# ERP Interface - Data and Events via XML

Version 5.11

*Manual*

Document: Manual - ERP Interface Data and Events via XML.docx

Release date: 26.05.21

Document version: 1

Author: STernes

FB_8-41 CREATED: 02.06.21 LAST CHANGES: 12.07.21 VERSION: V1
APPROVED BY: HEAD OF TTE AUTHOR: TECHNICAL WRITER CLASSIFICATION: PUBLIC

# Content

# 1 General

This document describes the communication link between FORCAM FORCE™ and the ERP system, and gives more details about the transmitted data, event and process structures.

There are several alternatives or options for this type of communication interface.
First, communication can be done directly through the FORCAM FORCE™ runtime using the open and standardized XSL transformation (XSLT) method. The XSLT is part of the XSL (Extensible Stylesheet Language) and performs an application-specific transformation of the transferred data structures. This procedure is used, for example, when connecting ERP systems that exchange data sets in SAP IDoc format and is described in more detail in the **Manual - ERP Interface - Data and Events via SAP IDoc**.

More advanced, more secure, and considerably more reliable is the alternative connection provided by the FORCAM FORCE™ programming interface FORCAM FORCE™ Bridge API, a RESTful interface to FORCAM FORCE™. This procedure is described in the **Manual - Bridge API** and is also used as the basis of this manual.

Both procedures exchange data sets in the form of XML records via the HTTP or HTTPS protocol. This uses an HTTP POST request with the XML datasets as payload.

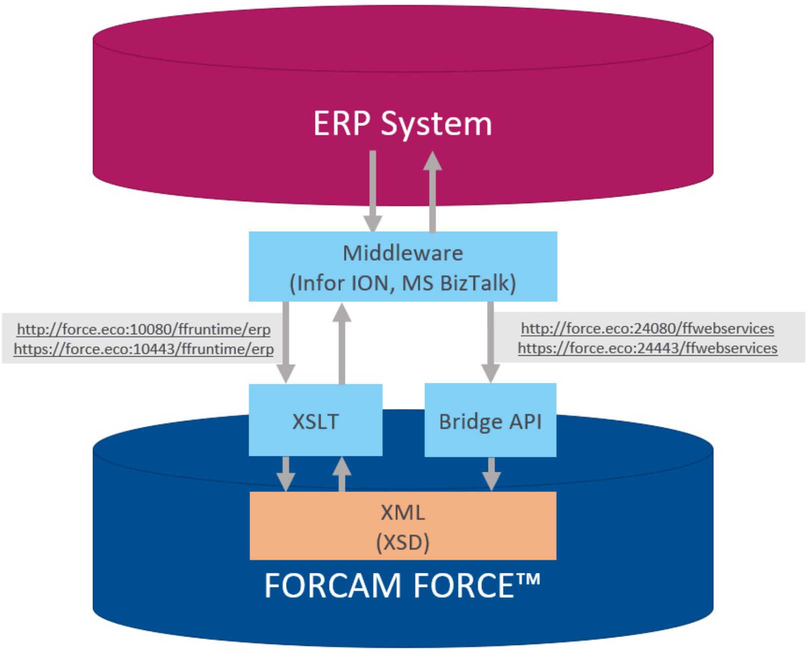

ERP System
Middleware
(Infor ION, MS BizTalk)
http://force.eco:10080/ffruntime/erp
https://force.eco:10443/ffruntime/erp
http://force.eco:24080/ffwebservices
https://force.eco:24443/ffwebservices
XSLT
Bridge API
XML
(XSD)
FORCAM FORCE™

**Fig. 1: Alternative connections for the ERP system with FORCAM FORCE™**

In addition, middleware such as Infor ION or Microsoft BIZ Talk can be used for the connection to the ERP. The functionality available there can be used to transform the required datasets into the FORCAM FORCE™ internal XML structure.

## 1.1 ERP Download and upload data service

A download and upload data service is available for the communication between FORCAM FORCE™ and the connected ERP system. The download service is used to supply the FORCAM FORCE™ Rule Engine with master data and order data. During this process, the incoming data (data message stream) is processed asynchronously. The upload data service is used for confirming quantities, times and states (status messages) of orders (event message stream). With durations and quantities, the absolute values are transmitted in addition to any relative changes.

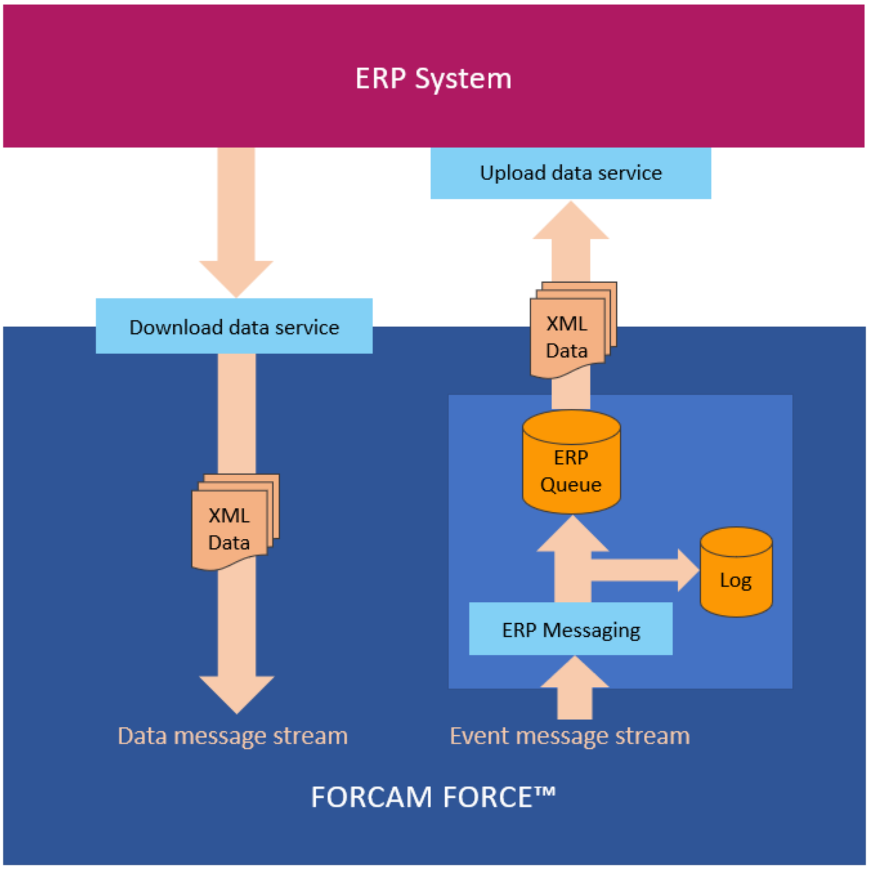


**Fig. 2: ERP Download and upload data service**

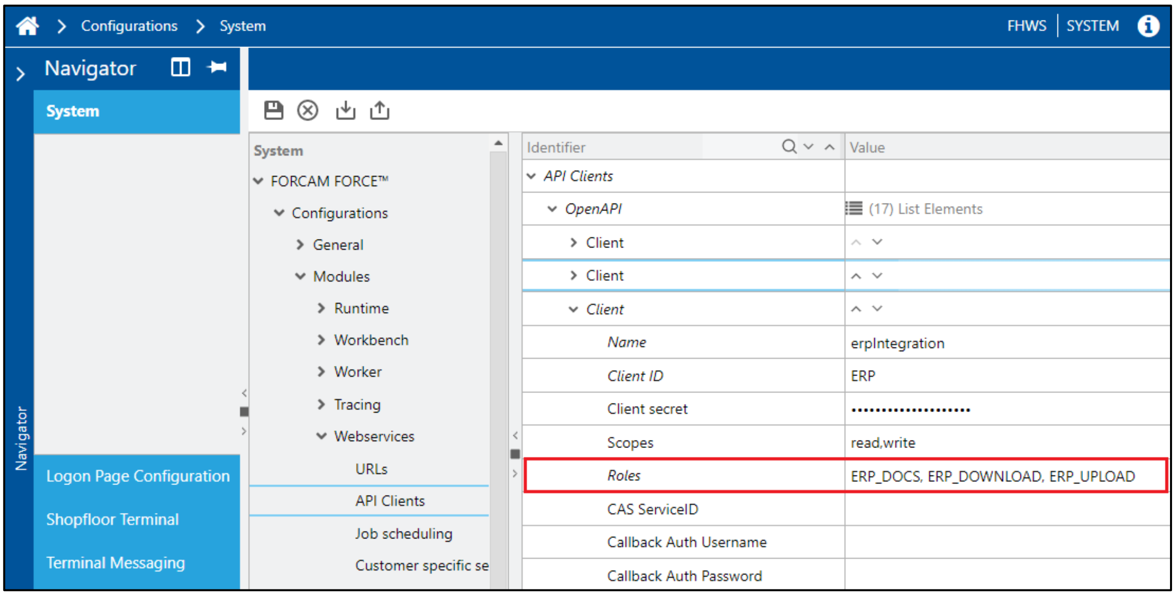
To prevent data loss in the event of system downtime, persistent queues are used for data transfer in the upload data service. Only messages successfully transmitted to the ERP are sent to the queue. This ensures system reliability and failed transfers remain in the queue.
A continuous log file is also used for the traceability of the communicated messages.

# 2 Configuration

## 2.1 ERP Download

Nothing needs to be configured within FORCAM FORCE™ for the download data service via runtime.

If data is to be transferred via Bridge API, a user with password and corresponding ERP user roles must be configured via the Workbench ( see **Manual - Bridge API**).



| Identifier | Value |
|---|---|
| API Clients | |
| OpenAPI | (17) List Elements |
| Client | |
| Client | |
| Client | |
| Name | erpIntegration |
| Client ID | ERP |
| Client secret | •••••••••••••••••••• |
| Scopes | read,write |
| Roles | ERP_DOCS, ERP_DOWNLOAD, ERP_UPLOAD |
| CAS ServiceID | |
| Callback Auth Username | |
| Callback Auth Password | |

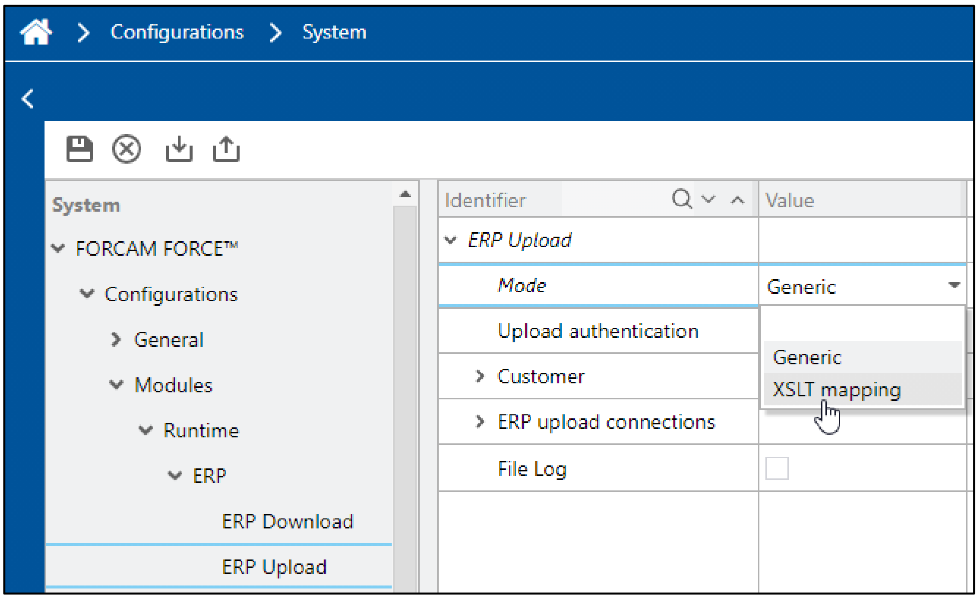
The following user roles can be selected under **Roles**:

| Role | Description |
|---|---|
| **ERP_DOCS** | Access to corresponding Swagger documentation (for ERP API). |
| **ERP_DOWNLOAD** | Access to download data service |
| **ERP_UPLOAD** | Access to upload data service |

## 2.2 ERP Upload

### 2.2.1 ERP Upload mode

The ERP upload mode determines how the XML records are transformed into the specified format of the ERP system.
The following drop-down menu is used to specify whether the transformation of XML records should be based on XSLT (for SAP IDoc format) or generic (for other XML formats).



