision status of the material	Optional	STRING	matData[]@revision	Revision status of the material that the station has to install.
Batch	Optional	STRING	matData[]@batch	Manufacturer batch
Manufacturer/Supplier	Optional	STRING	matData[]@vendor	
Residual quantity	Optional	LREAL	matData[]@qty	Residual amount of material
Master Mat ID	Optional	STRING	matData[]@masterMatID	
ERP ID of the material	Optional	STRING	matData[]@erpID	
Time stamp of material entry	Optional	STRING	matData[]@rcvTimeStamp	Specifies when the material was posted in the incoming goods station.
ID of the incoming goods station	Optional	STRING	matData[]@rcvStation	
Operator of the incoming goods station	Optional	STRING	matData[]@operator	
Version number of the stored Mat Label	Optional	DINT	matData[]@labelVersion	
Manufacturer part number	Optional	STRING	matData[]@manufacturerTypeNo	
Order number	Optional	STRING	matData[]@orderingNo	
Additional data for the material	Optional	STRING	matData[]@partAddInfo	
Manufacturing time stamp	Optional	STRING	matData[]@productionDate	
Expiry date	Optional	STRING	matData[]@expirationDate	
roHS Info	Optional	STRING	matData[]@roHS	
MSD level	Optional	STRING	matData[]@msLevel	
Order number	Optional	STRING	matData[]@purchaseOrderNo	
Delivery note number	Optional	STRING	matData[]@shippingNoteNo	
Supplier	Optional	STRING	matData[]@supplierID	
Package ID	Optional	STRING	matData[]@quantityUnit	
Unit of measurement	Optional	STRING	matData[]@quantityUnit	For example meters, kilograms, units, etc.
Number of batches	Optional	DINT	matData[]@batchCounter	Specifies how many batches were grouped together to form this material unit
Batch ID 1	Optional	STRING	matData[]@batch1	Batch ID of material
Batch ID2	Optional	STRING	matData[]@batch2	Optional additional batch
Supplier data	Optional	STRING	matData[]@supplierData	
No. of SPUs	Optional	DINT	matData[]@numberOfSPU	
Residual quantity	Optional	LREAL	matData[]@remainingQty	
Time stamp of the first opening	Optional	STRING	matData[]@firstOpenedTimestamp	Specifies when the material was first opened.
Consumed time stamp	Optional	STRING	matData[]@usedUpTimestamp	Specifies when the material has been fully consumed.
Current setup location	Optional	STRING	matData[]@currentLocation	
Current setup	Optional	STRING	matData[]@currentSetup	Only relevant in connection with component insertion machines.

Position within the machine	Optional	STRING	matData[] @stationInfo	Only relevant in connection with component insertion machines.
Subsequently set expiry date	Optional	STRING	matData[] @expirationDateOverride	
Material to which this material has been spliced.	Optional	STRING	matData[] @splicedPrefixMatID	Only relevant in connection with component insertion machines.
Original unit of measure	Optional	LREAL	matData[] @QuantityRawUnit	
Format of the Mat Label	Optional	STRING	matData[]@labelFormat	

Example

The following example message shows how the master data for the material with MAT ID '101B100000000001' is queried.

Query

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="88" version="2.3" eventName="dataDownloadRequired"
    eventSwitch="16" contentType="3">
    <location lineNo="8030" statNo="40" statIdx="1" fuNo="1"
      processNo="100" processName="Connector Gel Dispensing"
      application="PLC"/>
  </header>
  <event>
    <dataDownloadRequired />
  </event>
  <body>
    <items>
      <item name="MatID" value="101B100000000001" dataType="8"/>
    </items>
  </body>
</root>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="88" version="2.3" eventName="dataDownloadRequired"
    eventSwitch="16" contentType="3">
    <location lineNo="8030" statNo="40" statIdx="1" fuNo="1"
      processNo="100" processName="Connector Gel Dispensing"
      application="PLC"/>
  </header>
  <event>
    <result returnCode="0"/></event>
  <body>
    <structArrays>
      <array name="matData">
        <structDef>
          <item name="matID" dataType="8"/>
          <item name="state" dataType="2"/>
          <item name="typeNo" dataType="8"/>
          <item name="typeVar" dataType="8"/>
          <item name="qty" dataType="5"/>
        </structDef>
        <values>
```




