



ERP Interface - Data and Events via SAP IDoc

Version 5.11

Manual



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



Release date: 26.05.21



Document version: 1



Author: STernes

Content

1	General.....	4
1.1	ERP Download and upload data service	5
1.2	Connection per RFC procedure	6
2	Configuration.....	8
2.1	ERP Download.....	8
2.2	ERP Upload	8
2.2.1	ERP Upload mode	9
3	Download	10
3.1	SAP download per WSDL	10
3.2	Order data	11
3.2.1	Order header	11
3.2.2	Product characteristics of the material	12
3.2.3	Operations	14
3.2.4	Components.....	20
3.2.5	Production resources and tools.....	22
3.3	Shift data	23
3.4	Personnel data	25
3.5	Data for overhead costs	26
3.6	Warehouse data from warehouse management system	29
3.7	Response to quantity requests	30
4	Upload.....	31
4.1	Upload base data types	31
4.1.1	ERP key.....	31
4.1.2	Messages to the ERP system	32
4.2	Messages to Operations	33
4.2.1	Change of an operation phase.....	33
4.2.2	Operation quantity bookings and reasons	34
4.2.3	Operation duration.....	35
4.2.4	Component quantities.....	36
4.2.5	Assembling components.....	37
4.3	Operation scheduling	38

Content

4.4	Overhead cost reports	39
4.5	Warehouse data to warehouse management system	40
4.6	Quantity requests	41
5	Annex	42
5.1	Abbreviations and terms	42
5.2	Document history	42

1 General

This document describes the communication link between FORCAM FORCE™ and an SAP ERP-S. Here, the focus is placed on details of the transmitted data, event and process structures. This document is based on the **Manual - ERP Interface - Data and Events via XML** for the general connection of an ERP system.

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 when connecting ERP systems that exchange datasets in SAP IDoc format. This especially applies to SAP ERP systems and is described in more detail in this manual.

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

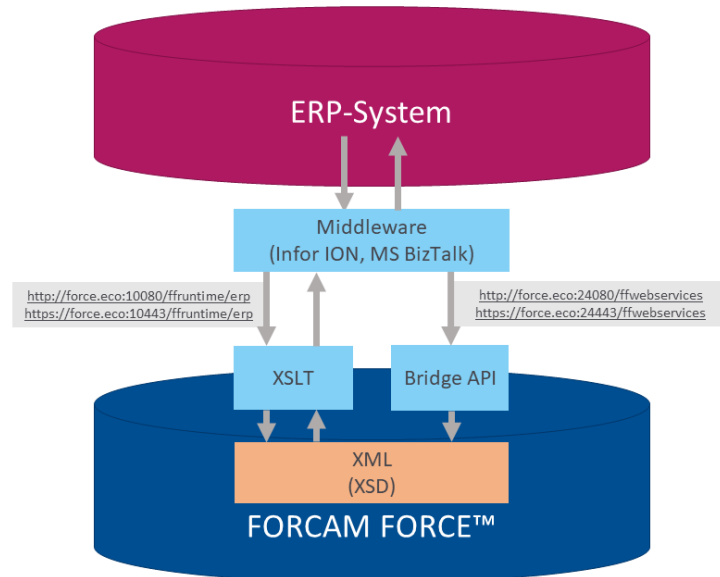


Fig. 1: Alternative connections for the ERP system with FORCAM FORCE™

In addition, middleware such as Infor ION or Microsoft BIZ Talk can be used for the connection to the ERP. This functionality can be used to transform the required datasets already 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 primarily 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 primarily used for the confirming quantities, times and states (status messages) of orders (event message stream). With durations and quantities, the absolute value is transmitted in addition to any relative changes.

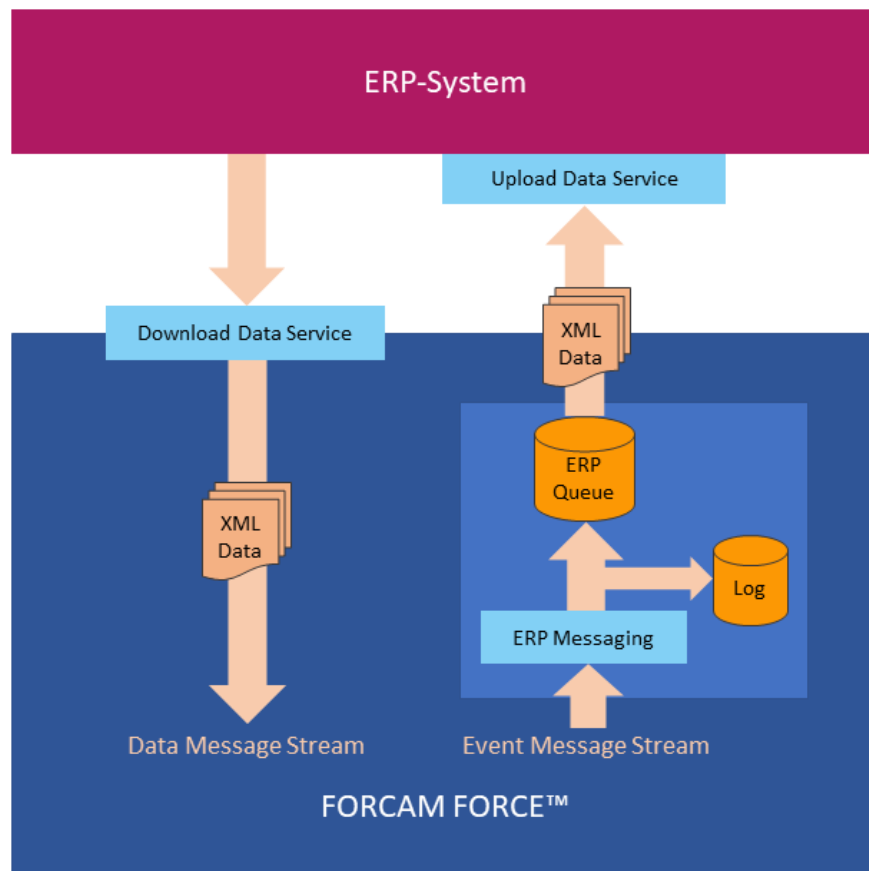


Fig. 2: ERP Download and upload data service

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.