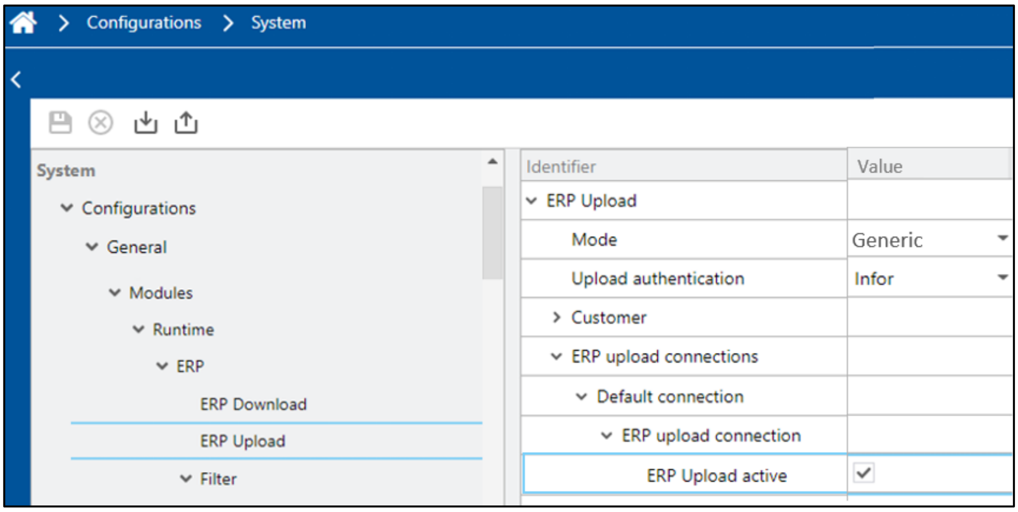
System
- FORCAM FORCE™
  - Configurations
    - General
    - Modules
      - Runtime
        - ERP
          - ERP Download
          - ERP Upload

| Identifier | Value |
|---|---|
| ERP Upload | |
| Mode | Generic |
| Upload authentication | |
| Customer | |
| ERP upload connections | |
| File Log | □ |

**Fig. 3: Configuration ERP upload mode**

### 2.2.2 ERP Upload activation

The upload data service can be activated in the next menu by checking **ERP Upload active**.



System
- Configurations
  - General
    - Modules
      - Runtime
        - ERP
          - ERP Download
          - ERP Upload
            - Filter

| Identifier | Value |
|---|---|
| ERP Upload | |
| Mode | Generic |
| Upload authentication | Infor |
| Customer | |
| ERP upload connections | |
| Default connection | |
| ERP upload connection | |
| ERP Upload active | ✓ |

**Fig. 4: ERP Upload activation**

Besides the standard connection, it is also possible to set up and activate additional ERP upload connections to other ERP systems.

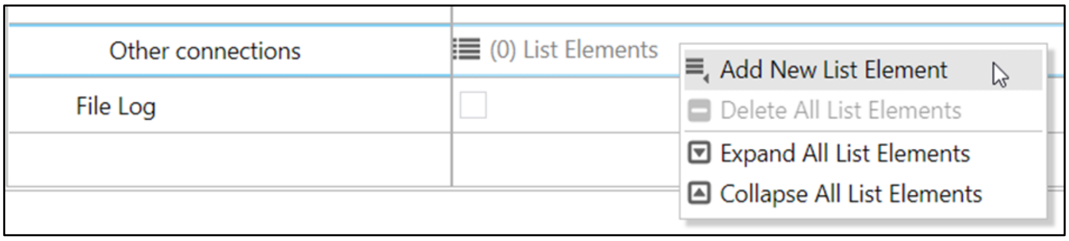


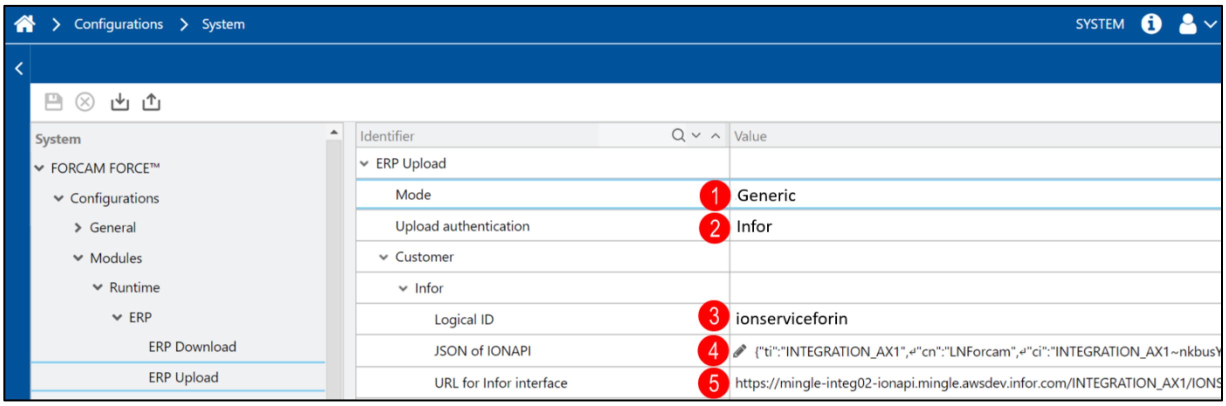
**Fig. 5: Activate new upload connection**

**To activate another upload connection:**

1. Go to More Connections and right-click on **List Items**.
2. Select **Add new list element**.
3. Expand the new upload connection and configure the new feedback and serialization interface according to the specifications of the additional ERP system.
4. Activate the additional connection by checking **ERP Upload active**.

### 2.2.3 Connection via Infor ION

If Infor ION is used for the connection, the following configuration must be made in the Workbench of FORCAM FORCE™:



| Identifier | Value |
|---|---|
| ERP Upload | |
| Mode | 1 Generic |
| Upload authentication | 2 Infor |
| Customer | |
| Infor | |
| Logical ID | 3 ionserviceforin |
| JSON of IONAPI | 4 {"ti":"INTEGRATION_AX1",↵"cn":"LNForcam",↵"ci":"INTEGRATION_AX1~nkbusY |
| URL for Infor interface | 5 https://mingle-integ02-ionapi.mingle.awsdev.infor.com/INTEGRATION_AX1/IONS |

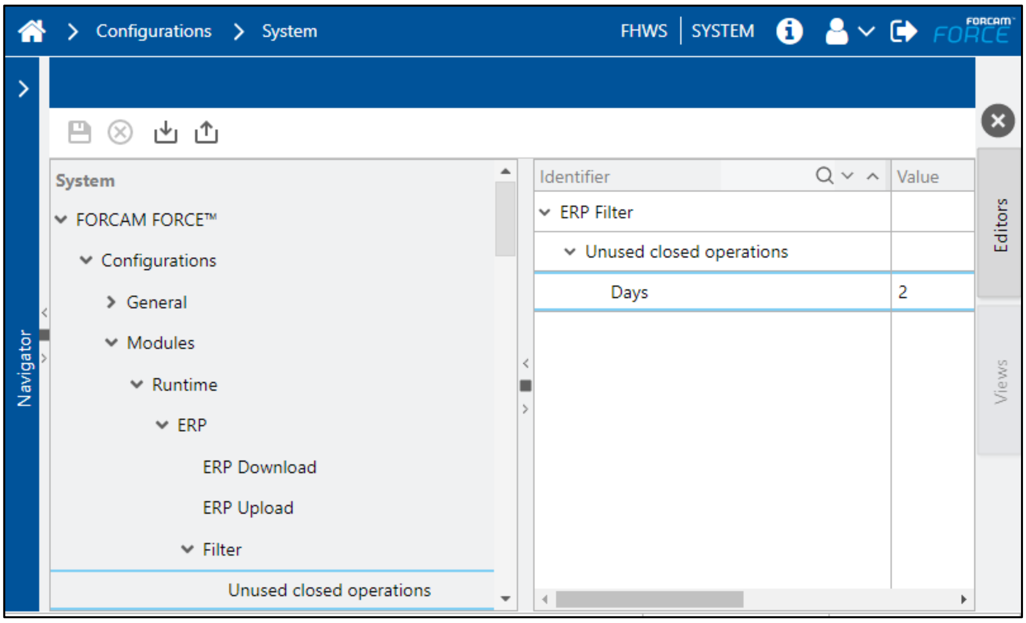
**Fig. 6: Configuration of upload data service for Infor ION**

**To configure the upload data service for Infor ION:**

1. Set the mode to **Generic** .
2. Set the Upload authentication to **Infor**.
3. Enter the Logical ID specified by Infor ERP.
4. Enter the JSON document specified by Infor ERP for the IONAPI.
5. Enter the URL for the Infor interface.

## 2.3 ERP Filter

Update messages can be filtered out or ignored for processes that have been completed for at least a certain period of time (in days). This period is preset to 2 days and can be adjusted here:



| Identifier | Value |
|---|---|
| ERP Filter | |
| Unused closed operations | |
| Days | 2 |

## 2.4 Procedure for assigning the operations to workplace groups

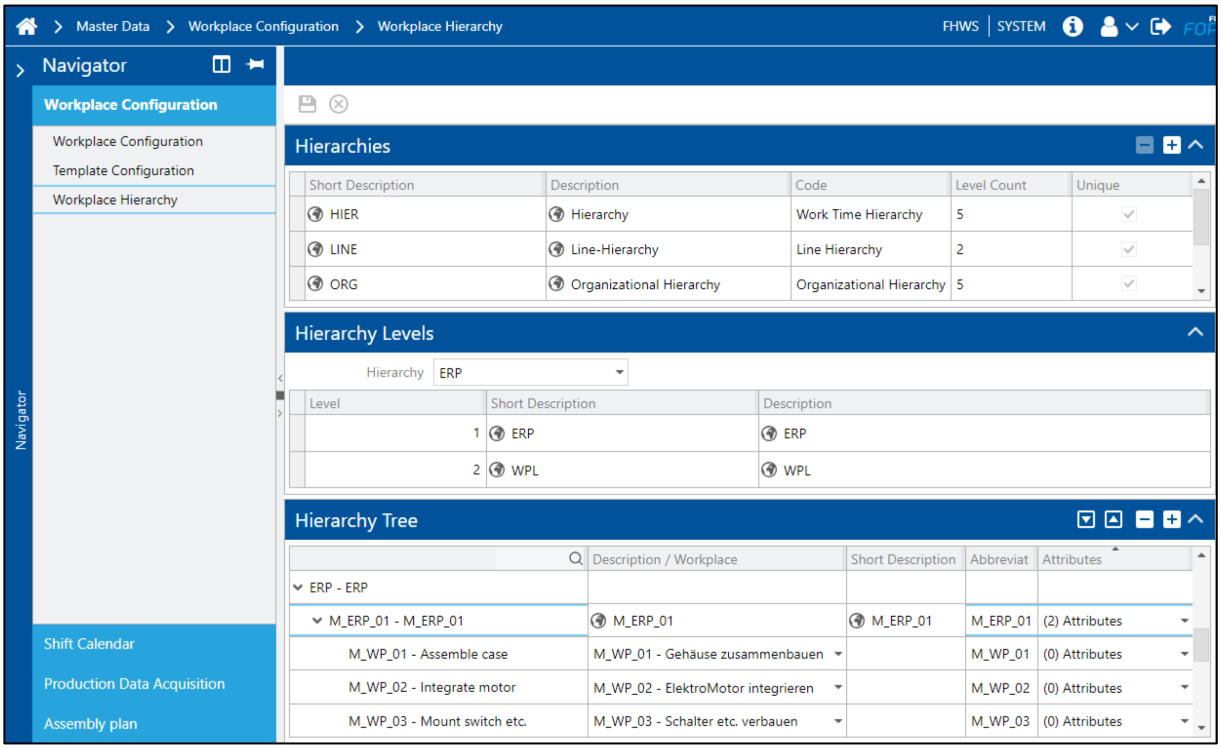
Usually, individual operations within a production order are assigned to a specific workplace with the attribute **TargetWorkplace.** However**,** FORCAM FORCE™ also offers advanced functions for dynamically assigning operations to workplaces within a grouping.
Workplaces can be arranged in any hierarchy in the Workbench using the workplace hierarchy functionality (see **Manual - Master Data and System Configuration**).
The ERP hierarchy can be used for dynamically assigning the operations. At the lowest level, workplaces can be grouped there as **workplace group**. If workplaces in such a grouping take over the same function, the grouping is also called **capacity group**. For a **capacity group**, you can specify how the dynamic assignment of the operations of released orders should be done.