```

        <item matID="101B100000000001" state="0"
              typeNo="3001010010" typeVar="" qty="4982,90"/>
      </values>
    </array>
    <array name="matBlocks">
      <structDef>
        <item name="matID" dataType="8"/>
        <item name="reasonID" dataType="3"/>
        <item name="reasonText" dataType="8"/>
      </structDef>
      <values/>
    </array>
  </structArrays>
</body>
</root>

```

4.4.2 Manual Material Message

If the current material consumption/residual quantity is reported independently of the processing result, a `dataUploadRequired` event with `eventSwitch = 15` must be sent.

 The manual material message also includes the special case in which a station must explicitly report that a material has been fully consumed. In this case, a `dataUploadRequired`-event with `eventSwitch=15` and residual quantity of the material = 0 must also be sent.

If multiple materials have to be reported as consumed, a `dataUploadRequired` event must be sent for each material. It is not permitted to group multiple materials together into one event.

After receiving the message, MES sends an acknowledgment to the station. No further information about the material is returned to the station.

Query

Information	Required	Data type	OpCon Xml	Comments
Material ID	Mandatory	STRING	<code>item.MatID</code>	Mat ID of the material
Quantity	Mandatory	DINT	<code>item.Qty</code>	Must always be set to '0' for the consumption message.
Type of quantity	Mandatory	INT	<code>item.QuantityMode</code>	Defines how the quantity must be interpreted: 0 = Residual quantity 1 = Consumption

Example

The following example message shows how the material with MAT ID '101B100000000001' is explicitly reported as fully consumed.

Query

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="88" version="2.3" eventName="dataUploadRequired"
          eventSwitch="15" contentType="3">
    <location lineNo="8030" statNo="40" statIdx="1" fuNo="1"
              processNo="100" processName="Connector Gel Dispensing"
              application="PLC"/>
  </header>
  <event>
    <dataUploadRequired />
  </event>
</body>

```

```

    <items>
      <item name="MatID" value="101B100000000001" dataType="8"/>
      <item name="Qty" value="0.0" dataType="5"/>
      <item name="QuantityMode" value="0" dataType="2"/>
    </items>
  </body>
</root>

```

Response

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="88" version="2.3" eventName="dataUploadRequired"
    eventSwitch="15" contentType="3">
    <location lineNo="8030" statNo="40" statIdx="1" fuNo="1"
      processNo="100" processName="Connector Gel Dispensing"
      application="PLC"/>
  </header>
  <event>
    <result returnCode="0"/></event>
  <body>
  </body>
</root>

```

4.5 Other Actions

4.5.1 User Login

If a user needs to log in explicitly to the process, this is done using the event `plcLogIn`. In this event, the user ID and password must be sent. In the response to `plcLogIn`, the roles of the user are then transferred. From these roles, the process has to determine whether or not the user may execute the action.

Example:

The following example shows the login process for the user `xyz2zz`. The user data is transferred to OpCon MES via the event `plcLogIn`. The password is encrypted using the MD5 algorithm.

The user has been successfully logged in and the list of roles (`WE_Engineer` and `MAT_Default`) of the user are returned to the process.

Query:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="21" version="2.3" eventName="plcLogIn" contentType="3">
    <location lineNo="1" statNo="10" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processName="Receiving" processNo="90" application="IPC" />
  </header>
  <event>
    <plcLogIn user="xyz2zz" pwd="md5:a934x75yy23"/>
  </event>
  <body/>
</root>

```

Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="21" version="2.3" eventName="plcLogIn"

```