1.2 Connection per RFC procedure

The communication link per XSL Transformation (XSLT) between the SAP ERP system and FORCAM FORCE™ is done via SAP's own RFC interface.

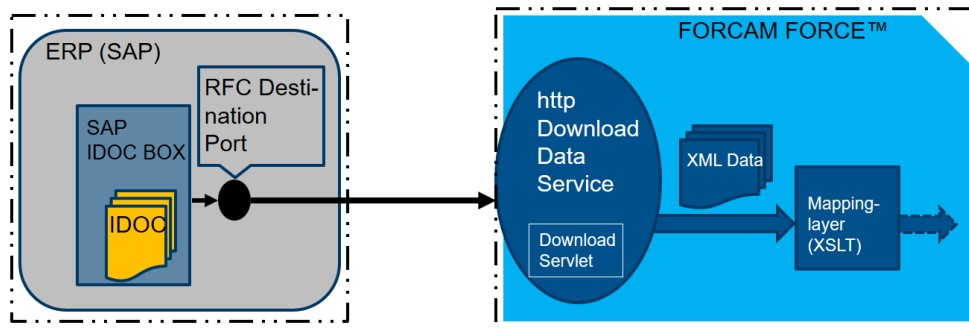


Figure 3: Download connection from SAP -> FORCAM FORCE™ via RFC

An XML-HTTP port with a corresponding RFC destination is created in the SAP ERP system (content type = text/xml). The RFC destination is created in the SAP ERP system and consists of the parameters

- HTTP connection to external server
- Target machine (FORCAM FORCE™ server)
- Path prefix
- Service number = communication port (e.g. 10080)

The IDoc is sent via the port as an XML file at the operating system level. FORCAM FORCE™ then waits for a transmittal and passes the XML through for further processing.

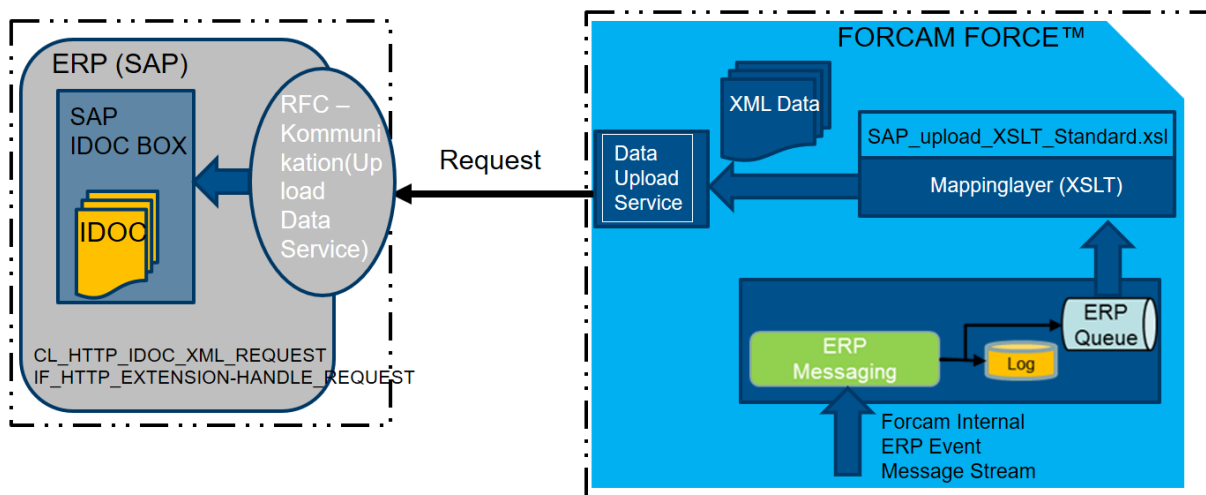


Fig. 4: Upload connection from FORCAM FORCE™ to target SAP ERP

General

The upload data service in SAP applies the IDoc Communication Framework using ALIAS (TRX SICF) with the current parameters

- HTTP port via SMICM,
- Spring – services (8000)
- System user (communication)

The FORCAM FORCE™ data upload service logs on to the SAP gateway (secure application server) with the system user, and the IDoc is transmitted.

Each logical message (IDoc basic type) is linked to a confirmation code via the partner profile (transaction WE20). The confirmation code is created in SAP and linked to an input function module that processes the IDoc documents received for this purpose.

2 Configuration

2.1 ERP Download

If data is to be transmitted via the Bridge API, the configuration is done as described in the **Manual - ERP Interface - Data and Events via XML**.

For the download data service via XSLT procedures, there are corresponding XSL transformation templates preconfigured which can be viewed via the following menu.

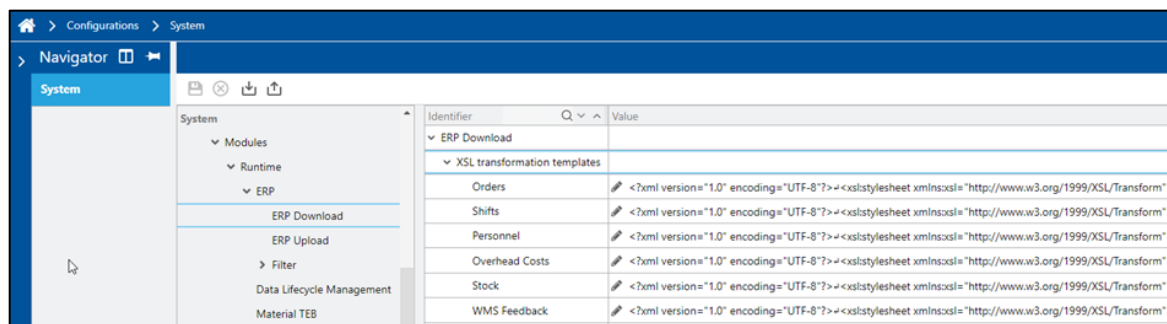


Fig. 5: Configuration of ERP download mode via runtime

2.2 ERP Upload

The configuration of the upload data service is basically done as described in the **Manual - ERP Interface - Data and Events via XML**.

