



Version 5.10

ERP Interface - Data and Events via XML Manual

Document:	Manual - ERP Interface - Data and Events via XML.docx
Created:	2017-09-14
Last Change:	2019-10-31
Author:	Ali Egilmez



Table of Contents

1 General.....	3
2 Download Interface	4
2.1 General.....	4
2.1.1 Implemented RFC Connection SAP (ERP) to FORCAM FORCE™	4
2.2 Download Data Structures.....	5
2.2.1 Transferring time fields.....	5
2.2.2 SAP – Download WSDL (web service description language)	6
2.2.3 Order Data	7
2.2.3.1 Order Header Data	7
2.2.3.2 Object Characteristics of the Material	14
2.2.3.3 Operation Data	18
2.2.3.4 Long Texts for Operations	28
2.2.3.5 OP Components.....	30
2.2.3.6 Production Resources and Tools	34
2.2.4 Machine Capacities / Shift Plans.....	38
2.2.5 Personnel Data (Mini Master Data Personnel).....	42
3 Upload Interface.....	45
3.1 General.....	45
3.1.1 Use of Implemented RFC Connection SAP (ERP)	46
3.2 FORCAM FORCE™ Feedback Set.....	46
3.3 Feedback Events.....	50
3.3.1 Mapping Layer XSLT (XSL Scheme Product Standard).....	51
3.4 Filled Tags in SAP IDOC Depending on the Set type	52
3.5 Example of a Quantity Report.....	58
3.6 Serialization & Correction Capability	59
3.6.1 Workplace – Serialization	59
3.6.2 OP-Serialization	60
4 Appendix	61
4.1 Upload XSLT Mapper	61
4.2 Download XSLT Mapper	77
4.2.1 Orders	77
4.2.2 Shifts	105
4.2.3 Personnel.....	109

1 General

The data / event and process structure, tailored to SAP as ERP system, currently implemented and provided by FORCAM for an upload and download is described in this document.

The FORCAM FORCE™ system subordinate to the ERP system provides a download service for the communication / data exchange. The ERP provides a corresponding upload service for the information feedback in form of a feedback process. E.g. characteristic values of events to be calculated can be added with a rule engine within the FORCAM FORCE™ system (as an example: the current job-time per piece of an order). This general context is pictured in Fig. 1.

The currently implemented download and upload data constructs are designed and implemented for the ERP system SAP. It can be adapted and modified with specifications to work with other ERP systems.

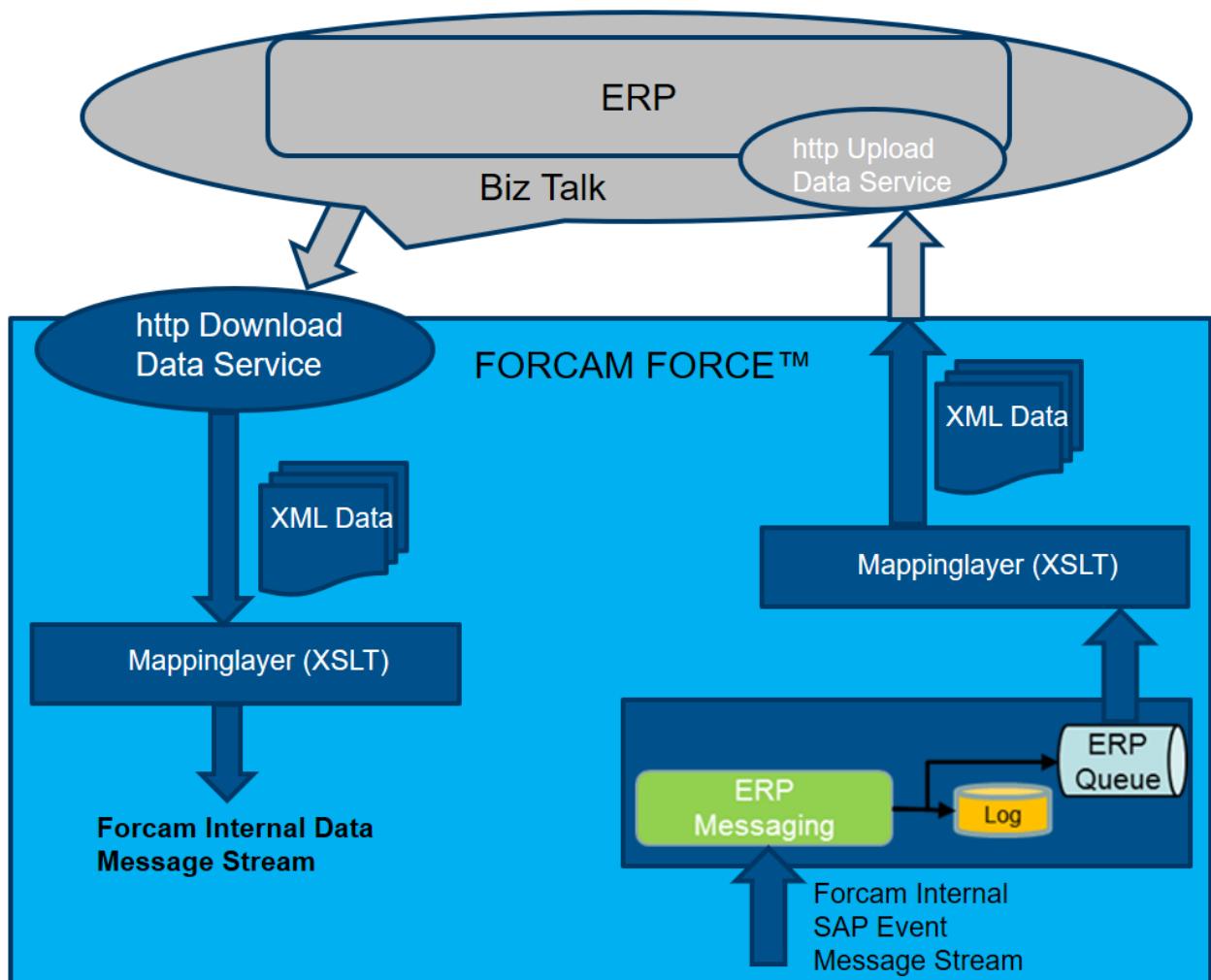


Fig. 1: Upload and download structure between ERP and FORCAM FORCE™

2 Download Interface

2.1 General

All necessary and defined data needed for the download are converted into a defined data structure in the upstream ERP. The data download to the downstream FORCAM FORCE™ system is triggered by events in the ERP system by using a download service that is provided in FORCAM FORCE™.