```

        timeStamp="2008-11-23T13:21:34.231+02:00" contentType="3">
        <location lineNo="1" statNo="10" statIdx="1" fuNo="1" workPos="1" toolPos="1"
            processName="Receiving" processNo="90" application="IPC" />
    </header>
    <event>
        <result returnCode="0"/>
        <trace/>
    </event>
    <body>
        <structArrays>
            <array name="Roles">
                <structDef>
                    <item name="name" dataType="8"/>
                    <item name="description" dataType="8"/>
                </structDef>
                <values>
                    <item name="WE_Engineer" description="Receiving station engineer"/>
                    <item name="MAT_Default" description="Default user for MatControl"/>
                </values>
            </array>
        </structArrays>
    </body>
</root>

```

As an alternative list of roles, the authorizations can be returned to the caller in bit-coded form. In this case, these authorizations only refer to the process locally.

4.5.2 Logging a User Off

The logoff of a user is carried out by sending the `plcLogOff` event. MES then logs the currently logged-in user off. Subsequent events are then carried out in the context of the standard user configured in MES.

4.5.3 Transferring Data for Customer Labels

Customer labels have different structure and content. In order to allow uniform transmission, all data of a customer label must be transferred in the element `idPlate`. This element is a structure field and has the following content:

Information	Required	Data type	OpCon Xml	Comments
Field in customer label	X	STRING	<code>idPlate[]@field</code>	ID of the field
Field name in the customer label	Optional	STRING	<code>idPlate[]@name</code>	Descriptive name of the field
Content of the field in the customer label	X	STRING	<code>idPlate[]@value</code>	-