Observe the following configuration for download data service via runtime using the XSLT method.

2.2.1 ERP Upload mode

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).

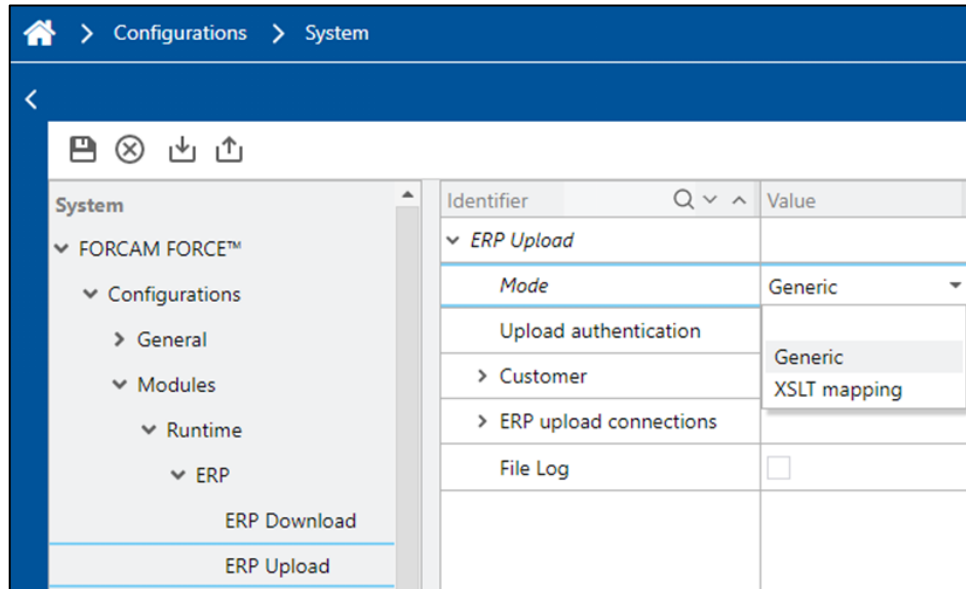


Fig. 6: Configuration ERP upload mode

3 Download

3.1 SAP download per WSDL

For the download data service, which is adapted to SAP ERP systems, a "download servlet" is available using the file type *.wsdl (webservice description language).

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="SAPDownload"
  targetNamespace="http://www.forcam.com/wsdl/SAPDownload"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.forcam.com/wsdl/SAPDownload.wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <!-- SAP IDOC -> XML -->
  <message name="idoc">
    <part name="idocXML" type="xsd:string"/>
  </message>

  <!-- Response from SAPDownloadDispatcherServlet (OK or ERROR) -> HTML -->
  <message name="response">
    <part name="responseHTML" type="xsd:string"/>
  </message>

  <!-- http://localhost:12080/ffruntime/sap -->
  <portType name="SAPDownloadPort">
    <operation name="ffruntime/sap">
      <input message="tns:idoc"/>
      <output message="tns:response"/>
    </operation>
  </portType>

  <service name="SAPDownloadService">
    <port name="downloadPort" binding="tns:SAPDownloadBinding">
      <http:address location="http://localhost:12080/">
    </port>
  </service>