It is possible to use the following procedures when assigning operations per capacity group:

| Procedure | Description |
|---|---|
| **AUTO split** | The operations are split among the workplaces within the capacity group based on the target quantity to be produced (**Operation split**). In the process, and as far as possible, each workplace receives the same proportion of the target quantity to be produced.<br>If the target quantity cannot be distributed equally among the number of workplaces, the remaining quantity is distributed among single workplaces. For example, with a target quantity of 100 pieces and 3 workstations, it results in a split of 33+33+34 pieces. |
| **Pull** | The operations are first available to all workplaces within the capacity group. The operation can then be assigned manually (e.g. by a foreman) to a workplace in the capacity group, or the first workplace that wants to start processing assigns the |

| Procedure | Description |
|---|---|
| | operation to itself. The assigned operation disappears in the shopfloor display of all other workplaces.<br>If an assigned procedure is canceled, it becomes available to all of them again and the procedure starts over from the beginning. |

The assignment procedure can be selected accordingly in the Workbench:



Hierarchies

| Short Description | Description | Code | Level Count | Unique |
|---|---|---|---|---|
| HIER | Hierarchy | Work Time Hierarchy | 5 | ✓ |
| LINE | Line-Hierarchy | Line Hierarchy | 2 | ✓ |
| ORG | Organizational Hierarchy | Organizational Hierarchy | 5 | ✓ |

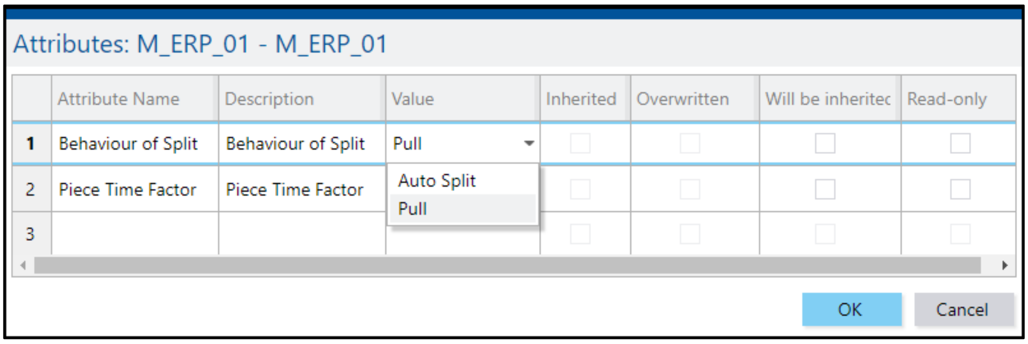
Hierarchy Levels

Hierarchy: ERP

| Level | Short Description | Description |
|---|---|---|
| 1 | ERP | ERP |
| 2 | WPL | WPL |

Hierarchy Tree

| | Description / Workplace | Short Description | Abbreviat | Attributes |
|---|---|---|---|---|
| ERP - ERP | | | | |
| M_ERP_01 - M_ERP_01 | M_ERP_01 | M_ERP_01 | M_ERP_01 | (2) Attributes |
| M_WP_01 - Assemble case | M_WP_01 - Gehäuse zusammenbauen | | M_WP_01 | (0) Attributes |
| M_WP_02 - Integrate motor | M_WP_02 - ElektroMotor integrieren | | M_WP_02 | (0) Attributes |
| M_WP_03 - Mount switch etc. | M_WP_03 - Schalter etc. verbauen | | M_WP_03 | (0) Attributes |



Attributes: M_ERP_01 - M_ERP_01

| | Attribute Name | Description | Value | Inherited | Overwritten | Will be inheritec | Read-only |
|---|---|---|---|---|---|---|---|
| 1 | Behaviour of Split | Behaviour of Split | Pull | □ | □ | □ | □ |
| 2 | Piece Time Factor | Piece Time Factor | | □ | □ | □ | □ |
| 3 | | | | □ | □ | □ | □ |

**Fig. 7: Procedure selection when assigning operations**

To assign operations to a workplace group in the **AUTO split** procedure:

1. Select the appropriate node in the lowest hierarchy level in the ERP hierarchy tree.
2. Open the Attributes menu in the **Attributes** column.
3. Select **AUTO split** for the attribute **Behavior of split.**

To assign the operations to a workplace group with the **Pull** procedure:

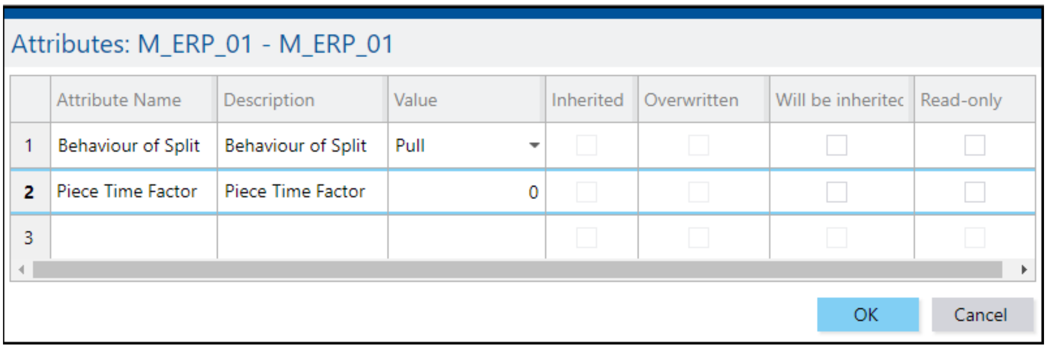
1. Select the appropriate node in the lowest hierarchy level in the ERP hierarchy tree.
2. Open the Attributes menu in the **Attributes** column.
3. Select **Pull** for the attribute **Behavior of split.**

### 2.4.1 Piece time factor

The piece time factor can only be used in combination with the **AUTO split** procedure. It is always used when the piece time at the workplace is not specified as **time per piece**, but as **time per s piece**. With the **unit time factor s** it is then possible to "counteract" so that the time per unit produced (TIME_PER_UNIT) is again calculated correctly:

**Time per unit = piece time * unit / piece time factor**

Different machining speeds may occur, especially in a workplace group of machines with different technologies. Thus, the piece time of a slower machine can be readjusted by using a piece time factor s < 1 and a faster machine by using a piece time factor s > 1 (e.g. a value of 2.0 for a machine producing twice-as-fast, or 0.75 for a 25% slower machine).
The default value of the piece time factor is 1 and will not lead to any adjustment, just like a value of 0.



Attributes: M_ERP_01 - M_ERP_01

| | Attribute Name | Description | Value | Inherited | Overwritten | Will be inheritec | Read-only |
|---|---|---|---|---|---|---|---|
| 1 | Behaviour of Split | Behaviour of Split | Pull | □ | □ | □ | □ |
| 2 | Piece Time Factor | Piece Time Factor | 0 | □ | □ | □ | □ |
| 3 | | | | □ | □ | □ | □ |

**Fig. 8: Setting the piece time factor**

To set the piece time factor:

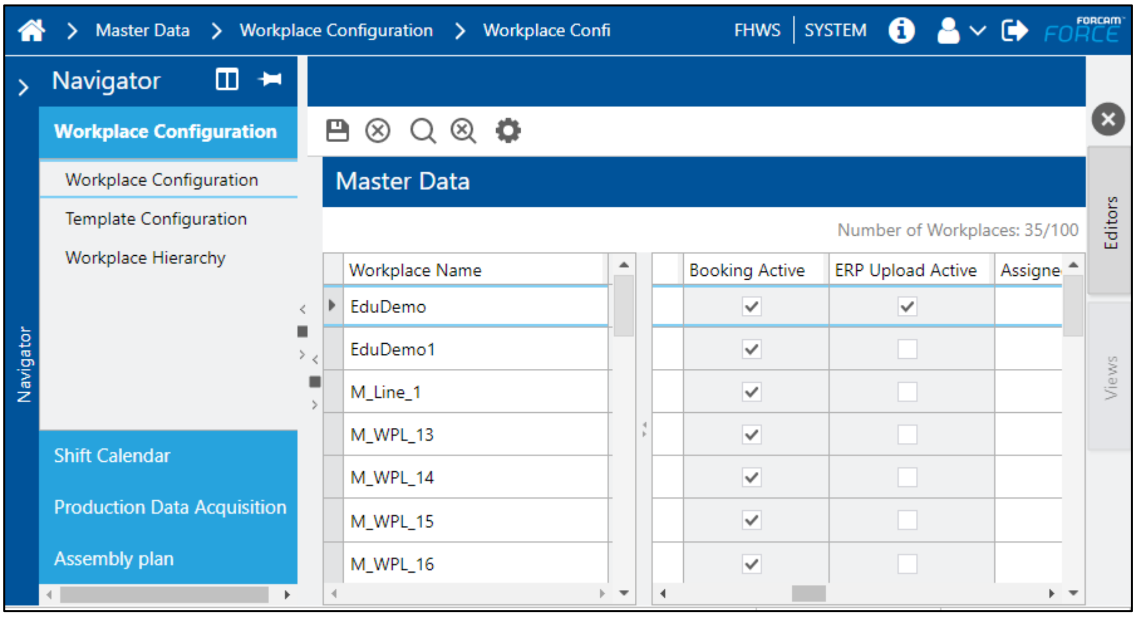
1. Select the appropriate node in the lowest hierarchy level in the ERP hierarchy tree.
2. Open the Attributes menu in the **Attributes** column.
3. Select the desired value for the attribute **Piece time factor**.

## 2.5 Workplace-specific configuration

### 2.5.1 ERP Upload activation

The upload data service can also be activated or deactivated per workplace via the **ERP upload active** setting.

ⓘ A temporary deactivation is usually applied when a new workplace is created. The ERP upload of the new workplace is only activated when the workplace is ready for operation. This prevents double bookings in the transition phase.



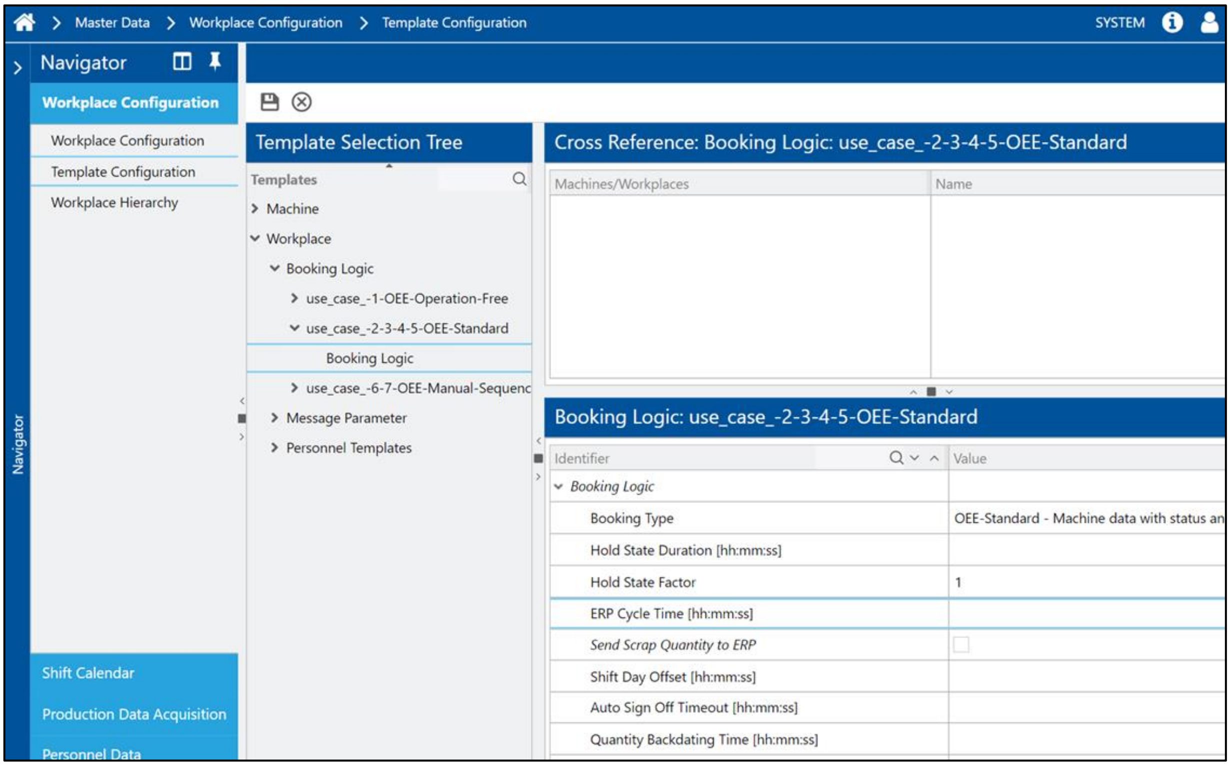
Number of Workplaces: 35/100

| Workplace Name | Booking Active | ERP Upload Active | Assigne |
|---|---|---|---|
| EduDemo | ☑ | ☑ | |
| EduDemo1 | ☑ | □ | |
| M_Line_1 | ☑ | □ | |
| M_WPL_13 | ☑ | □ | |
| M_WPL_14 | ☑ | □ | |
| M_WPL_15 | ☑ | □ | |
| M_WPL_16 | ☑ | □ | |

**Fig. 9: ERP upload activation per workplace**

## 2.5.2 ERP Cycle time

The time interval for ERP uploads regarding quantities can be set in the Workbench. Increasing the time interval is useful, for instance, if the system load caused by the ongoing data transfers is too high.