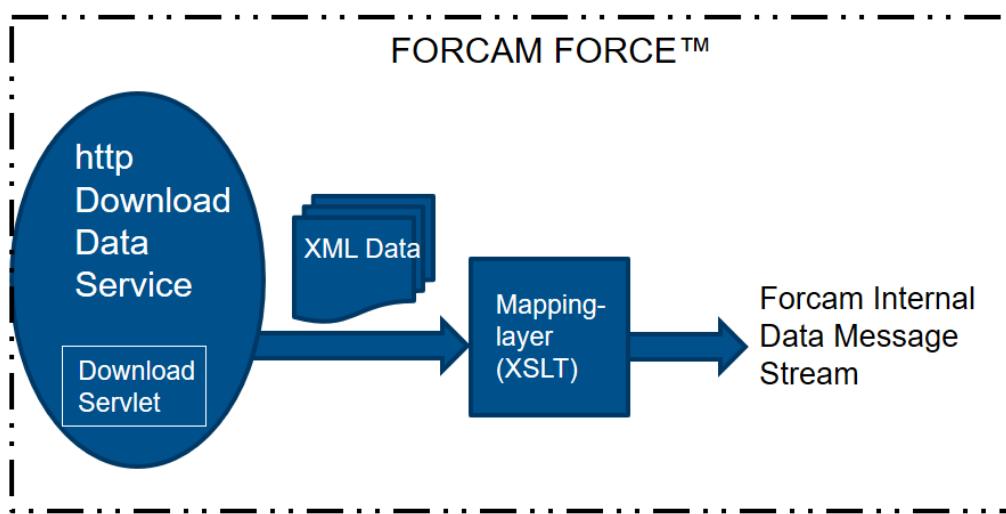


Fig. 2: Common download structure for ERP → FORCAM FORCE™

2.1.1 Implemented RFC Connection SAP (ERP) to FORCAM FORCE™

The communication connection between SAP (ERP) and FORCAM FORCE™ is made in the current implementation and form always by using an explicit RFC communication (destination). This structure is pictured in Fig. 3.

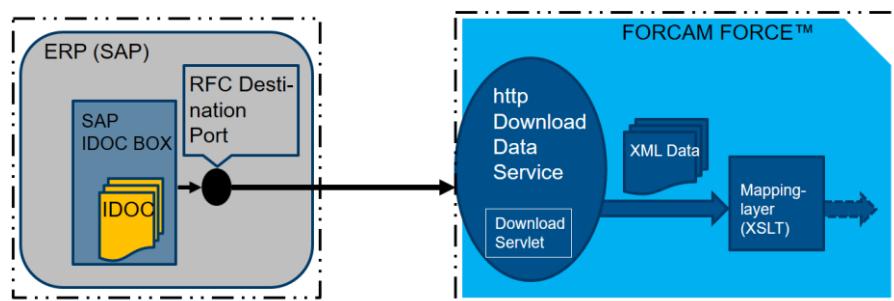


Fig. 3: Download connection SAP -> FORCAM FORCE™ by using RFC

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

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

The IDOC is sent via the port as XML file on the operating system level. The FORCAM FORCE™ service waits for a transmission and passes-through the XML for further processing.

2.2 Download Data Structures

Defined XML data structures are transferred to the download process. The data are interpreted by using a mapping layer via XSL and are then transformed in a FORCAM-intern format.

The following abstract data structures are currently used in the download process:

- Order data
 - Order
 - Product features are not included in the standard
 - OP
 - Production resources and tools: Only necessary, if needed for display or if TDM tool data management is used or if FORCAM NC-data management is used.
 - Components: Only necessary, if the components to be used shall be displayed or shall be reported back to the ERP.
 - Personnel data can also be managed in FORCAM FORCE™
 - Shift data can also be managed in FORCAM FORCE™

i Only the order data and the related operations are mandatory. All other data are needed only for display purposes or for functional extensions like TDM tool data management, NC program management or component verification (trace).

2.2.1 Transferring time fields

When defining a time field, you always have to define a separate field for the unit.

Based on the transferred unit, the transferred value of the time field is converted to a FORCAM FORCE™ internal uniform time base value. Without a valid unit, it cannot be guaranteed that the value is correctly calculated and interpreted.

The following units for time values are supported for the ERP download:

Table 1: Supported units for time values

Type of Time	Description
H	Hour
STD	Hour
M_90	Minute / 90
MIN	Minute
MS	Millisecond
S	Second
TMU	Technical Minute

2.2.2 SAP – Download WSDL (web service description language)

A “Download Servlet” is available for the SAP download, by using the file type *.wsdl (web service 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/frruntime/sap -->
  <portType name="SAPDownloadPort">
    <operation name="frruntime/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="frruntime/sap">
      <http:operation location="frruntime/sap"/>
      <input>
        <mime:content type="text/xml"/>
      </input>
      <output>
        <mime:content type="text/html"/>
      </output>
    </operation>
  </binding>