  <binding name="SAPDownloadBinding" type="SAPDownloadPort">
    <http:binding verb="POST"/>
    <operation name="ffruntime/sap">
      <http:operation location="ffruntime/sap"/>
      <input>
        <mime:content type="text/xml"/>
      </input>
      <output>
        <mime:content type="text/html"/>
      </output>
    </operation>
  </binding>
</definitions>
```

3.2 Order data

Order data is read in the IDoc segment /FFMES/SVSART with the supply type (VSART) **production order**.

3.2.1 Order header

The order header data is read in the IDoc segment /FFMES/SAUFTR with the record type (SART) **AUFTR**.

The individual IDoc fields are mapped to corresponding XML attributes as follows:

Table 1: XSLT mapping dataset order header

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
VAKZ	1	Action <ul style="list-style-type: none"> — NOP (if VAKZ = X) — DELETE (if VAKZ = D) — ADD/UPDATE (if otherwise) 	Processing flag/dataset action <ul style="list-style-type: none"> — Do not adopt dataset (only child elements) — Delete dataset — Create/change dataset 	ct:actionType
MANDT	3	Client	ERP key client	ct:ERPKeyType
BUKRS	4	CompanyCode	ERP key company code	
WERKS	4	Plant	ERP key plant	
SYSID	4	SystemId	ERP key system ID	
AUFNR	12	OrderNumber	Order number	xs:string(32)
ASPLT	1	OrderSplit	Order split	xs:string(8)
STATU_ID	80	ERPStatusIds	Status IDs from ERP system (with space as separator)	xs:string(255)
STATU	80	ERPStatusCodes	Corresponding status codes from ERP system (with space as separator)	xs:string(255)
STATU_LA	80	ERPStatusCodesTranslation	Translation for status codes (with space as separator)	xs:string(255)
APRIO	1	OrderPriority	Priority	xs:string(255)
FRDAT	10	ReleaseDateTime	Release date	xs:dateTime

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
SSTRT	10	TargetStartDateTime	Target start date	xs:dateTime
SLIEF	10	TargetEndDateTime	Target end date	xs:dateTime
AFANZ	17	TargetQuantity	Target quantity	xs:double
AMEIN	3	DisplayQuantityUnit	Quantity unit	xs:string(255)
VERID	4	ProductionVersion	Production version	xs:string(255)
VAGRP	3	PlannerGroupInCharge	Responsible planning group/department	xs:string(255)
ATEXT	40	Description	Description	ct:MultilanguageTextType
GSTRS	8	ScheduledStartDate	Scheduled start date	xs:dateTime
GLTRS	8	ScheduledEndDate	Scheduled end date	xs:dateTime
MTEXT	40	Description	Produced material: description	ct:MaterialType
MATNR	18	Number	Produced material: material number	

3.2.2 Product characteristics of the material

These characteristics of the produced material are read in the IDOC segment **/FFMES/SAUFMK** with the record type (SART) **AUFMK**, if it was set this way in the configuration. The configuration is set via the **/FFMES/AUFTR** table with the **ACTIVE_MKL** attribute.

The complex data type **ord:MaterialCharacteristicType** is available for defining and transmitting the dataset **material characteristics**.

The individual IDoc fields are mapped to corresponding XML attributes as follows:

Table 2: XSLT mapping dataset of material characteristics

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
VAKZ — S	1	Action — ADD/UPDATE	Processing flag/dataset action — Create/change dataset	ct:actionType
POSNR	4	orderItem	Order item	xs:string(255)

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
CLASS	40	classType	Label class type	xs:string(255)
ATNAM	30	characteristicName	Characteristic name	xs:string(255)
ATWRT	30	characteristicValue1	Characteristic value 1	xs:string(255)
ATW02	30	characteristicValue2	Characteristic value 2	xs:string(255)
ATW03	30	characteristicValue3	Characteristic value 3	xs:string(255)
ATW04	30	characteristicValue4	Characteristic value 4	xs:string(255)
ATW05	30	characteristicValue5	Characteristic value 5	xs:string(255)
ATW06	30	characteristicValue6	Characteristic value 6	xs:string(255)
ATW07	30	characteristicValue7	Characteristic value 7	xs:string(255)
ATW08	30	characteristicValue8	Characteristic value 8	xs:string(255)
ATW09	30	characteristicValue9	Characteristic value 9	xs:string(255)
ATW10	30	characteristicValue10	Characteristic value 10	xs:string(255)
ATW11	30	characteristicValue11	Characteristic value 11	xs:string(255)
ATW12	30	characteristicValue12	Characteristic value 12	xs:string(255)
ATW13	30	characteristicValue13	Characteristic value 13	xs:string(255)

3.2.3 Operations

This data is read in the IDOC segment **/FFMES/SAFOLG** with the record type (SART) **AFOLG**, if this was set in the configuration. The configuration is set via the **/FFMES/AUFTR** table with the **AKTIV_AVO** flag.

The complex data type **ord:OperationType** is used for defining and transmitting an operation. The individual IDoc fields are mapped to corresponding XML attributes as follows:

Table 3: XSLT mapping dataset operations

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
VAKZ	1	Action <ul style="list-style-type: none"> — NOP (in case VAKZ = X) — DELETE (in case VAKZ = D) — ADD/UPDATE (otherwise) 	Processing flag/dataset action <ul style="list-style-type: none"> — Do not adopt dataset (only child elements) — Delete dataset — Create/change dataset 	ct:actionType
MANDT	3	Client	ERP key client	ct:ERPKeyType
BUKRS	4	CompanyCode	ERP key company code	
WERKS	4	Plant	ERP key plant	
SYSID	4	SystemId	ERP key system ID	
VORNR	12	OperationNumber	Operation number	xs:string(32)
VSPLT	1	OperationSplit	Operation split <ul style="list-style-type: none"> — 0: Without operation split (pull procedure) — 1: Assignment to 1st workplace of the workplace group capacity group — 2: Assignment to 2nd workplace of the workplace group 3: Etc. 	xs:string(8)
RUECK	10	ConfirmationNumber	Confirmation number	xs:string(255)
LMNGA	17	ERPYieldQuantity	Reported yield	xs:double

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
RMNGA	17	ERPReWorkQuantity	Reported rework	xs:double
XMNGA	17	ERPScrapQuantity	Reported scrap quantity	xs:double
AUSSS	17	ERPPlannedScrapQuantity	Planned scrap quantity	xs:double
STATU_ID	80	ERPStatusIds	Status IDs in the ERP system (with spaces as separators)	xs:string(255)