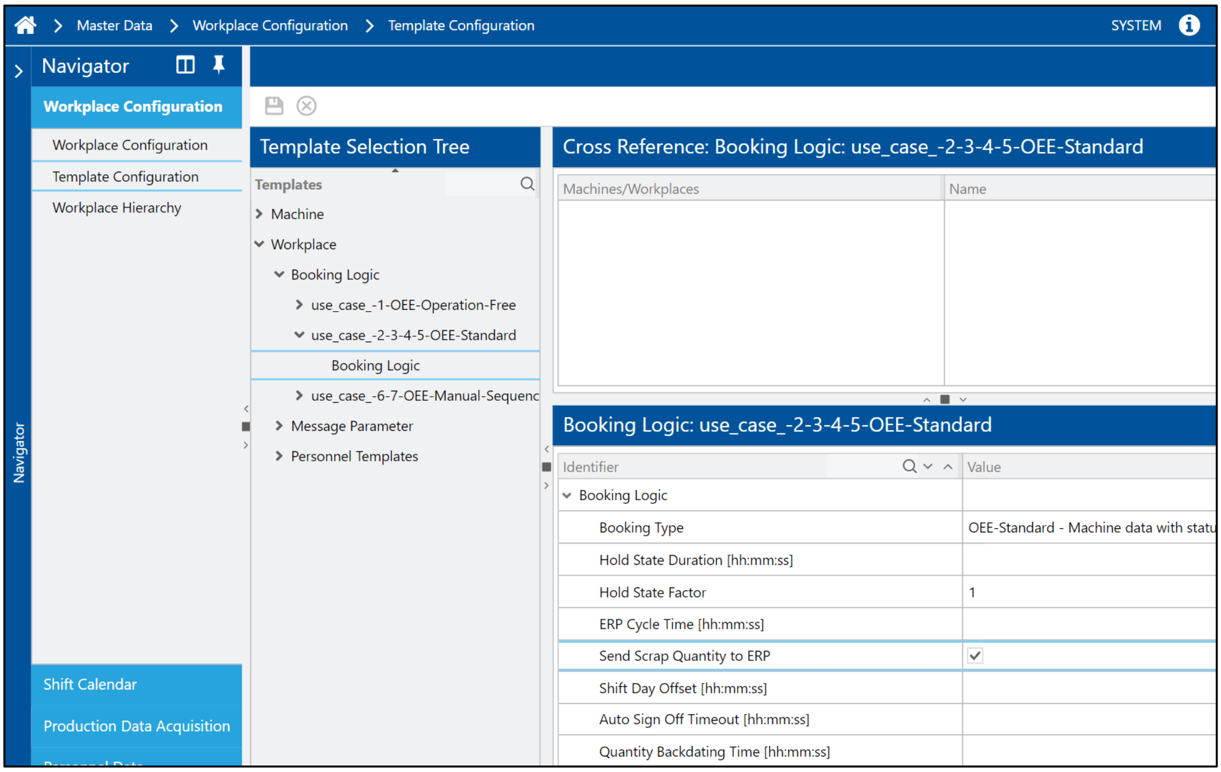
Booking Logic: use_case_-2-3-4-5-OEE-Standard

| Identifier | Value |
|---|---|
| *Booking Logic* | |
| Booking Type | OEE-Standard - Machine data with status an |
| Hold State Duration [hh:mm:ss] | |
| Hold State Factor | 1 |
| ERP Cycle Time [hh:mm:ss] | |
| *Send Scrap Quantity to ERP* | □ |
| Shift Day Offset [hh:mm:ss] | |
| Auto Sign Off Timeout [hh:mm:ss] | |
| Quantity Backdating Time [hh:mm:ss] | |

**Fig. 10: Setting the ERP cycle time**

### 2.5.3 Send scrap quantity to ERP

In case scrap quantity is posted, the quantity can be reported to the ERP system along with the scrap reason. It is also possible to activate or deactivate in the Workbench.

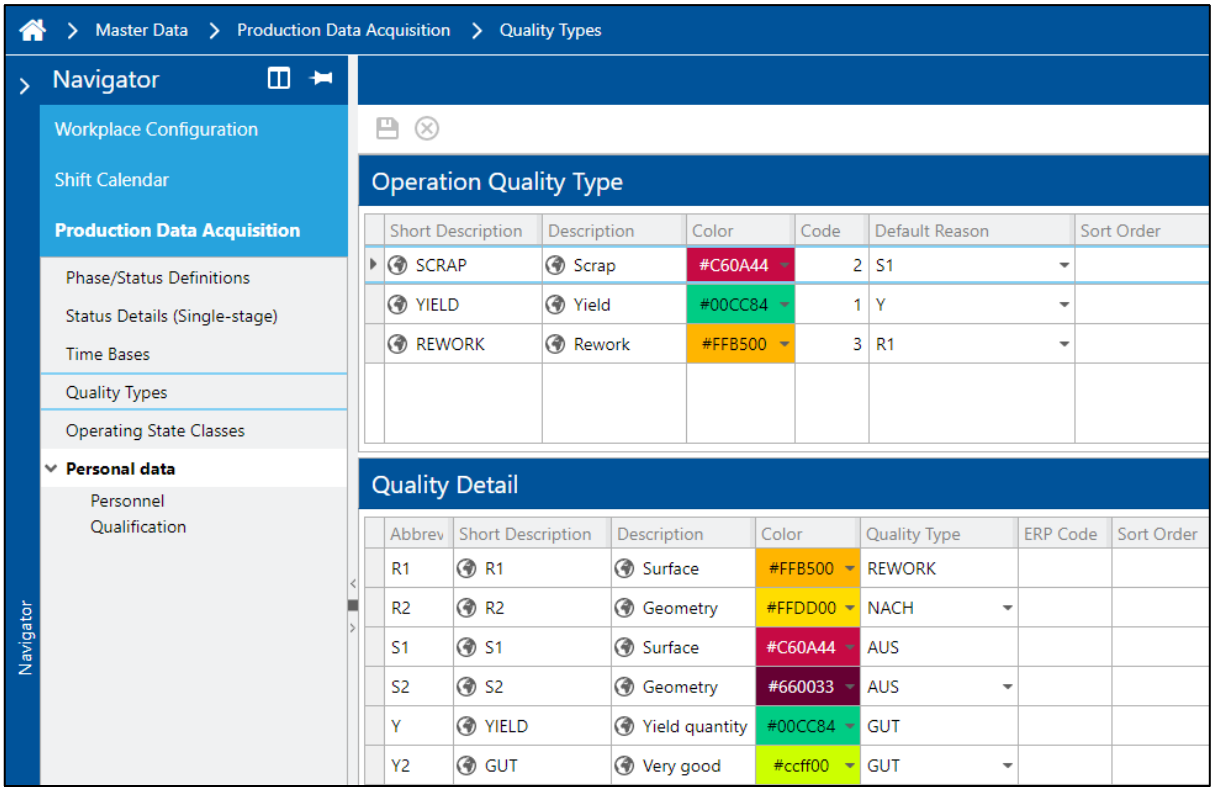


Booking Logic: use_case_-2-3-4-5-OEE-Standard

| Identifier | Value |
|---|---|
| Booking Logic | |
| Booking Type | OEE-Standard - Machine data with statu |
| Hold State Duration [hh:mm:ss] | |
| Hold State Factor | 1 |
| ERP Cycle Time [hh:mm:ss] | |
| Send Scrap Quantity to ERP | ☑ |
| Shift Day Offset [hh:mm:ss] | |
| Auto Sign Off Timeout [hh:mm:ss] | |
| Quantity Backdating Time [hh:mm:ss] | |

**Fig. 11: Send scrap quantity to ERP**

The scrap reason must coincide with the scrap reasons in the ERP system. ERP scrap reasons should be maintained in the workbench master data:



Operation Quality Type

| Short Description | Description | Color | Code | Default Reason | Sort Order |
|---|---|---|---|---|---|
| SCRAP | Scrap | #C60A44 | 2 | S1 | |
| YIELD | Yield | #00CC84 | 1 | Y | |
| REWORK | Rework | #FFB500 | 3 | R1 | |

Quality Detail

| Abbrev | Short Description | Description | Color | Quality Type | ERP Code | Sort Order |
|---|---|---|---|---|---|---|
| R1 | R1 | Surface | #FFB500 | REWORK | | |
| R2 | R2 | Geometry | #FFDD00 | NACH | | |
| S1 | S1 | Surface | #C60A44 | AUS | | |
| S2 | S2 | Geometry | #660033 | AUS | | |
| Y | YIELD | Yield quantity | #00CC84 | GUT | | |
| Y2 | GUT | Very good | #ccff00 | GUT | | |

**Fig. 12: Managing scrap reasons**

# 3 Data structures

## 3.1 XSL Basic data types

The transformation of the transmitted XML data is governed via XSDs (XML Schema Definitions), which are hardcoded into FORCAM FORCE™. XSDs contain information about the setup of the data structures used. Data elements or attributes are specified based on data types that are available by default in XSL. This includes the following data types:

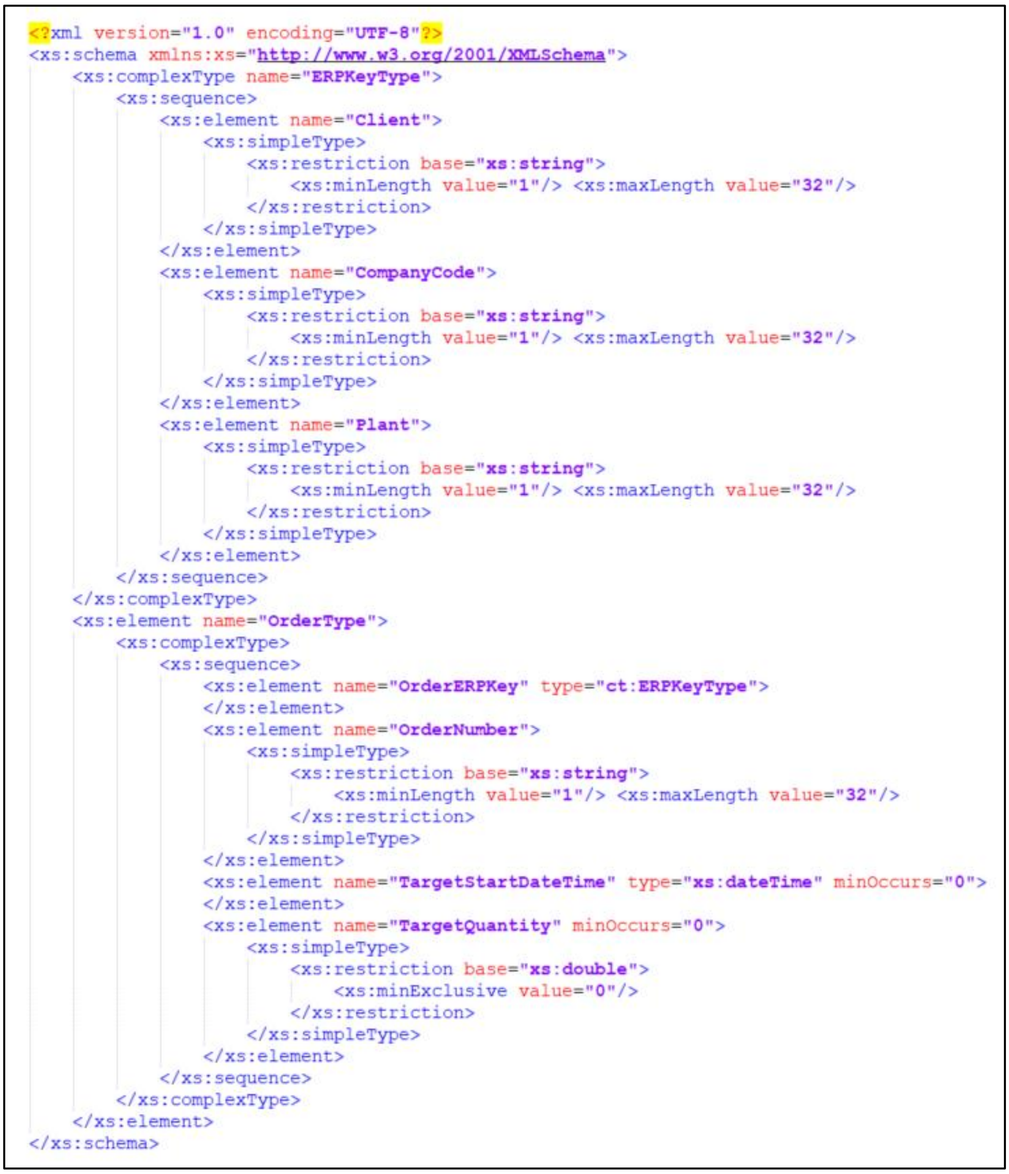
**Table 1: Applied XSL basic data types**

| Data type | Application |
|---|---|
| **xs:string** | String |
| **xs:boolean** | Flag |
| **xs:long** | Integer |
| **xs:double** | Floating-point number |
| **xs:date** | Date specified in format:<br>**YYYY-MM-DD**<br>**Y** = Year<br>**M** = Month<br>**D** = Day<br><br>Example: 2021-01-30 |
| **xs:dateTime** | Date specified in format:<br>**YYYY-MM-DDTHH:mm:SS**<br>**Y** = Year<br>**M** = Month<br>**D** = Day<br>**H** = Hour<br>**m** = Minute<br>**S** = Second<br><br>Example: 2021-01-30T09:10:20 |
| **xs:duration** | Time interval in format:<br>P**y**Y**m**M**d**DT**h**H**m**M**s**S<br>**y** = Years<br>**m** = Months<br>**d** = Days<br>**h** = Hours<br>**m** = Minutes<br>**s** = Seconds<br><br>Examples:<br>P1Y11M30D<br>P30D23H59M45S |

## 3.2 XSL complex data types

Complex data types can be formed from the XSL basic data types via XSL. This is done by combining several data elements or attributes in a hierarchical structure. These formed complex data types are in turn used for creating application-specific data structures. Examples of application-specific data structures in FORCAM FORCE™ are production orders, personnel data or shifts.