</definitions>
```

2.2.3 Order Data

2.2.3.1 Order Header Data

These data are transferred to the subsystem in the IDOC segment /FFMES/SAUFTR with the set type **AUFTR**.

The Data structure is listed in Table 2.

Customer individual

- ⓘ All **elements marked red** are special fields that are generated in the XML data structure as empty elements and are not automatically content of the FORCAM adapter for SAP product standard. If these fields shall be used, specification and customizing efforts are made, which source and in what form these data shall be retrieved from. This customer specific development can be implemented by applying user exits.

Table 2: Data structure order header data (source SAP (ERP) and target FORCAM FORCE™)

SAP (ERP)			Label	FORCAM FORCE™					
				Mandatory fields			Persist		
SAP field	Type (Length)	IDOC XML SAP		Down-load	Dele-tion record	Internal XML transfor-mation	Table	Column	Type
	C (5)	SART	Record type (Here: AUFTR)			-	-	-	-
	C (1)	VAKZ	processing flag D or S (D= dele-tion rec ord)			-	-	-	-
AUFK-MANDT	C (3)	MANDT	Client			client	FR_MD_ERP_KEY	CLIENT	VAR-CHAR2(255 CHAR)
AUFK-BUKRS	C (4)	BUKRS	Company code			companyCode	FR_MD_ERP_KEY	COMPANY_CODE	VAR-CHAR2(255 CHAR)
AUFK-WERKS	C (4)	WERKS	Plant			plant	FR_MD_ERP_KEY	PLANT	VAR-CHAR2(255 CHAR)
SY-SYSID	C (8)	SYSID	SAP Sys-tem ID			systemId	FR_MD_ERP_KEY	SYSTEM_ID	VAR-CHAR2(255 CHAR)

AUFK-AUFNR	C (12)	AUFNR	Order number			orderNumber	FR_MD_ORDER	ORDER_NUMBER	VAR-CHAR2(255 CHAR)
	C (1)	ASPLT	Order split (Default: 0)			orderSplit	FR_MD_ORDER	ORDER_SPLIT	VAR-CHAR2(255 CHAR)
AFPO-POSNR (always 0001)	N (4)	POSNR	Order position (always 0001)			position-Number	FR_MD_ORDER	POSITION_NUMBER	VAR-CHAR2(255 CHAR)
AFPO-KNTTP	C (1)	KNTTP	Accounting type			accounting-Type	FR_MD_ORDER	ACCOUNT_TYPE	VAR-CHAR2(255 CHAR)
AFPO-KDAUF	C (25)	KOOBJ	Adapter: accounting object			accountingObject	FR_MD_ORDER	ACCOUNT_OBJECT	VAR-CHAR2(255 CHAR)
AFPO-PROJN	C (40)	ATEXT	Description			description	FF_MD_LITERAL_TEXT	TEXT	VAR-CHAR2(1024 CHAR)
or empty	C (18)	MATNR	Material number			material	FR_MD_MATERIAL	MATERIAL_NUMBER	VAR-CHAR2(255 CHAR)
AUFK-KTEXT	C (40)	MTEXT	Material description			material/description	FF_MD_LITERAL_TEXT	TEXT	VAR-CHAR2(1024 CHAR)
empty	C (20)	KUNNR	Customer number			customer-Number	FR_MD_MATERIAL	CUSTOMER_NUMBER	VAR-CHAR2(255 CHAR)
AFKO-GSTRP	DD.MM.YYYY (10)	SSTRT	Target start date/order start date			targetStartDate	FR_MD_ORDER	TARGET_START	TIMESTAMP(6)
AFPO-STRMP	DD.MM.YYYY (10)	PSTRT	Start date of the			planStartDate	FR_MD_ORDER	PLAN_START	TIMESTAMP(6)

			planned order					
AFKO-GLTRP	DD.MM.YYYY (10)	SLIEF	Target delivery date/order finish date		targetEndDate	FR_MD_ORDER	TARGET_END	TIMESTAMP(6)
AFPO-LTRMP	DD.MM.YYYY (10)	PLIEF	Delivery date from planned order		planEndDate	FR_MD_ORDER	PLAN_END	TIMESTAMP(6)
AFKO-GAMNG	C (17)	AFANZ	Target quantity		targetQuantity	FR_MD_ORDER	TARGET_QUANTITY	VAR-CHAR2(255 CHAR)
AFKO-GMEIN	C (3)	AMEIN	Quantity unit for display (Default: stk)		displayQuantityUnit	FR_MD_ORDER	DISPLAY_QUANTITY_UNIT	VAR-CHAR2(255 CHAR)
TJ02T-TXT04 in ENGLISH	C (80)	STATU	Order status text, e.g. FREI SSAP... (SAP system status)		ppsStatus	FR_MD_ORDER	ERP_STATUS	VAR-CHAR2(255 CHAR)
TJ02T-ISTAT (internal status codes)	C (80)	STATU_ID	Status_ID (system status) e.g. I0002 (order header)		erpStatusIds	FR_MD_ORDER	ERP_STATUS_ID	VAR-CHAR2(255 CHAR)
TJ02T-TXT04 in the language	C (80)	STATU_LA	System status text		erpStatusLa	FR_MD_ORDER	ERP_STATUS_LA	VAR-CHAR2(255 CHAR)

from the table /FFMES/AUFTR			in the order language (order header)					
Empty or X (will be set to X if order status SPER (system status) is set	C (1)	SPRKZ	Lock indicator		lockIndicator	FR_MD_ORDER	LOCK_INDICATOR	VAR-CHAR2(255 CHAR)
AFKO-APRIO	C (1)	APRIO	Order priority		orderPriority	FR_MD_ORDER	ORDER_PRIORITY	VAR-CHAR2(255 CHAR)
AFKO-FEVOR	C (3)	FEVOR	Production scheduler		production-Scheduler	FR_MD_ORDER	PRODUCTION_SCHEDULER	VAR-CHAR2(255 CHAR)
AFKO-PLGRP	C (3)	VAGRP	Planner group/department in charge		planner-GroupInCharge	FR_MD_ORDER	PLANNER_GROUP_IN_CHARGE	VAR-CHAR2(255 CHAR)
AFKO-FTRMI	DD.MM.YYYY (10)	FRDAT	Release date		releaseDate	FR_MD_ORDER	RELEASE_DATE	TIMESTAMP(6)
AFKO-REDKZ	C (1)	REDKZ	Reduction indicator		reductionIndicator	FR_MD_ORDER	REDUCTION_INDICATOR	VAR-CHAR2(255 CHAR)
empty	C (12)	PRFLS	Test batch number		testBatch-Number	FR_MD_ORDER	TEST_BATCH_NUMBER	VAR-CHAR2(255 CHAR)
empty	C (20)	SERNR	Serial number		serialNumber	FR_MD_ORDER	SERIAL_NUMBER	VAR-CHAR2(255 CHAR)
empty	C (80)	ZCHNR	Drawing number		drawing-Number	FR_MD_ORDER	DRAWING_NUMBER	VAR-CHAR2(255 CHAR)

empty	C (3)	PROJN	Project			project	FR_MD_ORDER	PROJECT	VAR-CHAR2(255 CHAR)
empty	C (6)	MODEL	Model			model	FR_MD_ORDER	MODEL	VAR-CHAR2(255 CHAR)