STATU	80	ERPStatusCodes	Status codes in the ERP system (with spaces as separators)	xs:string(255)
STATU_L A	80	ERPStatusCodesTranslation	Translation for status codes	xs:string(255)
LTXA1	80	Description	Description	ct:MultilanguageTextType
AFOTXT	80	OperationText	Long text	ct:MultilanguageTextType
STEUS	4	ControlKey	Control key	xs:string(255)
AUART	4	OrderType	Order type	xs:string(255)
		FunctionType	Function of the operation <ul style="list-style-type: none"> 1: Normal (default) 2: Training 3: Maintenance 	xs:long
		LeadingOperation	Leading operation within operation block (yes/no) <ul style="list-style-type: none"> True (default) False 	xs:boolean
		BlockNumber	Operation block	xs:string(255)
		BlockDescription	Description of operation block	ct:MultilanguageTextType
PLNFL	6	OperationSequence	Order sequence	xs:string(255)
		AutoStartId	ID that comes from the machine to automatically start the process	xs:string
ARBPLI	8	TargetWorkplace	Workplace for the production	xs:string(255)
		TargetWorkGroup	Workplace group ID of the workplace	xs:string
ARBPLG		WorkplaceGroup	Workplace group of the workplace	xs:string(255)

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
SSAVD	10	TargetStartDateTime	Planned start date	xs:dateTime
SSAVZ	5			
SSEDD	10	TargetEndDateTime	Planned end date	xs:dateTime
SSEDZ	5			
		DefaultStrokeFactor	Stroke factor. The quantity produced per stroke or per tact/production cycle 1 (default)	xs:double
BMSCH	17	DefaultPieceTimeFactor	Piece time factor Used for calculating Piece time per quantity unit	xs:double
MEINH	3	DisplayQuantityUnit	Quantity unit	xs:string(255)
MGVRG	17	TargetQuantity	Target quantity	xs:double
VGW01	17	StandardValue1	Target setup time	xs:string(255)
VGE01	3	StandardUnit1	Target unit setup time	ct:timeUnitType
VGW02	17	StandardValue2	Time per unit	xs:string(255)
VGE02	3	StandardUnit2	Time per unit	ct:timeUnitType
VGW03	17	StandardValue3	Default value 3	xs:string(255)
VGE03	3	StandardUnit3	Default value 3 unit	ct:timeUnitType
VGW04	17	StandardValue4	Default value 4	xs:string(255)
VGE04	3	StandardUnit4	Default value 4 unit	ct:timeUnitType
VGW05	17	StandardValue5	Default value 5	xs:string(255)
VGE05	3	StandardUnit5	Default value 5 unit	ct:timeUnitType
VGW06	17	StandardValue6	Default value 6	xs:string(255)
VGE06	3	StandardUnit6	Default value 6 unit	ct:timeUnitType

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
EVORN	4	AlternateOperationNumber	Alternative operation number	xs:string(255)
VERID	4	ProductionVersion	Production version	xs:string(255)
		CounterNumber	Counter number 0 (default)	xs:long
ASTTX	40	UserStatus	User state in the ERP system	xs:string(255)
UELIC	1	OverdeliveryCheck	Check for compliance with the overdelivery tolerance	xs:boolean
UELIT	17	OverdeliveryQuantity	Overdelivery tolerance (limitation in percent for the booked quantity above the target quantity)	xs:double
UNLIC	1	UnderdeliveryCheck	Check for compliance with the underdelivery tolerance	xs:boolean
UNLIT	17	UnderdeliveryQuantity	Underdelivery tolerance (limitation in percent for the booked quantity below the target 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	Predefined consumption quantity of a component per produced material unit, if this component is not included in the bill of material	xs:double
FSAVD	8	EarliestStartDate	Earliest start date	xs:dateTime
SSAVD	8	LatestStartDate	Latest start date	xs:dateTime
FSEDD	8	EarliestEndDate	Earliest end date	xs:dateTime
SSEDD	8	LatestEndDate	Latest end date	xs:dateTime
EPANF	8	ScheduledStartDate	Scheduled start date	xs:dateTime
EPEND	8	ScheduledEndDate	Scheduled end date	xs:dateTime

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
ZWNOR	8	TargetQueueTime	Target queue time, default duration of the queue time in ms.	xs:long
		TargetQueueTimeUnit	Target queue time unit	ct:timeUnitType
BEARZ	16	TargetProcessingTime	Target processing time, default duration of the processing time in ms: 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
ABRUE	16	TargetTeardownTime	Target teardown time, default duration of teardown time (in ms)	xs:long
		TargetTeardownTimeUnit	Target teardown time unit	ct:timeUnitType
LIEGZ	16	TargetWaitTime	Target wait time, target duration of the wait time	xs:long
		TargetWaitTimeUnit	Target wait time unit	ct:timeUnitType
TRANZ	16	TargetMoveTime	Target move time	xs:long
		TargetMoveTimeUnit	Target move time unit	ct:timeUnitType
MINWE	13	MinimumSendAheadQuantity	Minimum Send Ahead Quantity. After production of this quantity, the start followup process may begin.	xs:double
ZMINU	9	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:UserDataTypes

Download

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
MTEXT	40	Description	Produced material: description	ct:MaterialType
MATNR	18	Number	Produced material: material number	

3.2.4 Components

This data is read in the IDoc segment **/FFMES/SAFOKO** with the record type **AFOKO**, if this was set in the configuration. The configuration is set via the **/FFMES/AUFTR** table with the **ACTIVE_COM** flag. By default, there are no dummy modules (RESB-DUMPS = X) or text items (TXTPS = X) created. Likewise, no record is sent for co-products (RESB-KZKUP = X)

The complex data type **ord:OperationComponentType** is used for defining and transmitting a component.

The individual IDoc fields are mapped to corresponding XML attributes as follows:

Table 4: XSLT mapping dataset component

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
VAKZ	1	Action <ul style="list-style-type: none"> — DELETE (in case VAKZ = D) — ADD/UPDATE (otherwise) 	Processing flag/dataset action <ul style="list-style-type: none"> — Delete dataset — Create/change dataset 	ct:actionType
MATNR	18	ComponentNumber	Component number	xs:string(255)
RSPOS	4	PositionNumber	Position number of the booking / dependent requirement	xs:string(255)
MEINS	3	BaseUnitOfMeasure	Basic quantity unit	xs:string(255)
CHARG	10	BatchNumber	Batch number	xs:string(255)
RUECK	10	ConfirmationNumber	Confirmation number of the operation	xs:string(255)
KTX01	40	Description	Description	ct:MultilanguageTextType