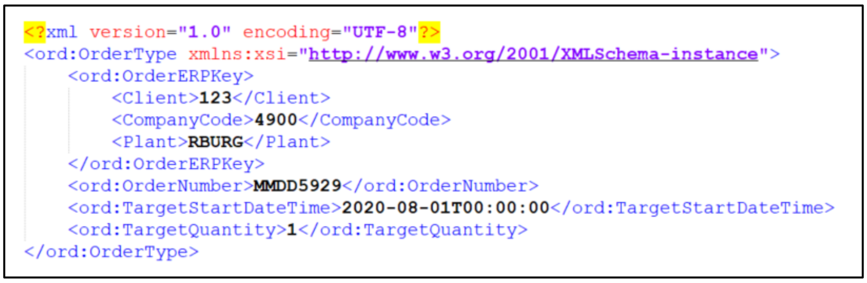
For example, an XSD for ERP keys (**ERPKeyType)** and a data type for orders (**OrderType)** based on it looks like this:



```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:complexType name="ERPKeyType">
        <xs:sequence>
            <xs:element name="Client">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:minLength value="1"/> <xs:maxLength value="32"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="CompanyCode">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:minLength value="1"/> <xs:maxLength value="32"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="Plant">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:minLength value="1"/> <xs:maxLength value="32"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="OrderType">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="OrderERPKey" type="ct:ERPKeyType">
                </xs:element>
                <xs:element name="OrderNumber">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:minLength value="1"/> <xs:maxLength value="32"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="TargetStartDateTime" type="xs:dateTime" minOccurs="0">
                </xs:element>
                <xs:element name="TargetQuantity" minOccurs="0">
                    <xs:simpleType>
                        <xs:restriction base="xs:double">
                            <xs:minExclusive value="0"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

**Fig. 13: Example XSD for the data types ERPKeyType and OrderType**

XML records of an order created according to this template (of data type **OrderType**) could then look like this:



```
<?xml version="1.0" encoding="UTF-8"?>
<ord:OrderType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ord:OrderERPKey>
        <Client>123</Client>
        <CompanyCode>4900</CompanyCode>
        <Plant>RBURG</Plant>
    </ord:OrderERPKey>
    <ord:OrderNumber>MMDD5929</ord:OrderNumber>
    <ord:TargetStartDateTime>2020-08-01T00:00:00</ord:TargetStartDateTime>
    <ord:TargetQuantity>1</ord:TargetQuantity>
</ord:OrderType>
```

**Fig. 14: Example XML data set of a production order**

Thus, an XSD provides the information for the structure and arrangement of the elements or attributes of a data set in XML format.

This includes the following information:

- Number and sequence of elements and child elements
- Data types of the elements
- Preset and consistent values for the elements

The following chapters describe the data structures or XSDs used in FORCAM FORCE™ in more detail. A differentiation is made between download and upload service.

# 4 Download

The download data service provides FORCAM FORCE™ with the following data sets relevant for production:

- Orders
  - Order header
  - Operations
  - Components
  - Production resources and tools
  - Product characteristics of the material
- Shift data
- Personnel data
- Overhead costs
- Warehouse data from warehouse management system
- Response to quantity requests

These data sets are transmitted as a sequence of attributes and their structure is clearly specified via respective XSDs.

ⓘ Only the order header data and the corresponding operations are mandatory. The other data sets are only required for display purposes on the FORCE user interface or for functional extensions such as tool data management (TDM), document management and the container trace.

ⓘ In case of an error during data processing, the data is rolled back, i.e. the processing of the message is discarded.

It is mandatory to specify several attributes. This will depend on whether the data set action is a new entry, a modification, or a deletion. This is shown in the following tables on the right wherever relevant.

## 4.1 Download basic data types

FORCAM FORCE™ uses special data types in addition to the standard data types available in XSL, which are also used for specifying download data sets.

### 4.1.1 Time fields

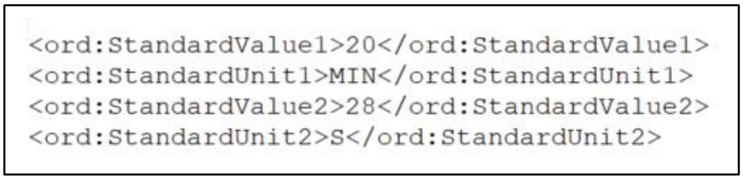
In addition to the **xs:dateTime** data type already present in XLS (see Table 1), time specifications are also transmitted by way of a sequence of time fields. For this, FORCAM FORCE™  has an XSD for the complex data type **ct:timeUnitType**.

The definition of a time field always requires the definition of a time value and an associated time unit.
Based on this unit, the transmitted value of the time field is converted to a FORCAM FORCE™ internal common time base value. Without a valid time unit, it is not possible to ensure that the time value is converted and interpreted correctly.
The following units for time values are supported for ERP download:

**Table 2: Supported time units for time fields**

| Time unit | Description |
|---|---|
| **MS** | Millisecond |
| **S** | Second |
| **MIN** | Minute |
| **HR** | Hour |
| **H** | Hour |
| **M90** | Minute / 90 |
| **TMU** | Technical minute |

An XML message with a time specification created based on time fields may look like this:



```
<ord:StandardValue1>20</ord:StandardValue1>
<ord:StandardUnit1>MIN</ord:StandardUnit1>
<ord:StandardValue2>28</ord:StandardValue2>
<ord:StandardUnit2>S</ord:StandardUnit2>
```

**Fig. 15: Example of a time specification based on time fields**

### 4.1.2 Multilingual texts

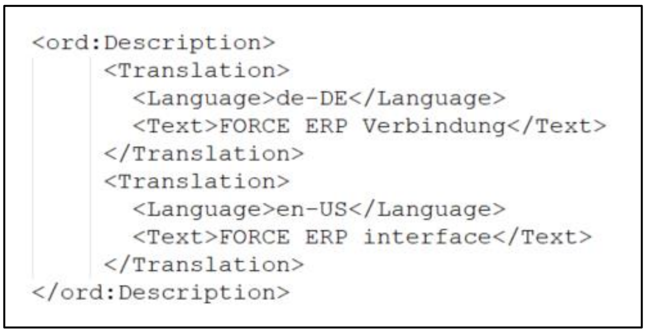
Texts can generally be transmitted as multilingual texts. At present, FORCAM FORCE™ supports the following languages on the graphical user interface:

- German
- English (US)
- Englisch (GB)
- French
- Chinese

**For this purpose, FORCAM FORCE™<t0/> has <t1/>XSDs for the following complex data types: table 3: Data types for multilingual texts.**

| Data type | Application |
|---|---|
| **ct:SupportedLanguagesType** | Supported language with the following possible values (codes) according to ISO 639:<br>de-DE<br>en-US<br>en-GB<br>fr-FR<br>zh-CN |
| **ct:MultilanguageTextType** | Multilingual text as a sequence of language codes and associated texts. |

An XML message with a multilingual text that was created based on these data types may look like this:



```
<ord:Description>
     <Translation>
       <Language>de-DE</Language>
       <Text>FORCE ERP Verbindung</Text>
     </Translation>
     <Translation>
       <Language>en-US</Language>
       <Text>FORCE ERP interface</Text>
     </Translation>
</ord:Description>
```

**Fig. 16: Example of a multilingual text**

If a text translation is missing, the text of the default language that is defined in the system will be used.
If this text is missing as well, the language codes are sorted in descending order and the translation of the first language code is taken.

**Table 4: Example of text replacement when there is no translation in the default language**

| Configured languages | Default | Transmitted translations | Adopted translations |
|---|---|---|---|
| **de-DE** | x | n/a | Englisch (GB) |
| **en-GB** | | Englisch (GB) | Englisch (GB) |
| **en-US** | | English (US) | English (US) |
| **fr-FR** | | n/a | Englisch (GB) |

### 4.1.3 Dataset action

Whether the transmitted dataset is to serve a new entry, a change or a deletion of the corresponding dataset in FORCAM FORCE™ is controlled by an attribute of the complex data type **ct:actionType**.

**Table 5: Possible actions during the data transmission**

| Action | Description |
|---|---|
| **ADD** | Create/change dataset |
| **DELETE** | Delete dataset |
| **NOP** | Do not adopt dataset.<br>This does not apply to child elements. This means that specific subordinate attributes can be transferred for processing; superordinate attributes are not transferred. |

### 4.1.4 ERP key

An ERP key is an individual key for the assignment of orders, machines, personnel or shifts to a client, company code or company building in the ERP system. ERP keys can be configured in the Workbench to match the ERP system. To transfer an ERP key, the complex data type **ct:ERPKeyType** is used with the following attributes:

**Table 6: Attributes in the ERP key (see also ERPDownloadOrder.XSD)**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **Client** | Client | xs:string(32) | x | x |
| **CompanyCode** | Company code | xs:string(32) | x | x |
| **Plant** | Plant | xs:string(32) | x | x |
| **SystemId** | System ID | xs:string(32) | x | |

The company code can be used as a company's financial accounting unit to distinguish the business transaction.

The **System ID** is used for unique identification of the ERP key in the system. This is important in case the attributes client, company code and plant are not unique. So the system ID can be used e.g. to distinguish connections between the test system and the productive system.

The following chapters describe in more detail where the ERP key is applied.

### 4.1.5 Material

The complex data type **ct:MaterialType** is available for defining and transmitting a material with the following attributes:

**Table 7: Attributes in dataset material**

| Attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **Description** | Description | ct:MultilanguageTextType | x | |
| **Number** | Material number | xs:string(255) | x | |
| **Type** | Material type | xs:string(255) | | |

The material number, also called part number, is primarily used to identify the material. Specifying the material type is optional.
In FORCAM FORCE™, the term material serves as a general term for the processed parts in manufacturing. Material is processed (e.g. machined or installed) at a workplace and new material can be created from it (but does not have to be). Any materials entering this processing stage at the workplace are referred to as components if a different or new material is created in the process. If a material is only processed at the workplace, e.g. milled or shaped, it is still only referred to as material.

So there is material that is composed of several components. In this context, the material to be manufactured requires a minimum number of these components in order to be correctly assembled and result in the desired product. This forms the bill of material, which can be transmitted with the operations (see section 4.2.3 and 4.2.4).
For example, if the desired end material is a cylinder engine. Simplified, it consists of 4 cylinders, 4 screws and 1 crankshaft. These parts then form the components of the material cylinder engine. This is also known as a parts list.

When a material is transmitted to FORCE for the first time, it is stored in a FORCE internal material list and is then searchable (e.g. in the performance analysis of the Office Client). For this reason it is useful to transmit not only the material number, but also a material type and a suitable description. For a more detailed specification, it is also possible to use the characteristics according to section 4.2.2.

### 4.1.6 User data fields

The complex data type **ct:UserDataType** is available for the transmission of user-defined data and its display on the FORCAM FORCE™ user interface with the following freely usable attributes:

**Table 8: Attributes in the dataset user-defined data fields**

| Attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **UF1** | User field 1 | xs:string(255) | | |
| **UF2** | User field 2 | xs:string(255) | | |
| **…** | | | | |
| **UF50** | User field 50 | xs:string(255) | | |

## 4.2 Order data

An XSD is used for orders being supplied from the ERP.
This XSD controls the transmission of the job header data and the following subordinate objects:

- Operations of the order
- Product characteristics of the material used
- Components used
- Production resources and tools used

An individual XML dataset is created for each of the order header data and subordinate objects.

ⓘ Components are only necessary if the components to be installed are to be displayed in FORCAM FORCE™ or reported back to the ERP.

ⓘ Production resources and tools are only necessary if they are required for display in FORCAM FORCE™ or if Tool Data Management (TDM) is used.

### 4.2.1 Order header

The most important components in the **order header** dataset are:

- The **ERP key** and the **order number** to identify the order
- The **target start date** and **target end date** as time specifications for the production
- The **produced material** (final product) of the order
- The **target quantity with quantity unit** as the quantity specification of the material produced

**Table 9: Attributes in order header dataset**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| Name | Description | Data type (Field length) | Input/ Change | Delete |
| Action | Dataset action | ct:actionType | x | x |
| ERPKey | ERP key | ct:ERPKeyType | x | x |
| OrderNumber | Order number | xs:string(32) | x | x |
| OrderSplit | Order split | xs:string(8) | x | x |
| ERPStatusIds | Status IDs from ERP system (with space as separator) | xs:string(255) | | x |
| ERPStatusCodes | Various order status codes corresponding with the ERP system (with space as separator) | xs:string(255) | | |
| ERPStatusCodesTranslation | Translation for status codes (with space as separator) | xs:string(255) | | |
| OrderPriority | Priority | xs:string(255) | | |
| ReleaseDateTime | Release date | xs:dateTime | | |
| TargetStartDateTime | Target start date | xs:dateTime | x | |
| TargetEndDateTime | Target end date | xs:dateTime | x | |
| TargetQuantity | Target quantity | xs:double | x | |
| DisplayQuantityUnit | Quantity unit | xs:string(255) | x | |
| ProductionVersion | Production version | xs:string(255) | | |

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **PlannerGroupInCharge** | Responsible planning group/department | xs:string(255) | | |
| **Description** | Description | ct:MultilanguageTextType | | |
| **ScheduledStartDate** | Scheduled start date | xs:dateTime | | |
| **ScheduledEndDate** | Scheduled end date | xs:dateTime | | |
| **MaterialCharacteristics** | Sequence of product characteristics of the produced material | sequence of ord:MaterialCharacteristicType | | |
| **UserDataFields** | User data fields | ct:UserDataType | | |
| **Operations** | Sequence of operations | Sequence of ord:OperationType | x | |
| **Material** | Produced material | ct:MaterialType | x | |

### 4.2.2 Product characteristics of the material

The complex data type **ord:MaterialCharacteristicType** is available for the defining and transmitting of the dataset product characteristics. This has the following attributes:

**Table 10: Attributes in the dataset product characteristics**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **orderItem** | Order item | xs:string(255) | x | |
| **classType** | Label class type | xs:string(255) | x | |
| **characteristicName** | Characteristic name | xs:string(255) | x | |
| **characteristicValue1** | Characteristic value 1 | xs:string(255) | | |
| **characteristicValue2** | Characteristic value 2 | xs:string(255) | | |
| **characteristicValue3** | Characteristic value 3 | xs:string(255) | | |
| **characteristicValue4** | Characteristic value 4 | xs:string(255) | | |
| **characteristicValue5** | Characteristic value 5 | xs:string(255) | | |
| **characteristicValue6** | Characteristic value 6 | xs:string(255) | | |
| **characteristicValue7** | Characteristic value 7 | xs:string(255) | | |
| **characteristicValue8** | Characteristic value 8 | xs:string(255) | | |
| **characteristicValue9** | Characteristic value 9 | xs:string(255) | | |
| **characteristicValue10** | Characteristic value 10 | xs:string(255) | | |
| **characteristicValue11** | Characteristic value 11 | xs:string(255) | | |
| **characteristicValue12** | Characteristic value 12 | xs:string(255) | | |
| **characteristicValue13** | Characteristic value 13 | xs:string(255) | | |

### 4.2.3 Operations

The operations belonging to the order are transmitted as a sequence of individual operations. The main components of an operation are:

- **ERP key**, **operation number** and **long text** to identify the operation
- **Workplace** or **workplace group** for the production
- **Operation split** for defining the assignment to a workplace of the workplace group (capacity group in the ERP hierarchy) in the AUTO split procedure
- **Confirmation number** used as identifier for the ERP system in order to reassign a later confirmation to the order or operation
- The **target quantity** with quantity unit as the quantity specification of the material to be produced
- Planned **start date** and **finish date** as time limits for the production of the material
- **Setup time** and **time per unit** with time unit as additional time setting for the production of the material
- **Time per unit factor** to make a **quantity unit calculation per unit**
- **Bill of matieral** for defining the required components
- **Production resources/tools** for defining the required resources/tools

The time per unit factor can be used to adjust the **time per unit quantity** of the machine technology:
**Time per unit = piece time / piece time factor**

The complex data type **ord:OperationType** is used for defining and transmitting an operation. This has the following attributes:

**Table 11: Attributes in dataset operations**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| Name | Description | Data type (Field length) | Input/ Change | Delete |
| Action | Dataset action | ct:actionType | x | x |
| OrderERPKey | ERP key | ct:ERPKeyType | x | x |
| OperationNumber | Operation number | xs:string(32) | x | x |
| OperationSplit | Operation split<br>– 0: Without operation split (pull procedure)<br>– 1: Assignment to 1st workplace of the workplace group capacity group<br>– 2: Assignment to 2nd workplace of the workplace group<br>– 3: Etc. | xs:string(8) | x | x |

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **ConfirmationNumber** | Confirmation number | xs:string(255) | x | x |
| **ERPYieldQuantity** | Reported yield | xs:double | | |
| **ERPReworkQuantity** | Reported rework | xs:double | | |
| **ERPScrapQuantity** | Reported scrap quantity | xs:double | | |
| **ERPPlannedScrapQuantity** | Planned scrap quantity | xs:double | | |
| **ERPStatusIds** | Status IDs in the ERP system (with spaces as separators) | xs:string(255) | | x |
| **ERPStatusCodes** | Status codes in the ERP system (with spaces as separators) | xs:string(255) | | |
| **ERPStatusCodesTranslation** | Translation for status codes | xs:string(255) | | |
| **Description** | Description | ct:MultilanguageTextType | | |
| **OperationText** | Long text | ct:MultilanguageTextType | x | |
| **ControlKey** | Control key | xs:string(255) | | |
| **OrderType** | Order type | xs:string(255) | | |
| **FunctionType** | Function of the operation<br>– 1: Normal<br>– 2: Training<br>– 3: Maintenance | xs:long | | |
| **LeadingOperation** | Leading operation within operation block (yes/no)<br>– True<br>– False | xs:boolean | | |
| **BlockNumber** | Operation block | xs:string(255) | | |
| **BlockDescription** | Description of operation block | ct:MultilanguageTextType | | |
| **OperationSequence** | Order sequence | xs:string(255) | | |
| **AutoStartId** | ID that comes from the machine to automatically start the process | xs:string | | |
| **TargetWorkplace** | Workplace for the production | xs:string(255) | x | |

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| Name | Description | Data type (Field length) | Input/ Change | Delete |
| TargetWorkGroup | Workplace group ID of the workplace | xs:string | | |
| WorkplaceGroup | Workplace group of the workplace | xs:string(255) | | |
| TargetStartDateTime | Planned start date | xs:dateTime | x | |
| TargetEndDateTime | Planned end date | xs:dateTime | x | |
| DefaultStrokeFactor | Stroke factor<br>The quantity produced per stroke or per tact/production cycle | xs:double | | |
| DefaultPieceTimeFactor | Piece time factor<br>Used for calculating<br>Piece time per quantity unit | xs:double | x | |
| DisplayQuantityUnit | Quantity unit | xs:string(255) | x | |
| TargetQuantity | Target quantity | xs:double | x | |
| StandardValue1 | Target setup time | xs:string(255) | x | |
| StandardUnit1 | Target unit setup time | ct:timeUnitType | x | |
| StandardValue2 | Time per unit | xs:string(255) | x | |
| StandardUnit2 | Time per unit | ct:timeUnitType | x | |
| StandardValue3 | Default value 3 | xs:string(255) | | |
| StandardUnit3 | Default value unit 3 | ct:timeUnitType | | |
| StandardValue4 | Default value 4 | xs:string(255) | | |
| StandardUnit4 | Default value unit 4 | ct:timeUnitType | | |
| StandardValue5 | Default value 5 | xs:string(255) | | |
| StandardUnit5 | Default value unit 5 | ct:timeUnitType | | |
| StandardValue6 | Default value 6 | xs:string(255) | | |
| StandardUnit6 | Default value unit 6 | ct:timeUnitType | | |
| AlternateOperationNumber | Alternative operation number | xs:string(255) | | |

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| Name | Description | Data type (Field length) | Input/ Change | Delete |
| ProductionVersion | Production version | xs:string(255) | | |
| CounterNumber | Counter number (Press station counter) | xs:long | | |
| UserStatus | User state in the ERP system | xs:string(255) | | |
| OverdeliveryCheck | Check for compliance with the overdelivery tolerance | xs:boolean | | |
| OverdeliveryQuantity | Max overdelivery quantity amount (check overdelivery quantity against booked quantity - target quantity) | xs:double | | |
| UnderdeliveryCheck | Check for compliance with the underdelivery tolerance | xs:boolean | | |
| UnderdeliveryQuantity | Min underdelivery quantity (check underdelivery quantity against target quantity - booked quantity) | xs:double | | |
| DefaultTransportQuantity | Preset quantity for material movements in Track & Trace | xs:double | | |
| QualityCheckCycle | The produced quantity after which a quality inspection is to be carried out | xs:double | | |
| StandardPackSize | Maximum quantity that can be produced into the output container (Track&Trace) | xs:double | | |
| DefaultComponentConsumptionFactor | Defines the default consumption quantity of a component per output quantity if this component is not listed in the BOM. | xs:double | | |
| EarliestStartDate | Earliest start date | xs:dateTime | | |
| LatestStartDate | Latest start date | xs:dateTime | | |
| EarliestEndDate | Earliest end date | xs:dateTime | | |
| LatestEndDate | Latest end date | xs:dateTime | | |
| ScheduledStartDate | Scheduled start date | xs:dateTime | | |
| ScheduledEndDate | Scheduled end date | xs:dateTime | | |