empty	C (4)	GERAE	Device			device	FR_MD_ORDER	DEVICE	VAR-CHAR2(255 CHAR)
AFPO-LGORT	C (4)	LGORT	Storage location			storageLoca-tion	FR_MD_ORDER	STORAGE_LOCATION	VAR-CHAR2(255 CHAR)
empty (only at REM (Repetitive Manufacturing))	C (4)	VERID	Production version (used at REM (Repetitive Manufacturing))			production-Version	FR_MD_ORDER	PRODUCTION_VERSION	VAR-CHAR2(255 CHAR)
TJ30T-ASTTX	C (40)	ASTTX	User status text in English (order header)			userStatus	FR_MD_ORDER	USER_STATUS	VAR-CHAR2(255 CHAR)
TJ30T-ESTAT (external status codes)	C (40)	ASTTX_ID	Status ID (user status) e.g. E0005 (order header)		-	-	-	-	
TJ30T-TXT04 in the language from the table /FFMES/AUFTR	C (40)	ASTTX_LA	User status text in the order language		-	-	-	-	



			(order header)						
AUFK-AUART	C (4)	AUART	Order type						

2.2.3.2 Object Characteristics of the Material

These data are transferred in the IDOC segment /FFMES/SAUFMK to the subsystem with the set type **AUFMK**, if it was preset in the configuration. The configuration is set-up via the table /FFMES/AUFTR with the property **AKTIV_MKL**.

The product standard provides a general data definition for these data. But the specification of object characteristics is defined customer specific and are not filled in the product standard. When using them, specification and customizing efforts are made, which source and in what form these data shall be retrieved from. The customer specific characteristic is implemented by implementing an extension method by applying user exits.

This extension method as structure is contained in the subprogram for filling the segment /FFMES/SAUFMK and is delivered with the product standard. The method is called **FILL_CLASSIFICATION** and the used parameters are listed in Table 3.

Table 3: Parameters used with method FILL_CLASSIFICATION in the FORCAM adapter for SAP

Parameter	Typification	Description
VALUE(IM_AUFK)	TYPE AUFK OPTIONAL	Order master data
VALUE(IM_AFKO)	TYPE AFKO OPTIONAL	Order header data PPS orders
VALUE(IM_AFPO)	TYPE AFPO OPTIONAL	Order position
VALUE(STRUCT_AUFTR)	TYPE /FFMES/AUFTR OPTIONAL	FORCAM adapter for SAP: parameter: Order provision
VALUE(IM_MARA)	TYPE MARA OPTIONAL	Common material data
VALUE(IM_MARC)	TYPE MARC OPTIONAL	Plant data to the material
VALUE(RETURN)	TYPE BAPIRET2 OPTIONAL	Return parameters
VALUE(STRUCT_SAUFMK)	TYPE /FFMES/SAUFMK OPTIONAL	Object characteristics of the material
VALUE(STRUCT_ENHDATA)	TYPE /FFMES/ENHDATA OPTIONAL	Extension with customer's fields

The data structure is listed in Table 4.

Customer Specific

- ⓘ All **elements marked red** are special fields that are generated in the XML data structure as empty elements and are not automatically content of the FORCAM adapter for SAP product standard. If these fields shall be used, specification and customizing efforts are made, which source and in what form these data shall be retrieved from. This customer specific development can be implemented by applying user exits.

Table 4: Data structure object characteristics material (source SAP (ERP) and target FORCAM FORCE™)

SAP (ERP)			Label	FORCAM FORCE™					
				Mandatory fields			Persist		
SAP field	Type (Length)	IDOC XML SAP		Download	Deletion record	Internal XML transformation	Table	Column	Type
AUFMK	C (5)	SART	Record type (Here: UFMK)			-	-	-	-
S	C (1)	VAKZ	Processing flag (Always S)			-	-	-	-
AUFK-MANDT	C (3)	MANDT	Client			client	FR_MD_ERP_KEY	CLIENT	VAR-CHAR2(255 CHAR)
AUFK-BUKRS	C (4)	BUKRS	Company Code			companyCode	FR_MD_ERP_KEY	COMPANY_CODE	VAR-CHAR2(255 CHAR)
AUFK-WERKS	C (4)	WERKS	Plant			plant	FR_MD_ERP_KEY	PLANT	VAR-CHAR2(255 CHAR)
SY-SYSID	C (8)	SYSID	SAP System ID			systemId	FR_MD_ERP_KEY	SYSTEM_ID	VAR-CHAR2(255 CHAR)
AUFK-AUFNR	C (12)	AUFNR	Order Number			orderNumber	FR_MD_ORDER	ORDER_NUMBER	VAR-CHAR2(255 CHAR)
Always 0	C (1)	ASPLT	Order split (Default: 0)			orderSplit	FR_MD_ORDER	ORDER_SPLIT	VAR-CHAR2(255 CHAR)

AFPO-POSNR	N (4)	POSNR	Order item			orderItem	FR_MD_MATERIAL_CHARACTERISTIC	ORDER_ITEM	VAR-CHAR2(255 CHAR)
Empty	C (40)	CLASS	Class type			classType	FR_MD_MATERIAL_CHARACTERISTIC	CLASS_TYPE	VAR-CHAR2(255 CHAR)
Always ,CLASS'	C (30)	ATNAM	Characteristic name			characteristic-Name	FR_MD_MATERIAL_CHARACTERISTIC	CHARACTERISTIC_NAME	VAR-CHAR2(255 CHAR)
Empty	C (30)	ATWRT	Characteristic value1			characteris-ticValue1	FR_MD_MATERIAL_CHARACTERISTIC	CV01	VAR-CHAR2(255 CHAR)
Empty	C (30)	ATW02	Characteristic value2			characteris-ticValue2	FR_MD_MATERIAL_CHARACTERISTIC	CV02	VAR-CHAR2(255 CHAR)
Empty	C (30)	ATW03	Characteristic value3			characteris-ticValue3	FR_MD_MATERIAL_CHARACTERISTIC	CV03	VAR-CHAR2(255 CHAR)
Empty	C (30)	ATW04	Characteristic value4			characteris-ticValue4	FR_MD_MATERIAL_CHARACTERISTIC	CV04	VAR-CHAR2(255 CHAR)