MATKL	9	MaterialGroup	Material group	xs:string(255)
MTART	4	MaterialType	Material type	xs:string(255)
BDTER	8	RequirementDate	Requirements date	xs:dateTime
BDMNG	15	RequirementQuantity	Requirement quantity	xs:double
RSNUM	10	ReservationNumber	Number of the booking or dependent requirement	xs:string(255)
LGPLA	10	StorageBin	Storage bin	xs:string(255)
LGORT	4	StorageLocation	Storage location	xs:string(255)

Download

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
		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:UserDataTypes

3.2.5 Production resources and tools

This data is read in the IDOC segment **/FFMES/SAUFFH** with the **record type AUFFH**, if this was set in the configuration. The configuration is set via the **/FFMES/AUFTR** table with the **flag ACTIVE_FHM**.

The complex data type **ord:ProductionResourceToolType** is used for defining and transmitting an individual production tool.

The individual IDoc fields are mapped to corresponding XML attributes as follows:

Table 5: XSLT mapping dataset production resources and tools

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
VAKZ	1	Action <ul style="list-style-type: none"> — DELETE (in case VAKZ = D) — ADD/UPDATE (otherwise) 	Processing flag/dataset action <ul style="list-style-type: none"> — Delete dataset — Create/change dataset 	ct:actionType
MATNR	18	Number	Number	xs:string(255)
FHART	1	Type	Type identifier	xs:string(255)
LFDNR	4	Sequence	Number in sequence	xs:string(255)
MGVGW	13	Quantity	Quantity	xs:double
MGEINH	3	Unit	Quantity unit	xs:string(255)
		Group	Group	xs:string(255)
		UserFields	User data fields	ct:UserDataTypes
FHTXT	40	Description	Description	ct:MultilanguageTextType

3.3 Shift data

The shift data is read in the IDoc segment **/FFMES/TRANSFER_SHIFT_DATA**, if this was set in the configuration.

The complex data type **shift:ShiftType** is used to define and transfer individual shifts.

The individual IDoc fields are mapped to corresponding XML attributes as follows:

Table 6: XSLT mapping dataset shifts

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
MANDT	3	Client	ERP key client	ct:ERPKeyType
BUKRS	4	CompanyCode	ERP key company code	
WERKS	4	Plant	ERP key plant	
SYSID	4	SystemId	ERP key system ID	
ARBPL	8	Workplace	Workplace	xs:string(255)
KAPTPROG	4	ShiftTypeCode	Shift type 1- 10: Free shift 11: Early shift 12: Late shift 13: Night shift 14-15: Manual	xs:long
VALID_FR	10	ShiftDate	Shift day	xs:date
SHIFT_STR	5	StartDateTime	Start date	xs:dateTime
SHIFT_END	5	EndDateTime	End date	xs:dateTime

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

Table 7: XSLT mapping dataset break

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
_FFMES_BREAK/ BREAK_STR	5	StartDateTime	Start date	xs:dateTime

Download

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
_FFMES_BREAK/ BREAK_EMD	5	EndDateTime	End date	xs:dateTime

3.4 Personnel data

The personnel data is read in the IDOC segment **/FFMES/HRDATA**, if this was set in the configuration. The personnel data is summarized and transmitted as a sequence of all employees in the personnel master:

The complex data type **pers:PersonType** is used to define an employee.

The individual IDoc fields are mapped to corresponding XML attributes as follows:

Table 8: XSLT mapping dataset personnel

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
MANDT	3	Client	Client	pers:PersonERPKey yType
BUKRS	4	CompanyCode	Company code	
WERKS	4	PersonnelArea	Personnel area	
SYSID	4	SystemId	System ID	
PERNR	8	PersonnelNumber	Personnel number	xs:string(255)
ZAUSW	8	IdentificationNumber	ID number	xs:string(255)
VORNA	10	FirstName	First name	xs:string(255)
NACHN	20	LastName	Last name	xs:string(255)
KOSTL	10	CostCenter	Cost center	xs:string(255)

3.5 Data for overhead costs

Overhead cost data will be read in the IDOC segment **/FFMES/OCMDT** if it was set that way in the configuration.

The complex data type **oc:OverheadCostsCollectorType** is used for defining and transmitting.

The individual IDoc fields are mapped to corresponding XML attributes as follows:

Table 9: XSLT mapping dataset overhead costs

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
MANDT	3	Client	Client	oc:OverheadCostERPKeyType
BUKRS	4	CompanyCode	Company code	
KOKRS	4	ControllingArea	Controlling area	
SYSID	4	SystemId	System ID	
WERKS	4	Plant	Plant	xs:string
VAKZ	1	OverheadCostsCollectorType — OVERHEAD_COST_CENTER	Cost collector type: OVERHEAD_COST_CENTER (for overhead cost collector)	xs:string
KOSTL	10	CostUnit	Cost center	xs:string
KTEXT	40	Description	Description	ct:MultilanguageTextType
VERAN	20	ResponsiblePerson	Responsible person	xs:string
		Color #FFFFFF (weiß)	Color code for visualization (RGB value)	xs:string
START	10	ValidFrom	Start date of validity	xs:dateTime
ENDE	10	ValidTo	End date of validity	xs:dateTime
		ExecutionCostUnitChangeable — False	Defines whether the sender cost center may be entered and changed in the SFT.	xs:boolean

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
		ReceiverCostUnitChangeable — False	Defines whether the originating cost center may be entered and changed in the SFT.	xs:boolean
MABUK		ERPAutomaticBooking — AUTO (if MABUK=X) — MANUAL (if otherwise)	Booking type of the confirmed overhead costs in the ERP system: — AUTO: The assignment or booking of costs in the ERP system is automated. — 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
ENFED		ERPReportingEnabled — true (if ENFED=X) — False (if otherwise)	Activating the confirmation of overhead costs incurred.	xs:boolean

In the case of an internal order, the individual IDoc fields are mapped to corresponding XML attributes as follows:

Table 10: XSLT mapping dataset overhead costs for internal orders

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
MANDT	3	Client	Client	oc:OverheadCost ERPKeyType
BUKRS	4	CompanyCode	Company code	
KOKRS	4	ControllingArea	Controlling area	
SYSID	4	SystemId	System ID	
WERKS		Plant	Plant	xs:string

IDoc		XML attribute		
Field	Length	Name	Description	Data type (Field length)