| TargetQueueTime | Target queue time, default duration of the queue time in ms | xs:long | | |

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **TargetQueueTimeUnit** | Target queue time unit | ct:timeUnitType | | |
| **TargetProcessingTime** | Target processing time, default duration of the processing time in ms:<br>When processing time does not depend on the number of pieces. Example: Baking 1 roll takes 30 minutes. If 3 rolls are put in the oven, it would not take any longer. | xs:long | | |
| **TargetProcessingTimeUnit** | Target processing time unit | ct:timeUnitType | | |
| **TargetTeardownTime** | Target teardown time, default duration of teardown time (in ms) | xs:long | | |
| **TargetTeardownTimeUnit** | Target teardown time unit | ct:timeUnitType | | |
| **TargetWaitTime** | Target wait time, target duration of the wait time | xs:long | | |
| **TargetWaitTimeUnit** | Target wait time unit | ct:timeUnitType | | |
| **TargetMoveTime** | Target move time | xs:long | | |
| **TargetMoveTimeUnit** | Target move time unit | ct:timeUnitType | | |
| **MinimumSendAheadQuantity** | Minimum Send Ahead Quantity. After production of this quantity, the start follow-up process may begin. | xs:double | | |
| **MinimumOverlapTime** | Minimum overlap time (for target overlaps). The next operation should start before the end of the preceding one, and the overlap must be at least this specified duration. | xs:long | | |
| **MinimumOverlapTimeUnit** | Minimum overlap time unit | ct:timeUnitType | | |
| **UserFields** | User data fields | ct:UserDataType | | |

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **OperationComponents** | Used components or bill of material (see section 4.2.4) | Sequence of ord:OperationComponentType | | |
| **ProductionResourceTools** | Used production resources and tools (see section 4.2.5) | Sequence of ord:ProductionResourceToolType | | |
| **Material** | Produced material | ct:MaterialType | x | |
| **CustomerEnhancements** | User fields | Sequence of **ord:PartialOrderQuantityDataSection** (*) | | |

(*) Attribute in data ype **ord:PartialOrderQuantityDataSection**:

| | | | | |
|---|---|---|---|---|
| **PartialOrderQuantityDataSection** | User fields | Sequence of **ord:PartialOrderQuantityDataType** (**) | | |

(**) Attribute in data type **ord:PartialOrderQuantityDataType**:

| | | | | |
|---|---|---|---|---|
| **UF1** | User field | xs:string(255) | | |
| **…** | | | | |
| **UTF50** | User field | xs:string(255) | | |

### 4.2.4 Components

The components required for an operation are transmitted as a sequence of individual components, thus forming the component list (also often called a bill of materials).

The **component number** and **position number** are used to identify a component.
The position number is used to book or define the dependent requirements.

The complex data type **ord:OperationComponentType** is used for defining and transmitting a component. This has the following attributes:

**Table 12: Attributes in dataset component**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| Name | Description | Data type (Field length) | Input/ Change | Delete |
| Action | Dataset action | ct:actionType | x | x |
| ComponentNumber | Component number | xs:string(255) | x | x |
| PositionNumber | Position number of the booking or dependent requirement | xs:string(255) | x | x |
| BaseUnitOfMeasure | Basic quantity unit | xs:string(255) | | |
| BatchNumber | Batch number | xs:string(255) | | |
| ConfirmationNumber | Confirmation number of the operation | xs:string(255) | | |
| Description | Description | ct:MultilanguageTextType | | |
| MaterialGroup | Material group | xs:string(255) | | |
| MaterialType | Material type | xs:string(255) | | |
| RequirementDate | Requirement date | xs:dateTime | | |
| RequirementQuantity | Requirement quantity | xs:double | | |
| ReservationNumber | Number of the booking or dependent requirement | xs:string(255) | | |
| StorageBin | Storage bin | xs:string(255) | | |
| StorageLocation | Storage location | xs:string(255) | | |

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **TraceNeeded** | Defines if the batch number must be entered by the user of the SFT for the removal of this component. | xs:boolean | | |
| **UserFields** | User data fields | ct:UserDataType | | |

### 4.2.5 Production resources and tools

The production resources/tools required for an operation are defined as a sequence of individual production resources/tools.

A number, type identifier or optionally a group is used for identifying an individual production resource/tool.

The complex data type **ord:ProductionResourceToolType** is used for defining and transmitting an individual production tool. This has the following attributes:

**Table 13: Attributes in dataset production resources/tools**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **Number** | Number | xs:string(255) | x | x |
| **Type** | Type identifier | xs:string(255) | x | x |
| **Sequence** | Number in sequence | xs:string(255) | | |
| **Quantity** | Quantity | xs:double | | |
| **Unit** | Quantity unit | xs:string(255) | | |
| **Group** | Group | xs:string(255) | | |
| **UserFields** | User data fields | ct:UserDataType | | |
| **Description** | Description | ct:MultilanguageTextType | | |

## 4.3 Shift data

Shift data is always transmitted totalled up. This means the data of all shifts are transmitted together in one dataset. All existing shifts will be deleted except for the current shift. This one will not be changed.

ⓘ Shifts can also be maintained fully in FORCAM FORCE™.

Each attribute that can be transferred has a corresponding entry in the database used in FORCAM FORCE™. Some attributes are mandatory, depending on whether the dataset event is a new entry, a change or a deletion. This can be seen in the following tables in the columns on the right.

The shifts are summarized and transmitted as a sequence of individual shift definitions:

**Table 14: Attributes in dataset shifts**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **Action** | Dataset action | ct:actionType | x | x |
| **ShiftsType** | Shifts | Sequence of **shift:ShiftType** | x | x |

The main components of a shift definition are:

- The **ERP key** for identification
- The **workplace** for which the shift definition is intended
- The **shift type** to specify whether it is a free shift, early shift, late shift, night shift or manual shift
- The **shift day** to define the day of the week
- The **start date** and **end date** to define the time interval (clock times) of the shift

Unlike the other shift types, manual shifts can be created via a special dialog in the Shopfloor Terminal (SFT).
Using this shift type, for example, lets workers on site assign unplanned activities or operations to a shift at short notice. This will prevent the need for shift correction at a later time.

To define a single shift, the complex data type **shift:ShiftType** is used with the following attributes:

**Table 15: Attributes in complex data type shift:ShiftType**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| Name | Description | Data type (Field length) | Input/ Change | Delete |
| ERPKey | ERP key | ct:ERPKeyType | x | x |
| Workplace | Workplace | xs:string(255) | x | x |
| ShiftTypeCode | Shift type<br>1- 10: Free shift<br>11: Early shift<br>12: Late shift<br>13: Night shift<br>14-15: Manual | xs:long | x | x |
| ShiftDate | Shift day | xs:date | x | x |
| StartDateTime | Start date | xs:dateTime | x | x |
| EndDateTime | End date | xs:dateTime | x | x |
| Breaks | Shift breaks | Sequence of **shift:ShiftBreak Type (*)** | | |

The complex data type **shift:ShiftBreakType** can be used to additionally define one or more time intervals for breaks within the shift.

**Table16: Attributes in complex data type shift:ShiftBreakType**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| Name | Description | Data type (Field length) | Input/ Change | Delete |
| StartDateTime | Start date | xs:dateTime | x | x |
| EndDateTime | End date | xs:dateTime | x | x |

## 4.4 Personnel data

Personnel data is always transmitted totalled up. This means that the data of all personnel involved in the production are transmitted together in one dataset. This will also delete personnel who are no longer part of that collective production process.

ⓘ Personnel master data can also be maintained fully in FORCAM FORCE™. Different or abstract individual names or roles (e.g. "Operator01") can also be defined. Overwriting can be prevented by setting the appropriate **Do not delete on resupply** attribute in FORCAM FORCE™ when a resupply is downloaded.

The personnel data is summarized and transmitted as a sequence of all employees in the personnel master:

**Table 17: Attributes in dataset personnel data.**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **Action** | Dataset action | ct:actionType | x | x |
| **PersonsType** | Persons | Sequence of **pers:PersonType** | x | x |

The most important data of an employee are:

- **ERP key**, **personnel number**, **ID number** and **name** to identify the employee
- **Workplace** for which the shift definition is intended
- **Shift type** to specify whether it is a free shift, early shift, late shift, night shift or manual shift
- **The shift day** to define the day of the week

To define an employee, the complex data type **pers:PersonType** is used with the following attributes:

**Table 18: Attributes in complex data type pers:PersonType**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **PersonERPKey** | ERP key | pers:PersonERPKeyType | | |
| **PersonnelNumber** | Personnel number | xs:string(255) | x | x |
| **IdentificationNumber** | ID number | xs:string(255) | x | x |
| **FirstName** | First name | xs:string(255) | x | x |
| **LastName** | Last name | xs:string(255) | x | x |
| **CostCenter** | Cost center | xs:string(255) | | |

**Table 19: Attributes in complex data type pers:PersonERPKeyType**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **Client** | Client | xs:string(32) | x | x |
| **CompanyCode** | Company code | xs:string(32) | x | x |
| **PersonnelArea** | Personnel area | xs:string(255) | x | x |
| **SystemId** | System ID | xs:string(32) | | |

The **System ID** is used for unique identification of the ERP key in the system. This is important in case the attributes **client**, **company code** and **personnel area** are not unique.

## 4.5 Data for overhead costs

FORCAM FORCE™ offers the functionality to accumulate overhead costs incurred during order processing via so-called overhead cost collectors and to report them back to the ERP system via the upload service. At present, however, feedback is only possible for SAP ERP systems or in IDoc format. The definitions of the overhead cost collectors or the overhead cost collection master data can be automatically transmitted from the ERP system to FORCAM FORCE™ as a totalled up supply.

To assign an overhead cost collector to a controlling area, the complex data type **oc:OverheadCostERPKeyType** is used as a separate ERP key with the following attributes:

**Table 20: Attributes in complex data type oc:OverheadCostERPKeyType**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **Client** | Client | xs:string | x | x |
| **CompanyCode** | Company code | xs:string | x | x |
| **ControllingArea** | Controlling area | xs:string | x | x |
| **SystemId** | System ID | xs:string | x | x |

The dataset for defining the overhead cost collectors used is summarized as a sequence of individual overhead cost collector definitions and transmitted from the ERP system to FORCAM FORCE™ :

**Table 21: Attributes in dataset overhead cost**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **Action** | Dataset action | ct:actionType | x | x |
| **OverheadCost** | Overhead costs | Sequence of **oc:OverheadCostsCollectorType** | x | x |

To define an individual overhead cost collector, the complex data type **oc:OverheadCostsCollectorType** is used with the following attributes:

**Table 22: Attribute in complex data type oc:OverheadCostsCollectorType**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **OverheadCostsERPKey** | ERP key | **oc:OverheadCostERPKeyType** | x | x |
| **Plant** | Plant | xs:string | | |
| **OverheadCostsCollectorType** | Cost collector type:<br>– OVERHEAD_COST_CENTER (for overhead cost collector)<br>– INTERNAL_ORDER (for internal orders) | xs:string | x | x |
| **CostUnit** | Cost center (for overhead cost collector) | xs:string | | |
| **OrderNumber** | Order number | xs:string | | |
| **OrderType** | Order type | xs:string | | |
| **Abbreviation** | Abbreviation | ct:MultilanguageTextType | | |
| **Description** | Description | ct:MultilanguageTextType | | |
| **ResponsiblePerson** | Responsible person | xs:string | | |
| **Color** | Color code for visualization (RGB value) | xs:string | | |
| **ValidFrom** | Start date of validity | xs:dateTime | | |
| **ValidTo** | End date of validity | xs:dateTime | | |
| **ExecutionCostUnitChangeable** | Refers to the sender cost center (cost center to be credited). Configuration for input/change. Defines whether the sender cost center may be entered and changed in the SFT. | xs:boolean | | |

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **ReceiverCostUnitChangeable** | Refers to the originating cost center (cost center to be debited) directly for internal activity booking. Defines whether the originating cost center may be entered and changed in the SFT. | xs:boolean | | |
| **ERPAutomaticBooking** | Booking type of the confirmed overhead costs in the ERP system:<br>– AUTO: The assignment or booking of costs in the ERP system is automated.<br>– MANUAL: The costs are first written to a separate table. Assignment or booking of the costs in the ERP system is done manually at a later time. | xs:string | | |
| **ERPReportingEnabled** | Activating the confirmation of overhead costs incurred. | xs:boolean | | |
| **UserStatus** | User status | xs:string | | |

## 4.6 Warehouse data from warehouse management system

ERP systems with an existing warehouse management can additionally transmit data regarding the stock of materials.
The complete stock data is then summarized as a sequence of stock levels of individual materials and transferred:

**Table 23: Attributes in dataset stock data**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **StocksType** | Stock data | Sequence of **sto:StockType** | x | x |

To define a stock level for a specific material, the complex data type **sto:StockType** is used with the following attributes:

**Table 24: Attributes in complex data type sto:StockType**

| XML attribute | | | Mandatory action field | |
|---|---|---|---|---|
| **Name** | **Description** | **Data type (Field length)** | **Input/ Change** | **Delete** |
| **Material** | Material | ct:MaterialType | x | x |
| **Quantity** | Quantity | xs:double | x | x |
| **QuantityUnit** | Quantity unit | xs:string(3) | x | x |
| **StockLocation** | Storage location | xs:string(10) | x | x |

## 4.7 Response to quantity requests

In addition to the datasets already mentioned, there are also datasets with quantity information that serve as a response to queries from FORCAM FORCE™  to the ERP system regarding the following contents:

- Inquiry to material units
- Inquiry to packing units
- Inquiry about remaining quantities in the warehouse management system (after completion of operations)

# 5 Upload

Events or event sequences and associated datasets are transmitted via the upload data service by way of upload messages from FORCAM FORCE™ to the ERP.

In FORCAM FORCE™, trigger events such as a change of operation phases (e.g. setup, production or interrupted) or quantity bookings generate a message in the event message stream and are enriched with the corresponding event data. Events can also be enriched with customer-specific calculation data ("characteristic values") via adjustments in the Rule Engine.

The following events are transmitted to the ERP via the upload data service:

- Change of an operation phase
- Messages about operation duration or work time
    - Generated in FORCAM FORCE™ only at the completion of an operation or at the end of a shift (operation runs across shifts)
    - Per activity type: Machine production time, machine setup time
    - For personnel time: Per person and activity type
- Quantity bookings
    - Asynchronous or individually &|| controlled according to intervals