Empty	C (30)	ATW05	Characteristic value5			characteris-ticValue5	FR_MD_MATERIAL_CHARACTERISTIC	CV05	VAR-CHAR2(255 CHAR)
Empty	C (30)	ATW06	Characteristic value6			characteris-ticValue6	FR_MD_MATERIAL_CHARACTERISTIC	CV06	VAR-CHAR2(255 CHAR)
Empty	C (30)	ATW07	Characteristic value7			characteris-ticValue7	FR_MD_MATERIAL_CHARACTERISTIC	CV07	VAR-CHAR2(255 CHAR)
Empty	C (30)	ATW08	Characteristic value8			characteris-ticValue8	FR_MD_MATERIAL_CHARACTERISTIC	CV08	VAR-CHAR2(255 CHAR)

Empty	C (30)	ATW09	Characteristic value9			characteristicValue9	FR_MD_MATERIAL_CHARACTERISTIC	CV09	VAR-CHAR2(255 CHAR)
Empty	C (30)	ATW10	Characteristic value10			characteristicValue10	FR_MD_MATERIAL_CHARACTERISTIC	CV10	VAR-CHAR2(255 CHAR)
Empty	C (30)	ATW11	Characteristic value11			characteristicValue11	FR_MD_MATERIAL_CHARACTERISTIC	CV11	VAR-CHAR2(255 CHAR)
Empty	C (30)	ATW12	Characteristic value12			characteristicValue12	FR_MD_MATERIAL_CHARACTERISTIC	CV12	VAR-CHAR2(255 CHAR)
Empty	C (30)	ATW13	Characteristic value13			characteristicValue13	FR_MD_MATERIAL_CHARACTERISTIC	CV13	VAR-CHAR2(255 CHAR)

2.2.3.3 Operation Data

These data are transferred to the subsystem in the IDOC segment /FFMES/SAFOLG with the set type **AFOLG**, if this was preset in the configuration. The configuration is set-up via the table /FFMES/AUFTR with the flag AKTIV_AVO.

The data structure is listed in Table 5.

Customer Specific

- ⓘ All **elements marked red** are special fields that are generated in the XML data structure as empty elements and are not automatically content of the FORCAM adapter for SAP product standard. If these fields shall be used, specification and customizing efforts are made, which source and in what form these data shall be retrieved from. This customer specific development can be implemented by applying user exits.
- ⓘ Specifically for the SAP fields AFVV-VGW01, AFVV-VGW02, AFVV-VGW03, there is the option available, to manage in table STVAL per workplace, which standard value corresponds to which type, to comply with the static chronological order standard.

Table 5: Data structure operation data (source SAP (ERP) and target FORCAM FORCE™)

SAP (ERP)			Label	FORCAM FORCE™					
				Mandatory fields			Persist		
SAP field	Type (Length)	IDOC XML SAP		Down-load	Deletion record	Internal XML transformation	Table	Column	Type
AUFMK	C (5)	SART	Record type (Here: AUFMK)			-	-	-	-
S	C (1)	VAKZ	Processing flag (Always S)			-	-	-	-
AUFK-MANDT	C (3)	MANDT	Client			client	FR_MD_ERP_KEY	CLIENT	VAR-CHAR2(255 CHAR)
AUFK-BUKRS	C (4)	BUKRS	Company code			companyCode	FR_MD_ERP_KEY	COMPANY_CODE	VAR-CHAR2(255 CHAR)
AUFK-WERKS	C (4)	WERKS	Plant			plant	FR_MD_ERP_KEY	PLANT	VAR-CHAR2(255 CHAR)
SY-SYSID	C (8)	SYSID	SAP System ID			systemId	FR_MD_ERP_KEY	SYSTEM_ID	VAR-CHAR2(255 CHAR)
AUFK-AUFNR	C (12)	AUFNR	Order number			orderNumber	FR_MD_OPERATION	ORDER_NUMBER	VARCHAR2(32 CHAR)
Always 0	C (1)	ASPLT	Order split (Always 0, no SAP Split)			orderSplit	FR_MD_OPERATION	ORDER_SPLIT	VARCHAR2(8 CHAR)

AFVC-RUECK	N (10)	RUECK	Confirmation number of the operation			confirmationNumber	FR_MD_OPERATION	CONFIRMATION_NUMBER	VAR-CHAR2(255 CHAR)
AFVC-VORNR	C (4)	VORNR	Operation			operation-Number	FR_MD_OPERATION	OPERATION_NUMBER	VARCHAR2(32 CHAR)
Always 0	C (1)	VSPLT	Operation split			operationSplit	FR_MD_OPERATION	OPERATION_SPLIT	VARCHAR2(8 CHAR)
AFVC-LTXA1 and AFVC-LTXA2	C (80)	LTXA1	Operation short description			description	FF_MD_LITERAL_TEXT	TEXT	VAR-CHAR2(1024 CHAR)
AFVV-SSAVD	DD.MM.YYYY (10)	SSAVD	Latest planned start date, target start			plan-StartDate	FR_MD_OPERATION	TARGET_START_DATE	TIMESTAMP(6)
AFVV-SSAVZ if 24:00 then 23:59 will be set, Format HH:MM	HH.MM (5)	SSAVZ	Latest planned start time			planStart-Time	FR_MD_OPERATION		
AFVV-SSEDD	DD.MM.YYYY (10)	SSEDD	Latest planned end date, target end			planEndDate	FR_MD_OPERATION	TARGET_END_DATE	TIMESTAMP(6)
AFVV-SSEDZ if 24:00 then 23:59 will be set, Format HH:MM	HH.MM (5)	SSEDZ	Latest planned end time			planEnd-Time	FR_MD_OPERATION		
AFVV-MGVRG quantity in the unit AFVV-MEINH	C (17)	MGVRG	Target quantity			targetQuantity	FR_MD_OPERATION	TARGET_QUANTITY	FLOAT
AFVV-MEINH	C (3)	MEINH	Quantity unit			displayQuantityUnit	FR_MD_OPERATION	DISPLAY_QUANTITY_UNIT	VAR-CHAR2(255 CHAR)
AFVV-LMNGA	C (17)	LMNGA	Reported yield quantity			yield quantity	FR_MD_OPERATION	ERP_YIELD_QUANTITY	FLOAT

AFVV-XMNGA	C (17)	XMNGA	Reported scrap quantity		scrap quantity	FR_MD_OPERATION	ERP_SCRAP_QUANTITY	FLOAT
AFVV-RMNGA	C (17)	RMNGA	Reported rework quantity		rework quantity	FR_MD_OPERATION	ERP_REWORK_QUANTITY	FLOAT
Calculated from AFVV-MGVRG / 100 * AFVC-AUFAK in the unit AFVV-MEINH	C (17)	AUSSS	Planned scrap quantity		planned scrap quantity	FR_MD_OPERATION	ERP_PLANNED_SCRAP_QUANTITY	FLOAT