VAKZ	1	OverheadCostsCollectorType — INTERNAL_ORDER	Cost collector type: Internal order	xs:string
AUFNR	12	OrderNumber	Order number (order for which the overhead costs are incurred)	xs:string
AUFTY	2	OrderType	Order type	xs:string
KTEXT	40	Description	Description	ct:MultilanguageTextType
VERAN	20	ResponsiblePerson	Responsible person	xs:string
		Color — #FFFFFF (white)	Color code for visualization (RGB value)	xs:string
		ExecutionCostUnitChangeable — False	Defines whether the sender cost center may be entered and changed in the SFT.	xs:boolean
		ReceiverCostUnitChangeable — False	Defines whether the originating cost center may be entered and changed in the SFT.	xs:boolean
ENFED		ERPReportingEnabled — true (if ENFED=X) — False (if otherwise)	Activating the confirmation of overhead costs incurred.	xs:boolean
ASTTX	40	UserStatus	User status	xs:string

3.6 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 11: Attributes in dataset stock data

XML attribute			Mandatory action field	
Name	Description	Data type (Field length)	Input/Change	Delete
StocksType	Stocks 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 12: Attribute in the 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

3.7 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)

4 Upload

4.1 Upload base data types

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

4.1.1 ERP key

The dataset **ERPContext** of data type **ERPContextType** is available for identifying ERP resources such as workplaces, operations or personnel.

The individual XML attributes are mapped to corresponding IDoc fields as follows:

Table 13: XSLT mapping dataset ERP context (ERP key)

XML attribute			IDoc	
Name	Description	Data type (Field length)	Field	Length
Client	Client	ERPContextType	MANDT	3
CompanyCode	Company code		BUKRS	4
Plant	ERP key plant		WERKS	4
SystemId	System ID		SYSID	4

4.1.2 Messages to the ERP system

The complex data type **ERPUploadMessage** is commonly used for messages to the ERP system. The individual XML attributes are mapped to corresponding IDoc fields as follows:

Table 14: XSLT mapping dataset upload message

XML attribute			IDoc	
Attribute	Sub attributes	Description	Field	Length
ConfirmationNumber	Confirmation number	DefaultStringType	RUECK	10
MessageIdentifier	Unique message ID	xs:string(36)	INTNR	
MessageType	Message type — REGULAR (regular) — CORRECTION (Correction)	xs:string		
MessageTimestamp	Timestamp	xs:dateTime		
ActionTimestamp	Timestamp. For corrections, the time of the correction.	xs:dateTime		
CustomMessageData	Sequence of customer-specific data (linked pairs of keys with value)	Sequence of xs:string		

4.2 Messages to Operations

The complex data type **OperationMessageType** is available for messages about operation events. The individual XML attributes are mapped to corresponding IDoc fields as follows:

Table 15: XSLT mapping dataset operation event

XML attribute			IDoc	
Attribute	Sub attributes	Description	Field	Length
WorkplaceIdentification	UUID	Globally unique ID		
	ERPContext	ERP key		
	ERPWorkplaceNumber	Workplace number in the ERP system	ARBPL	8
	WorkplaceNumber	Workplace	ARBPL	8
OperationIdentification	UUID	Globally unique ID		
	ERPContext	ERP key		
	OrderNumber	Order number	AUFNR	12
	OperationNumber	Operation number	VORNR	12
	OrderSplit	Order split	ASPLT	1
	OperationSplit	Operation split	VSPLT	1
	OrderProductionVersion	Production version	VERID	4
	OperationSequence	Sequence of the operation in the order		
PersonnelIdentification	UUID	Globally unique ID		
	ERPContext	ERP key		
	PersonnelNumber	Personnel number	PERNR	8
ShiftIdentification	UUID	Globally unique ID		
	ShiftDay	Date of the shift event (in UTC)	SCHIT	
	ShiftCode	Shift code	SCHIK	

4.2.1 Change of an operation phase

The individual XML attributes are mapped to corresponding IDoc fields as follows:

Table 16: XSLT mapping dataset operation phase change

XML attribute		IDoc	
Attribute	Description	Field	Length
OperationPhase	Change of operation phase in:	AUSTA	
	— SETUP (setup)	— 48 (Setup)	
	— PROCESSING (production)	— 40 (Production)	
	— INTERRUPTED (interrupted)	— 89 (Interrupted)	
	— COMPLETED (completed)	— 90 (Completed)	

4.2.2 Operation quantity bookings and reasons

The complex data type **OperationMessageType** with the specification **OperationQuantity** is available for messages of operation messages regarding quantity bookings and reasons.

The individual XML attributes are mapped to corresponding IDoc fields as follows:

Table 17: XSLT mapping dataset quantity bookings and reasons

XML attribute			IDoc	
Attribute	Sub attributes	Description	Field	Length
OperationQuantity	MaterialNumber	Number of the material produced in the operation corresponding to the ERP system.	MATNR	18
	RelativeQuantity	Reason carried out as a sequence from: <ul style="list-style-type: none"> Quality detail Quality detail as per ERP (optional) 	LMNGA (yield)	17
	AbsoluteQuantity	Quantity booking carried out as a sequence from <ul style="list-style-type: none"> Quality <ul style="list-style-type: none"> YIELD SCRAP (optional) REWORK (optional) Quantity Quantity unit Each quality type may only be booked once in this message.	XMNGA (scrap)	17
			RMNGA (rework)	17
			MEINH (quantity unit)	3
			AGRUN (scrap reason)	
			NGRUN (rework reason)	

4.2.3 Operation duration

The complex data type **OperationMessageType** with the specification **OperationDuration** is available for messages about operation messages regarding operation durations.

The individual XML attributes are mapped to corresponding IDoc fields as follows:

Table 18: XSLT mapping dataset operation duration

XML attribute			IDoc	
Attribute	Sub attributes	Description	Field	Length
OperationDuration	ActivityType	Activity type to be booked (e.g. machine time)	LSTAR	
	ActivityRelativeDuration	Duration to be booked (relative time)	DURATION	
	ActivityAbsoluteDuration	Duration to be booked (absolute time)		

4.2.4 Component quantities