```

<structArrays>
    <array name="idPlate">
        <structDef>
            <item name="field" dataType="8"/>
            <item name="name" dataType="8"/>
            <item name="value" dataType="8"/>
        </structDef>
        <values>
            <item field="Feld1" name="Werkskennzahl" value="078"/>
            <item field="Feld2" name="Hardware-Version" value="0123456789"/>
            <item field="Feld3" name="Geraetebezeichnung" value="Radarsensor"/>
            <item field="Feld4" name="Radarzulassungsnummer" value="1"/>
            <item field="Feld5" name="Tagesfertigungsdatum" value="10.01.2009"/>
            <item field="Feld6" name="Herstellerpruefstand-Nummer" value="4711"/>
        </values>
    </array>
</structArrays>

```

4.5.4 Posting and Querying Material Components

Components are posted by scanning the standardized MAT label, always using the same information. If you want component information to be posted during the manufacturing process, this is sent within the `partProcessed` event. If you want to query component information during the manufacturing process, this is typically sent to the requesting process within the `partReceivedresponse`.

The data of the Mat Labels is always sent in the following format (regardless of whether it is in the query or response):

Information	Required	Data type	OpCon Xml	Comments
Label format	Optional	DINT	<code>item.LabelFormat</code>	-
Bosch TTNR	Optional	STRING	<code>item.ComponentX.TypeNo</code>	Model part number that Bosch has assigned for the material.
Model variant	Optional	STRING	<code>item.ComponentX.TypeVar</code>	Bosch model variant
Manufacturer TTNR	Optional	STRING	<code>item.ComponentX.ManufacturerTypeNo</code>	Model part number that the manufacturer has assigned for the material.
Order number	Optional	STRING	<code>item.ComponentX.OrderingNo</code>	-
Manufacturer	Optional	STRING	<code>item.ComponentX.Manufacturer</code>	-
Place of manufacture	Optional	STRING	<code>item.ComponentX.ManufacturerLocation</code>	-
Additional Information	Optional	STRING	<code>item.ComponentX.PartAddInfo</code>	-
Date of manufacture	Optional	STRING	<code>item.ComponentX.ProductionDate</code>	-
Expiry date	Optional	STRING	<code>item.ComponentX.ExpirationDate</code>	-
RoHS	Optional	STRING	<code>item.ComponentX.RoHS</code>	-
Humidity classification	Optional	STRING	<code>item.ComponentX.MsLevel</code>	-
Purchase Order Number	Optional	STRING	<code>item.ComponentX.PurchaseOrderNo</code>	-
Consignment Note Number	Optional	STRING	<code>item.ComponentX.ShippingNoteNo</code>	-
Supplier	Optional	STRING	<code>item.ComponentX.SupplierID</code>	-
Packaging unit ID	Optional	STRING	<code>item.ComponentX.PackageID</code>	-
Quantity	Optional	REAL	<code>item.ComponentX.Quantity</code>	-
Unit of quantity	Optional	STRING	<code>item.ComponentX.QuantityUnit</code>	-
Number of batches in package	Optional	DINT	<code>item.ComponentX.BatchCounter</code>	A maximum of two batches may be included in a package.
Batch 1	Optional	STRING	<code>item.ComponentX.Batch1</code>	Contains the batch ID of the first batch in the package. If the package only contains one batch, the batch is entered in this field.
Batch 2	Optional	STRING	<code>item.ComponentX.Batch2</code>	Contains the batch ID of the second batch in the package. If the package only contains one batch, this field is omitted or contains an empty string ("").

	Optional	STRING	item.ComponentX. MasterMatID	
	Optional	STRING	item.ComponentX.ErpI D	
	Optional	STRING	item.ComponentX. RcvTimeStamp	
	Optional	STRING	item.ComponentX. RcvStation	
	Optional	STRING	item.ComponentX. Operator	
	Optional	STRING	item.ComponentX. supplierData	
	Optional	DINT	item.ComponentX. numberOfSPU	
	Optional	LREAL	item.ComponentX. remainingQty	
	Optional	STRING	item.ComponentX. firstOpenedTimestamp	
	Optional	STRING	item.ComponentX. usedUpTimestamp	
	Optional	STRING	item.ComponentX. currentLocation	
	Optional	STRING	item.ComponentX. stationInfo	
	Optional	STRING	item.ComponentX. expirationDateOverri de	
	Optional	STRING	item.ComponentX. splicedPrefixMatID	
	Optional	DINT	item.ComponentX. labelQuantity	
	Optional	STRING	item.ComponentX. labelFormat	
	Optional	STRING	item.ComponentX. revision	

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header eventId="21" version="2.3" eventName="..." contentType="3">
    <location lineNo="1" statNo="10" statIdx="1" fuNo="1" workPos="1" toolPos="1"
      processName="Receiving" processNo="90" application="IPC" />
  </header>
  <event>
    (...)
  </event>
  <body>
    (...)
    <items>
      <item name="LabelFormat" value="2" dataType="3"/>
      <item name="Component1.TypeNo" value="A2C53216419" dataType="8"/>
      <item name="Component1.TypeVar" value="AA" dataType="8"/>
      <item name="Component1.ManufacturerTypeNo" value="SL105C103MAA-S"
dataType="8"/>
      <item name="Component1.OrderingNo" value="SC441427CFNR2" dataType="8"/>
      <item name="Component1.Manufacturer" value="123456789" dataType="8"/>
      <item name="Component1.ManufacturerLocation" value="BEpl1FRp17DEpl2"
dataType="8"/>
      <item name="Component1.PartAddInfo" value="" dataType="8"/>
      <item name="Component1.ProductionDate" value="20080330" dataType="8"/>
      <item name="Component1.ExpirationDate" value="20081031" dataType="8"/>
      <item name="Component1.RoHS" value="Y" dataType="8"/>
      <item name="Component1.MsLevel" value="2a" dataType="8"/>
    </items>
  </body>
</root>
```

```
<item name="Component1.PurchaseOrderNo" value="753013" dataType="8"/>
<item name="Component1.ShippingNoteNo" value="54003333" dataType="8"/>
<item name="Component1.SupplierID" value="884566" dataType="8"/>
<item name="Component1.PackageID" value="S123456789012" dataType="8"/>
<item name="Component1.Quantity" value="10.020" dataType="5"/>
<item name="Component1.QuantityUnit" value="KGM" dataType="8"/>
<item name="Component1.BatchCounter" value="2" dataType="3"/>
<item name="Component1.Batch1" value="750160429" dataType="8"/>
<item name="Component1.Batch2" value="750160430" dataType="8"/>
<item name="Component1.SupplierData" value="" dataType="8"/>
(...)
</items>
</body>
</root>
```

5 Appendix A: Setpoint values and constants

5.1 Codes for Material Locks

The following table contains the lock codes known at the time of writing. These are the minimum codes that can be used; the list may be extended to include other codes in future.