AFVV-BMSCH, unit AFVV-MEINH	C (17)	BMSCH	Base quantity		piece time factor	FR_MD_OPERATION	TIME_PER_UNIT	NUMBER(19, 0)
AFVV-VGW01	C (17)	VGW01	Setup time for machine (standard value 1 in SAP, in SAP freely configurable, i.e. does not have to be setup time!)		standard value 1	FR_MD_OPERATION	STANDARD_VALUE_1 / STANDARD_VALUE_1_MS	VAR-CHAR2(255 CHAR)
AFVV-VGE01	C (3)	VGE01	Setup time unit (unit from SAP standard value 1)		standard unit 1	FR_MD_OPERATION	STANDARD_UNIT_1	VAR-CHAR2(255 CHAR)
AFVV-VGW02	C (17)	VGW02	Personnel setup time (standard value 2 in SAP, in SAP freely configurable, i.e. does not have to be personnel setup time!)		standard value 2, piece time	FR_MD_OPERATION	TIME_PER_UNIT / STANDARD_VALUE_2 / STANDARD_VALUE_2_MS	NUMBER(19, 0) / VAR-CHAR2(255 CHAR)

AFVV-VGE02	C (3)	VGE02	Personnel setup time unit (unit from SAP standard value 3)		standard unit 2, piece time unit	FR_MD_OPERATION	TIME_PER_UNIT / STANDARD_UNIT_2	NUMBER(19, 0) / VAR-CHAR2(255 CHAR)
AFVV-VGW03	C (17)	VGW03	Processing time for machine (standard value 3 in SAP, in SAP freely configurable, i.e. does not have to be machine time!)		standard value 3	FR_MD_OPERATION	STANDARD_VALUE_3 / STANDARD_VALUE_3_MS	NUMBER(19, 0)
AFVV-VGE03	C (3)	VGE03	Machine time unit (unit from SAP standard value 3)		standard unit 3	FR_MD_OPERATION	STANDARD_UNIT_3	VAR-CHAR2(255 CHAR)
AFVV-VGW04	C (17)	VGW04	Processing time personnel (standard value 4 in SAP, in SAP freely configurable, i.e. does not have to be processing time personnel!)		standard value 4	FR_MD_OPERATION	STANDARD_VALUE_4 / STANDARD_VALUE_4_MS	VAR-CHAR2(255 CHAR)
AFVV-VGE04	C (3)	VGE04	Personnel time unit (unit from SAP standard value 4)		standard unit 4	FR_MD_OPERATION	STANDARD_UNIT_4	VAR-CHAR2(255 CHAR)
AFVV-VGW05	C (17)	VGW05	Activity type 5 (standard value 5 in SAP, in SAP freely configurable which time		standard value 5	FR_MD_OPERATION	STANDARD_VALUE_5 / STANDARD_VALUE_5_MS	VAR-CHAR2(255 CHAR)

			this should represent)					
AFVV-VGE05	C (3)	VGE05	Time unit (unit from SAP standard value 5)		standard unit 5	FR_MD_OPERATION	STANDARD_UNIT_5	VAR-CHAR2(255 CHAR)
AFVV-VGW06	C (17)	VGW06	Activity type 6 (standard value 6 in SAP, in SAP freely configurable which time this should represent)		standard value 6	FR_MD_OPERATION	STANDARD_VALUE_6 / STANDARD_VALUE_6_MS	VAR-CHAR2(255 CHAR)
AFVV-VGE06	C (3)	VGE06	Time unit (unit from SAP standard value 6)		standard unit 6	FR_MD_OPERATION	STANDARD_UNIT_6	VAR-CHAR2(255 CHAR)
CRHD-ARBPL (read with PLPO-ARBID) currently not filled. Would normally be the workplace from the work plan (which was the template for the operation of the order). Customer specific, will be filled in expansion methods	C (8)	ARBPLP	Planned workplace (workplace ERP) – not relevant information		planned workplace	FR_MD_OPERATION	PLANNED_WORKPLACE_ID	NUMBER(19, 0)
CRHD-ARBPL, with the AFVC-ARBID the	C (8)	ARBPLI	Target ERP work-center		target workplace	FR_MD_OPERATION	TARGET_WORKPLACE_ID	NUMBER(19, 0)

workplace (label) will be read from the table CRHD. Is the real workplace from the operation			(capacity group or single equipment)					
CRHD-ARBPL (with CRHS-OB-JID_UP). With AFVC-ARBID the hierarchy table CRHS will be read = superior workplace from the ERP hierarchy	C (8)	ARBPLG	Target workplace group (superior hierarchy from the ERP) – not relevant information		workplace group	FR_MD_OPERATION	TARGET_WORKGROUP_ID	NUMBER(19, 0)
AFVC-LOART	C (4)	LOART	Wage type		wage type	FR_MD_OPERATION	WAGE_TYPE	VAR-CHAR2(255 CHAR)
AFVC-LOGRP	C (3)	LOGRP	Wage group		wage group	FR_MD_OPERATION	WAGE_GROUP	VAR-CHAR2(255 CHAR)
TJ02T-TXT04 in German	C (80)	STATU	Order status text e.g. FREI SSAP... (SAP system status, operation)		erp status	FR_MD_OPERATION	ERP_STATUS	VAR-CHAR2(255 CHAR)
TJ02T-ISTAT (internal system status codes)	C (80)	STATU_ID	ERP Status_ID Z.B. I0002 (SAP system status ID operation)		erp status id	FR_MD_OPERATION	ERP_STATUS_IDS	VAR-CHAR2(255 CHAR)
TJ02T-TXT04 in the language of the table /FFMES/AUFTR	C (80)	STATU_LA	ERP status text (operation) in		erp status language	FR_MD_OPERATION	ERP_STATUS_LA	VAR-CHAR2(255 CHAR)

			the order language (SAP system status)					
Empty or X, will be set to X if status SPER is set	C (1)	SPRKZ	Lock indicator		lock indicator	FR_MD_OPERATION	LOCK_INDICATOR	VAR-CHAR2(255 CHAR)
Calculated from AFKO-GAMNG * AFPO-UEBTO / 100 in UOM AFVV-MEINH	C (17)	UELIT	Overdelivery tolerance		overdelivery quantity	FR_MD_OPERATION	OVERDELIVERY_QUANTITY	FLOAT
If AFPO-UEBTK = 'X' then N, else Y	C (1)	UELIC	Overdelivery check		overdelivery check	FR_MD_OPERATION	OVERDELIVERY_CHECK	NUMBER(1, 0)
Calculated from AFKO-GAMNG * AFPO-UNTTO / 100, in UOM AFVV-MEINH	C (17)	UNLIT	Underdelivery tolerance		underdelivery quantity	FR_MD_OPERATION	UNDERDELIVERY_QUANTITY	FLOAT