The complex data type **OperationMessageType** with the specification **OperationComponent** is available for messages about operation messages regarding quantity bookings to components. The individual XML attributes are mapped to corresponding IDoc fields as follows:

Table 19: XSLT mapping dataset component quantities

XML attribute			IDoc	
Attribute	Sub attributes	Description	Field	Length
OperationComponent	ComponentNumber	Component number	MATNR	18
	RelativeQuantity	Reasons carried out as sequence of — Quality detail — Quality detail as per ERP (optional)	LMNGA (yield)	17
	AbsoluteQuantity	Quantity booking carried out as a sequence from — Quality ○ YIELD ○ SCRAP (optional) ○ REWORK (optional) — Quantity — Quantity unit Each quality type may only be booked once in this message.	XMNGA (scrap)	17
			RMNGA (rework)	17
			MEINH (quantity unit)	3
			AGRUN (scrap reason)	
			NGRUN (rework reason)	
	BatchNumber	Batch number	CHARG	

4.2.5 Assembling components

The complex data type **OperationMessageType** with the specification **OperationAsBuild** is available for messages regarding operation events during the assembly of components.

The individual XML attributes are mapped to corresponding IDoc fields as follows:

Table 20: XSLT mapping dataset assembling components

XML attribute			IDoc	
Attribute	Sub attributes	Description	Field	Length
OperationAsBuild	ComponentNumber	Component number	MATNR	18
	ComponentPosition	Position number	RSPOS	
	SerialNumberOutput	Serial number of the produced material	SNROM	
	ContainerNumberOutput	Output container	CONTNR	
	AssemblyType	Assembly type: — SERIAL — BATCH (Charge)	SOBNR	
	SerializationType	Serialization type: — INPUT (assemble) — DELETE (disassemble) — UPDATE (exchange)	SERTYP	

4.3 Operation scheduling

The complex data type **OperationScheduleType** is available for messages regarding a changed schedule (rescheduling) of an operation.

The individual XML attributes are mapped to corresponding IDoc fields as follows:

Table 21: XSLT mapping dataset operation scheduling

XML attribute			IDoc	
Attribute	Sub attributes	Description	Field	Length
OperationIdentification	UUID	Globally unique ID		
	ERPContext	ERP key		
	OrderNumber	Order number	AUFNR	12
	OperationNumber	Operation number	VORNR	12
	OrderSplit	Order split	ASPLT	1
	OperationSplit	Operation split	VSPLT	1
	OrderProductionVersion	Production version	VERID	4
	OperationSequence	Sequence of the operation in the order		
TargetResource	UUID	Globally unique ID		
	ERPContext	ERP key		
	ERPWorkplaceNumber	Workplace number in the ERP system	ARBPL	8
	WorkplaceNumber	Workplace	MES_ WORKP LACE	
	WorkplaceType	Workplace type: — SINGLE (single workplace) — GROUP (group workplace)		
TargetStart		Planned start of the operation	SOLLS	
TargetEnd		Planned end of the operation	SOLLE	

4.4 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.

The individual XML attributes are mapped to corresponding IDoc fields as follows:

Table 22: XSLT mapping dataset overhead cost reports

XML attribute			IDoc	
Attribute	Sub attributes	Description	Field	Length
Client		ERP key company code	MANDT	3
CompanyCode		Company code	BUKRS	4
ControllingArea		Controlling area	WERKS	4
ControllingArea		Controlling area	KOKRS	4
SystemId		System ID	SYSID	4
CostUnit		Cost center	KOSTL	10
ExecutionCostUnit		Defines whether the sender cost center may be entered and changed in the workflow control.	EXCUU	
ReceiverCostUnit		Defines whether the originating cost center may be entered and changed in the workflow control.	DEBCU	
ErpBookingType		Booking type of the confirmed overhead costs in the ERP system.	MABUK	

4.5 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.

The complex data type **WarehouseMessageType** is available for these types of messages involving changes in the stock levels of certain materials.

The individual XML attributes are mapped to corresponding IDoc fields as follows:

Table 23: XSLT mapping dataset stocks data

XML attribute			IDoc	
Attribute	Sub attributes	Description	Field	Length
OrderIdentification	UUID	Globally unique ID		
	ERPContext	ERP key		
	OrderNumber	Order number	AUFNR	12
	OrderSplit	Order split	ASPLT	1
	OrderProductionVersion	Production version		
OperationIdentification	UUID	Globally unique ID		
	ERPContext	ERP key		
	OrderNumber	Order number	AUFNR	12
	OperationNumber	Operation number	VORNR	12
	OrderSplit	Order split	ASPLT	1
	OperationSplit	Operation split	VSPLT	1
	OrderProductionVersion	Production version	VERID	4
	OperationSequence	Sequence of the operation in the order		
WarehouseMessageType		Warehouse type in the warehouse management system to which the message refers: <ul style="list-style-type: none"> — SERIAL (feedback of the stock level for serialized single part) — PRODUCTION_CONTAINER (feedback of the stock level of a production container unknown to the warehouse management system) — INVENTORY_CONTAINER (feedback of the stock level to a known storage container) 		
WarehouseEntity	ReferenceValue	Reference value. Tracking number of a container or leading serial number of an individual part.		
	MaterialNumber	Material number	MATNR	18
	AbsoluteQuantity	Sequence from <ul style="list-style-type: none"> — Quality <ul style="list-style-type: none"> ○ YIELD ○ SCRAP (optional) 		

XML attribute			IDoc	
Attribute	Sub attributes	Description	Field	Length
		<ul style="list-style-type: none"> ○ REWORK (optional) — Quantity (stock levels) — Quantity unit <p>Each quality type may only be booked once in this message.</p>		

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

4.6 Quantity requests

In addition to the above datasets, there are also datasets related to quantity requests 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 Annex

5.1 Abbreviations and terms

Abbreviations	Explanation
API	Application Programming Interface. 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. 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™.
MDE	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.

5.2 Document history

Version	Date	Name	Change
1	2019-02-01	Ali Egilmez	Initial document created
2	2021-05-19	Matthias Koranda	Revised according to Force 5.11.12