    - One separate ticket for each individual quantity type
        - Yield
        - Scrap quantity
        - Rework quantity
- Corrections
    - Delete/Insert
    - Restrictions must be specified in detail and must be supported by the ERP
    - Only possible with unique ticket number
- Key figures
    - To be defined according to the ERP interface characteristics
    - E.g. actual time per unit calculation → Rule Engine application
- Ad hoc maintenance via the Shopfloor Terminal
    - Ad hoc maintenance request (request maintenance)
    - Ad hoc maintenance start (start maintenance)
    - Ad hoc maintenance finished (finish maintenance)

The data structures of these events are created as DTOs (Data Transfer Objects) and configured according to the object-oriented inheritance principle. Each event contains a generic default content of the base data structure and other inheritances as appropriate, as well as its own event-specific content.
The DTOs are then converted into transferable XML documents by FORCAM FORCE™.

## 5.1 Upload base data types

The following basic data types are available for the messages to be transmitted:

### 5.1.1 ERP key

The dataset **ERPContext** of data type **ERPContextType** is available for identifying ERP resources such as workplaces, operations or personnel.
This has the following attributes:

**Table 25: Attributes in dataset ERP key**

| ERPContextType | | | |
|---|---|---|---|
| **Attribute** | **Description** | **Data type (Field length)** | **Mandatory field** |
| UUID | Globally unique ID of the ERP resource | xs:string(36) | x |
| Identifier1 | Client | xs:string(255) | x |
| Identifier2 | Company code | xs:string(255) | x |
| Identifier3 | Plant | xs:string(255) | x |
| SystemNumber | System ID | xs:string(255) | |

### 5.1.2 Messages to ERP system

The complex data type **ERPUploadMessage** is commonly used for messages to the ERP system. This has the following attributes:

**Table 26: Attributes in dataset Message to ERP system**

| ERPUploadMessage | | | |
|---|---|---|---|
| **Attribute** | **Description** | **Data type (Field length)** | **Mandatory field** |
| **ConfirmationNumber** | Confirmation number | DefaultStringType | x |
| **MessageIdentifier** | Unique messages ID | xs:string(36) | x |
| **MessageType** | Message type<br>– REGULAR (regular)<br>– CORRECTION (Correction) | xs:string | x |
| **MessageTimestamp** | Timestamp | xs:dateTime | x |
| **ActionTimestamp** | Timestamp.<br>For corrections, the time of the correction. | xs:dateTime | x |
| **CustomMessageData** | Sequence of customer-specific data (linked pairs of key with value) | Sequence of xs:string | |
| **OperationMessage** | Message about operations | OperationMessageType (see section 0) | |
| **OperationSchedule** | Message about operation scheduling | OperationScheduleType (See section 5.3) | |
| **WarehouseMessage** | Message about warehouse management system | WarehouseMessageType (See section 5.4) | |

Depending on the message type, either the **OperationMessage**, **OperationSchedule** or **WarehouseMessage** data structure is used.
The following chapters describe the use of these data structures in more detail.

## 5.2 Messages to operations

The complex data type **OperationMessageType** is available for messages about operation events. This has the following attributes:

**Table 27: Attributes in dataset operation event**

| OperationMessageType | | | | |
|---|---|---|---|---|
| **Attribute** | **Sub attributes** | **Description** | **Data type (Field length)** | **Mandatory field** |
| **WorkplaceIdentification** | UUID | Globally unique ID | xs:string(36) | x |
| | ERPContext | ERP key | ERPContextType | x |
| | ERPWorkplaceNumber | Workplace number in the ERP system | xs:string(255) | x |
| | WorkplaceNumber | Workplace | xs:string(255) | x |
| **OperationIdentification** | UUID | Globally unique ID | xs:string(36) | x |
| | ERPContext | ERP key | ERPContextType | x |
| | OrderNumber | Order number | xs:string(255) | x |
| | OperationNumber | Operation number | xs:string(255) | x |
| | OrderSplit | Order split | xs:string(255) | x |
| | OperationSplit | Operation split | xs:string(255) | x |
| | OrderProductionVersion | Production version | xs:string(255) | |
| | OperationSequence | Sequence of the operation in the order | xs:string(255) | |
| **PersonnelIdentification** | UUID | Globally unique ID | xs:string(36) | x |
| | ERPContext | ERP key | ERPContextType | x |
| | PersonnelNumber | Personnel number | xs:string(255) | x *) |
| **ShiftIdentification** | UUID | Globally unique ID | xs:string(36) | x |
| | ShiftDay | Date of the shift event (in UTC) | xs:date | x |
| | ShiftCode | Shift code | xs:string(255) | x |

*) No mandatory field for phase change in setup, production or interrupted.

More attributes in the data type **OperationMessageType** are available depending on the type of operation message. These are described in the following sections.

## 5.2.1 Change of an operation phase

The following additional attributes are available in the data type **OperationMessageType** for messages about operation events in case of operation phase change:

**Table 28: Attributes in the dataset change of operation phase**

| OperationMessageType | | | | |
|---|---|---|---|---|
| **Attribute** | **Sub attributes** | **Description** | **Data type (Field length)** | **Mandatory field** |
| OperationPhase | | Change of operation phase in:<br>– SETUP<br>– PROCESSING<br>– INTERRUPTED<br>– COMPLETED<br>– CLOSED | xs:string | x |

## 5.2.2 Operation quantity bookings and reasons

Additional attributes are available in the data type **OperationMessageType** for messages on operation notifications involving quantity bookings and reasons:

**Table 29: Attributes in the dataset quantity bookings and reasons**

| OperationMessageType | | | | |
|---|---|---|---|---|
| **Attribute** | **Sub attributes** | **Description** | **Data type (Field length)** | **Mandatory field** |
| **OperationQuantity** | MaterialNumber | Number of the material produced in the operation corresponding to the ERP system. | xs:string(255) | x |
| | RelativeQuantity | Reason carried out as a sequence from:<br>– Quality detail<br>– Quality detail as per ERP (optional) | Sequence of<br>– xs:string(255)<br>– xs:string(255) | x |
| | AbsoluteQuantity | Quantity booking carried out as a sequence from<br>– Quality<br>○ YIELD<br>○ SCRAP (optional)<br>○ REWORK (optional)<br>– Quantity<br>– Quantity unit<br>Each quality type may only be booked once in this message. | Sequence of<br>– xs:string(255)<br>– xs:double<br>– xs:string(255) | x |

### 5.2.3 Operation duration

The following additional attributes are available in the data type **OperationMessageType** for messages on operation notifications involving operation durations:

**Table 30: Attributes in dataset operation duration**

| OperationMessageType | | | | |
|---|---|---|---|---|
| **Attribute** | **Sub attributes** | **Description** | **Data type (Field length)** | **Mandatory field** |
| **OperationDuration** | ActivityType | Activity type to be booked (e.g. machine time) | xs:string(255) | x |
| | ActivityRelativeDuration | Duration to be booked (relative time) | xs:duration | x |
| | ActivityAbsoluteDuration | Duration to be booked (absolute time) | xs:duration | x |

### 5.2.4 Component quantities

Additional attributes are available in the data type **OperationMessageType** for messages for operation notifications involving quantity bookings for components:

**Table 31: Attributes in the dataset for component quantities**

| OperationMessageType | | | | |
|---|---|---|---|---|
| **Attribute** | **Sub attributes** | **Description** | **Data type (Field length)** | **Mandatory field** |
| **OperationComponent** | ComponentNumber | Component number | xs:string(255) | x |
| | RelativeQuantity | Reasons carried out as sequence of<br>– Quality detail<br>– Quality detail as per ERP (optional) | Sequence of<br>– xs:string(255)<br>– xs:string(255) | x |
| | AbsoluteQuantity | Quantity booking carried out as a sequence from<br>– Quality<br>○ YIELD<br>○ SCRAP (optional)<br>○ REWORK (optional)<br>– Quantity<br>– Quantity unit<br>Each quality type may only be booked once in this message. | Sequence of<br>– xs:string(255)<br>– xs:double<br>– xs:string(255) | x |

## 5.2.5 Assembling components

Additional attributes are available in the data type **OperationMessageType** for messages about operation events when components are assembled:

**Table 32: Attributes in dataset assembling components**

| OperationMessageType | | | | |
|---|---|---|---|---|
| **Attribute** | **Sub attributes** | **Description** | **Data type (Field length)** | **Mandatory field** |
| **OperationAsBuilt** | ComponentNumber | Component number | xs:string(255) | x |
| | ComponentPosition | Position number | xs:string(255) | x |
| | SerialNumberOutput | Serial number of the produced material | xs:string(255) | x |
| | ContainerNumberOutput | Output container | xs:string(255) | |
| | AssemblyType | Assembly type:<br>– SERIAL<br>– BATCH (Charge) | xs:string | x |
| | SerializationType | Serialization type:<br>– INPUT (assemble)<br>– DELETE (disassemble)<br>– UPDATE (exchange) | xs:string | x |

## 5.3 Operation scheduling

Additional attributes are available for messages regarding a changed schedule (rescheduling) of an operation:

**Table 33: Attributes in dataset operation scheduling**

| OperationScheduleType | | | | |
|---|---|---|---|---|
| **Attribute** | **Sub attributes** | **Description** | **Data type (Field length)** | **Mandatory field** |
| **OperationIdentification** | UUID | Globally unique ID | xs:string(36) | x |
| | ERPContext | ERP key | ERPContextType | x |
| | OrderNumber | Order number | xs:string(255) | x |
| | OrderSplit | Order split | xs:string(255) | x |
| | OperationSplit | Operation split | xs:string(255) | x |
| | OrderProductionVersion | Production version | xs:string(255) | |
| | OperationSequence | Sequence of the operation in the order | xs:string(255) | |
| **TargetResource** | UUID | Globally unique ID | xs:string(36) | x |
| | ERPContext | ERP key | ERPContextType | x |
| | ERPWorkplaceNumber | Workplace number in the ERP system | xs:string(255) | x |
| | WorkplaceNumber | Workplace | xs:string(255) | |
| | WorkplaceType | Workplace type:<br>– SINGLE (single workplace)<br>– GROUP (group workplace) | xs:string | x |
| **TargetStart** | | Planned start of the operation | xs:dateTime | x |
| **TargetEnd** | | Planned end of the operation | xs:dateTime | x |

## 5.4 Overhead cost reports

FORCAM FORCE™ offers the functionality to accumulate overhead costs incurred during order processing via so-called overhead cost collectors and to report them back to the ERP system via the upload service.
At present, however, feedback is only possible for SAP ERP systems via XSLT in IDoc format. See the **Manual - ERP Interface - Data and Events via SAP IDoc**.

## 5.5 Warehouse data to warehouse management system

The upload data service can be used to report back to the ERP warehouse management system any stock levels that change as a result of ongoing production. Stock levels are used for the Track & Trace application (e.g. for container quantities) (see also **Manual - Track & Trace Use Cases**).
The complex data type **WarehouseMessageType** is available for these types of messages involving changes in the stock levels of certain materials. This has the following attributes:

**Table 34: Attributes in dataset stock data**

| **WarehouseMessageType** | | | | |
|---|---|---|---|---|
| **Attribute** | **Sub attributes** | **Description** | **Data type (Field length)** | **Mandatory field** |
| **OrderIdentification** | UUID | Globally unique ID | xs:string(36) | x |
| | ERPContext | ERP key | ERPContextType | x |
| | OrderNumber | Order number | xs:string(255) | x |
| | OrderSplit | Order split | xs:string(255) | x |
| | OrderProduction Version | Production version | xs:string(255) | |
| **OperationIdentification** | UUID | Globally unique ID | xs:string(36) | x |
| | ERPContext | ERP key | ERPContextType | x |
| | OrderNumber | Order number | xs:string(255) | x |
| | OrderSplit | Order split | xs:string(255) | x |
| | OperationSplit | Operation split | xs:string(255) | x |
| | OrderProduction Version | Production version | xs:string(255) | |
| | OperationSequence | Sequence of the operation in the order | xs:string(255) | |
| **WarehouseMessageType** | | Warehouse type in the warehouse management system to which the message refers:<br>– SERIAL (feedback of the stock level for serialized single part)<br>– PRODUCTION_CONTAINER (Feedback of the stock level of a production container unknown to the warehouse management system)<br>– INVENTORY_CONTAINER (feedback of the stock level to a known inventory container) | xs:string | |
| **WarehouseEntity** | ReferenceValue | Reference value.<br>Tracking number of a container or leading serial number of an individual part. | xs:string(255) | |
| | MaterialNumber | Material number | xs:string(255) | |

| WarehouseMessageType | | | | |
|---|---|---|---|---|
| **Attribute** | **Sub attributes** | **Description** | **Data type (Field length)** | **Mandatory field** |
| | AbsoluteQuantity | Sequence from<br>– Quality<br>○ YIELD<br>○ SCRAP (optional)<br>○ REWORK (optional)<br>– Quantity (stock levels)<br>– Quantity unit<br>Each quality type may only be booked once in this message. | Sequence of<br>– xs:string(255)<br>– xs:double<br>– xs:string(255) | |

Depending on **WarehouseMessageType** either **OrderIdentification** (for PRODUCTION_CONTAINER) or **OperationIdentification** (for SERIAL) is used.

## 5.6 Quantity requests

In addition to the above datasets, there are also datasets related to quantity requests to the ERP system involving the following contents:

- Inquiry to material units
- Inquiry to packing units
- Inquiry about remaining quantities in the warehouse management system (after completion of operations)

## 5.7 Serialization and correction capability

A unique serialization of the transmitted data is necessary to ensure sequence orders and unambiguous assignments of tickets.

Correctability can only be ensured if each individual ticket is provided with a unique ticket number everywhere and the superordinate ERP also allows cancellation processes (delete/insert of tickets). A correction process must be specified for each ERP completely.

# 6 Annex

## 6.1 Abbreviations and terms

| Abbreviation | Explanation |
|---|---|
| API | Application Programming Interface. |
| BOD | Business Object Documents (BOD) are XML messages that enable standardized data exchange between the ERP system and Infor ION. |
| Bridge API | RESTful API to FORCAM FORCE™ that allows data to be both retrieved and written in a REST-compliant form.<br>The FORCAM FORCE™ Bridge API MDC (machine data collection) can collect quantity-based piece counter information from the machine to calculate the actual quantity produced. |
| Infor ION | Infor Intelligent Open Network. Infor ION is a middleware cloud platform that enables standardized data exchange in XML format between ERP systems and FORCAM FORCE ™. |
| MDC | Machine Data Collection |
| REST | Representational State Transfer: Programming paradigm for distributed systems (a cluster of independent computers that present themselves to the user as a single system). |
| RESTful API | API for data exchange based on HTTP requests using GET, PUT, POST and DELETE, which is subject to the requirements or restrictions of the REST architecture. |
| RFC | Remote Function Call. Standardized interface or procedure for communication with SAP systems. |
| SFT | Shopfloor Terminal |
| TDM | Tool Data Management. |
| XLS | Extensible Stylesheet Language |
| XSD | XML Schematic definition. Standard for defining data structures for XML documents based on many supported data types. The definition is done via an XSD file, which is in the form of an XML document. |
| XSLT | XSL Transformation. Part of the XSL. Used to convert XML documents between different systems or standards. FORCAM FORCE ™ uses the XSLT to convert between SAP IDoc documents and XML documents that FORCAM FORCE can process. |

## 6.2 Document history

| Version | Date | Name | Change |
|---|---|---|---|
| **1** | 2019-02-01 | Ali Egilmez | Initial document created |
| **2** | 2021-05-26 | Matthias Koranda | Revised according to Force 5.11.12 |