Always N	C (1)	UNLIC	Underdelivery check		underdelivery check	FR_MD_OPERATION	UNDERDELIVERY_CHECK	NUMBER(1, 0)
Empty	C (1)	STORA	Malfunction operation indicator		malfunction operation indicator	FR_MD_OPERATION	MALFUNCTION_OP_INDICATOR	VAR-CHAR2(255 CHAR)
Empty	C (4)	STEUS	Control key		control key	FR_MD_OPERATION	CONTROL_KEY	VAR-CHAR2(255 CHAR)
Empty	C (1)	RMDKZ	Label Confirmation provided		confirmation	FR_MD_OPERATION	CONFIRMATION	VAR-CHAR2(255 CHAR)
Empty	C (1)	BEDAV	Label Post-process		rework operation indicator	FR_MD_OPERATION	REWORK_OPERATION_NUMBER	VAR-CHAR2(255 CHAR)

Empty	C (4)	MVORN	Operation number (material)			operation material number	FR_MD_OPERATION	OPERATION_MATERIAL_NUMBER	VAR-CHAR2(255 CHAR)
AFPO-MATNR	C (18)	MATNR	Material number			material number	FR_MD_MATERIAL	MATERIAL_NUMBER	VAR-CHAR2(255 CHAR)
MAKT-MAKTX or AUFK-KTEXT	C (40)	MTEXT	Material short description			material description	FR_MD_LITERAL_TEXT	TEXT	VAR-CHAR2(1024 CHAR)
Empty	C (1)	PALKZ	Pallet note identifier			pallet note identifier	FR_MD_OPERATION	PALLET_NOTE_IDENTIFIER	VAR-CHAR2(255 CHAR)
Empty	C (2)	VASTA	Status predecessor operation			status predecessor operation	FR_MD_OPERATION	STATUS_PREDECESSOR_OPERATION	VAR-CHAR2(255 CHAR)
Only used at repetitive manufacturing, else always empty	C (4)	VERID	Production version			production version	FR_MD_OPERATION	PRODUCTION_VERSION	VAR-CHAR2(255 CHAR)
Empty	C (22)	ZEINR	Document number			document number	FR_MD_OPERATION	DOCUMENT_NUMBER	VAR-CHAR2(255 CHAR)
Empty	C (3)	ZEIAR	Document type			document type	FR_MD_OPERATION	DOCUMENT_TYPE	VAR-CHAR2(255 CHAR)
Empty	C (2)	ZEIVR	Document version			document version	FR_MD_OPERATION	DOCUMENT_VERSION	VAR-CHAR2(255 CHAR)
AFVC-PLNFL	C (6)	PLNFL	Sequence			sequence	FR_MD_OPERATION	OPERATION_SEQUENCE	VAR-CHAR2(255 CHAR)
If AFVC-PLNFL = '000000' then empty else 1	C (1)	FLGAT	Sequence category			sequence category	FR_MD_OPERATION	SEQUENCE_CATEGORY	VAR-CHAR2(255 CHAR)

TJ30T-TXT04 (in German)	C (40)	ASTTX	User status (text), operation			user status	FR_MD_OPERATION	USER_STATUS	VAR-CHAR2(255 CHAR)
Empty	C (4)	EVORN	Fill-in operation for alternative sequence			alternate operation number	FR_MD_OPERATION	ALTERNATE_OPERATION_NUMBER	VAR-CHAR2(255 CHAR)
Empty	C (4)	RVORN	Return operation for alternative sequence			confirmation operation number	FR_MD_OPERATION	CONFIRM_OPERATION_NUMBER	VAR-CHAR2(255 CHAR)
AUFK-AUART	C (4)	AUART	Order type			order type	FR_MD_OPERATION	ORDER_TYPE	VAR-CHAR2(255 CHAR)
TJ30T-ESTAT	C (40)	ASTTX_ID	User status codes (internal) e.g. E00001, operation		-	-	-	-	-
TJ30T-TXT04 in order language (from the table /FFMES/AUFTR)	C (40)	ASTTX_LA	User status text in the order language, operation		-	-	-	-	-

2.2.3.4 Long Texts for Operations

These data are transferred to the subsystem in the IDOC segment /FFMES/SAFOTX with set type **AFOTX**, if it was preset in the configuration. The configuration is set-up via the table /FFMES/AUFTR with the flag **AKTIV_TXT**.

The data structure is listed in Table 6.

Table 6: Data structure OP long texts (source SAP (ERP) and target FORCAM FORCE™)

SAP (ERP)			Label	FORCAM FORCE™					
				Mandatory fields			Persist		
SAP field	Type (Length)	IDOC XML SAP		Down-load	Dele-tion record	Internal XML trans-formation	Table	Col-umn	Type
AFOTX	C (5)	SART	Record type (Here: AFOTX)			-	-	-	-
S	C (1)	VAKZ	Processing flag (Always S)			-	-	-	-
/FFMES/AUFTR-SPRAS	C (1)	SPRAS	Language key from the table /FFMES/AUFTR			language (note: not needed or filled since predefined by XSL)	-	-	
incremented	N (4)	LFDNR	Ongoing number			(Note: not needed or filled since the texts will be concatenated)	-	-	
Long text about operation: Text ID = AVOT Text name = AFVC-MANDT + AFVC-AUFPL + AFVC-APLZL Text object = AUFK	C (80)	AFOTXT	Operation text (text line) (will be concatenated)			operationText	FR_MD_LITERAL_TEXT	TEXT	VAR-CHAR2(1024 CHAR)

2.2.3.5 OP Components

These data are transferred to the subsystem in the IDOC segment /FFMES/SAFOKO with set type **AFOKO**, if it was preset in the configuration. The configuration is set-up via the table /FFMES/AUFTR with the flag **AKTIV_KOM**. No dummy assemblies (RESB-DUMPS = X) and also no text positions (TXTPS = X) are created by default. In case of a co-product no set will be sent either (RESB-KZKUP = X).

The data structure is listed in Table 7.

Table 7: Data structure OP components (source SAP (ERP) and target FORCAM FORCE™)

SAP (ERP)			Label	FORCAM FORCE™					
				Mandatory fields			Persist		
SAP field	Type (Length)	IDOC XML SAP		Down-load	Deletion record		Internal XML transformation	Table	Column
AFOKO	C (5)	SART	Record type (Here: AFOKO)			-	-	-	-
If RESB-XLOEK = 'X' then D else S	C (1)	VAKZ	Processing flag			-	-	-	-