Code	Description
1	Material is not released for manufacturing
2	Mat Label is not yet verified
3	Material was manually locked
4	Material was locked due to a lock rule
5	Mat ID locked because the material was relabeled
6	Duplicate Mat ID
7	The maximum allowable lifetime of the material expired
8	MSD open time for the material was exceeded
9	Applying the Mat Label failed

Preset dataDownloadRequired event switch values

eventSwitch	Description
1	Model data
2	Model family
3	reserved
4	reserved
5	reserved
6	reserved
7	Station data
8	System data 1
9	reserved
10	reserved
11	reserved
12	Line data
13	System data 2
14	System data 3
15	reserved
16	Query of material master data
60	Multi-process type data (flexible subsequent loading of the type data of an individual process that is in a location with several other processes)

Preset dataUploadRequired event switch values

eventSwitch	Description
15	Explicit material quantity message
30	Incremental sending of the processing result (see page 77)

6 Index

A

About this Documentation - 4

Appendix A

Setpoint values and constants - 87

B

Basic Data - 55, 65

BDE/MDE - 9

Brief Description - 8

C

Canceling Processing - 54

Category

Container - 64

Programs - 64

Change Index - 2

Changeover to Product Type - 11

Changing the Material - 24

Changing the Setup Status - 9

Changing the Workpiece - 78

Codes for Material Locks - 87

Completing the Changeover Process - 21

Completing the Material Change - 30

Conclusion of the Tool Setup - 24

Conventions - 4

Current Setup Status (Optional) - 10

D

Data Structures and Conventions - 11

E

Examples - 8

G

Group Processing Request - 49

Group Request - 64

I

Icons used in this manual - 4

Incremental sending of the processing result - 77

Incremental Sending of the Processing Result - 77

Individual Request - 55

Initiating the Material Change - 24

Initiating Tool Setup - 21

Intended Use - 5

Introduction - 8

L

Legal Framework and Copyright - 7

Logging a User Off - 83

M

Manual Material Message - 81

Material data - 78

Material master data query - 78

Material requirements - 12

Materials - 38

O

Other Actions - 82

P

Part 3

Specification AE - 9

Posting and Querying Material Components - 84
Process Data - 58, 69
Process failures - 56
Process Specifications - 14
Processing Complete - 54
Processing Release - 31
Processing Result - 55, 65
Production Data - 31
Programs - 13

R

Reporting Station Errors - 9
Response from MES - 76

S

Safety Regulations and Warnings - 6
Scenario
 BPS Change - 16
 Simplified Changeover (Obsolete) - 19
Setting Up the Tool - 21
Setup Data - 11
SPC test results - 62
Special Information - 63
Starting to Process the Workpiece - 53

T

Tool requirements - 12
Tools - 38, 61
Trace Data of the Installed Materials - 56, 67
Transferring Data for Customer Labels - 83
Transport medium (workpiece carrier, magazine, container) - 13

U

User Login - 82

V

Verifying the Material Setup - 32
Verifying the Setup - 33
Verifying the Tool Setup - 32
Verifying Uniquely Identifiable Components (Additional Components) - 33

W

Workpiece Data - 14

Impressum

© Copyright 2015 Robert Bosch GmbH

Alle Rechte vorbehalten.

Jede Weitergabe oder Drittverwendung dieses Dokuments, ganz oder auszugsweise, ist nur mit schriftlicher Erlaubnis der Robert Bosch GmbH erlaubt.