AUFK-MANDT	C (3)	MANDT	Client			client	FR_MD_ERP_KEY	CLIENT	VARCHAR2(255 CHAR)
AUFK-BUKRS	C (4)	BUKRS	Company Code			companyCode	FR_MD_ERP_KEY	COMPANY_CODE	VARCHAR2(255 CHAR)
AUFK-WERKS	C (4)	WERKS	Plant			plant	FR_MD_ERP_KEY	PLANT	VARCHAR2(255 CHAR)
SY-SYSID	C (8)	SYSID	SAP system ID			systemId	FR_MD_ERP_KEY	SYSTEM_ID	VARCHAR2(255 CHAR)
AUFK-AUFNR	C (12)	AUFNR	Order number			orderNumber	FR_MD_OPERATION	ORDER_NUMBER	VARCHAR2(32 CHAR)
Always 0	C (1)	ASPLT	Order split (always 0, no SAP split)			orderSplit	FR_MD_OPERATION	ORDER_SPLIT	VARCHAR2(8 CHAR)
AFVC-RUECK	N (10)	RUECK	Confirmation number of the operation			confirmation-Number	FR_MD_OPERATION	CONFIRMATION_NUMBER	VARCHAR2(255 CHAR)
AFVC-VORNR	C (4)	VORNR	Operation			operation-Number	FR_MD_OPERATION	OPERATION_NUMBER	VARCHAR2(32 CHAR)

Always 0	C (1)	VSPLT	Operation split			operationSplit	FR_MD_OPERATION	OPERATION_SPLIT	VARCHAR2(8 CHAR)
RESB-RSNUM	C (10)	RSNUM	Number of the reservation/dependent requirements			reservation-Number		RESERVATION_NUMBER	VARCHAR2(255 CHAR)
RESB-RSPOS	C (4)	RSPOS	Position number of the reservation/dependent requirements			position-Number		POSITION_NUMBER	VARCHAR2(255 CHAR)
RESB-MATNR	C (18)	MATNR	Component number			component-Number		COMPONENT_NUMBER	VARCHAR2(255 CHAR)
RESB-WERKS	C (4)	MWERK	Plant from the reservation			reservation-Plant		RESERVATION_PLANT	VARCHAR2(255 CHAR)
RESB-LGORT	C (4)	LGORT	Storage location			storageLocation		STORAGE_LOCATION	VARCHAR2(255 CHAR)
RESB-CHARG	C (10)	CHARG	Charge			batchNumber		BATCH_NUMBER	VARCHAR2(255 CHAR)
RESB-SOBKZ	C (1)	SOBKZ	Special stock indicator			specialStockIndicator		SPECIAL_STOCK_INDICATOR	VARCHAR2(255 CHAR)
RESB-BDTER	TT.MM.JJJJ	BDTER	Requirements date of the component			requirementsDate		REQUIREMENTS_DATE	TIMESTAMP(6)
RESB-BDMNG	C (15)	BDMNG	Requirement quantity			requirementQuantity		REQUIREMENT_QUANTITY	FLOAT
RESB-MEINS	C (3)	MEINS	Basic quantity unit			baseUnitOf-Measure		BASE_UNIT_OF_MEASURE	VARCHAR2(255 CHAR)
RESB-SERNR	C (8)	SERNR	Serial number			serialNumber		SERIAL_NUMBER	VARCHAR2(255 CHAR)

MAKT-MAKTX for RESB-MATNR in lan- guage from the table /FFMES/AUFTR	C (40)	KTX01	Name technical object (article text)			description		TEXT	VAR- CHAR2(1024 CHAR)
RESB-LGNUM	C (3)	LGNUM	Warehouse number/ware- house complex			warehouse- Number		WAREHOUSE_NUMBER	VARCHAR2(255 CHAR)
RESB-LGTYP	C (3)	LGTYP	Warehouse type			warehouse- Type		WAREHOUSE_TYPE	VARCHAR2(255 CHAR)
RESB-LGPLA	C (10)	LGPLA	Storage bin			storageBin		STORAGE_BIN	VARCHAR2(255 CHAR)
MARA-MTART	C (4)	MTART	Material type			materialType		MATERIAL_TYPE	VARCHAR2(255 CHAR)
MARC-RGEKZ	C (1)	RGEKZ	Indicator of a retrograde withdrawal			retro- gradeWith- drawalIndica- tor		RETROGRD_WITH- DRAWAL_INDICATOR	VARCHAR2(255 CHAR)
MARC-SCHGT	C (1)	SCHGT	Indicator for bulk material			bulkMaterial- Indicator		BULK_MATERIAL_INDI- CATOR	VARCHAR2(255 CHAR)
MARA-MATKL	C (9)	MATKL	Material group			materi- alGroup		MATERIAL_GROUP	VARCHAR2(255 CHAR)
AFVC-APLFL	C (6)	PLNFL	Planning se- quence			-	-	-	-

2.2.3.6 Production Resources and Tools

These data are transferred to the subsystem in the IDOC segment /FFMES/SAUFFH with the set type AUFFH, if it was preset in the configuration. The configuration is set-up via the table /FFMES/AUFTR with the flag **AKTIV_FHM**.

The data structure is listed in Table 8.

Table 8: Data structure production resources and tools (source SAP (ERP) and target FORCAM FORCE™)

SAP (ERP)			Label	FORCAM FORCE™					
				Mandatory fields			Persist		
SAP field	Type (Length)	IDOC XML SAP		Down-load	Dele-tion record	Internal XML transfor-mation	Table	Column	Type
AUFFH	C (5)	SART	Record type (Here: AFOKO)			-	-	-	-
If AFFH-LOEKZ = '' then S, else D	C (1)	VAKZ	processing flag			-	-	-	-
AUFK-MANDT	C (3)	MANDT	Client			client	FR_MD_ERP_KEY	CLIENT	VAR-CHAR2(255 CHAR)
AUFK-BUKRS	C (4)	BUKRS	Company code			companyCode	FR_MD_ERP_KEY	COMPANY_CODE	VAR-CHAR2(255 CHAR)
AUFK-WERKS	C (4)	WERKS	Plant			plant	FR_MD_ERP_KEY	PLANT	VAR-CHAR2(255 CHAR)
SY-SYSID	C (8)	SYSID	SAP system ID			systemId	FR_MD_ERP_KEY	SYSTEM_ID	VAR-CHAR2(255 CHAR)
AUFK-AUFNR	C (12)	AUFNR	Order number			orderNumber	FR_MD_OPERATION	ORDER_NUMBER	VARCHAR2(32 CHAR)
Always 0	C (1)	ASPLT	Order split			orderSplit	FR_MD_OPERATION	ORDER_SPLIT	VARCHAR2(8 CHAR)

			(Always 0, split in the FORCAM language use no SAP split)					
AFVC-RUECK	N (10)	RUECK	Confirmation number of the operation			Not mapped anymore		
AFVC-VORNR	C (4)	VORNR	Operation			operation-Number	OPERATION_NUMBER	VARCHAR2(32 CHAR)
Always 0	C (1)	VSPLT	Operation split			operationSplit	OPERATION_SPLIT	VARCHAR2(8 CHAR)
AFFH-PSNFH	N (4)	LFDNR	Ongoing number of the long text position			sequence	FR_MD_PRODUCTION_RESOURCE_SOURCE_TOOL	PRT_SEQUENCE
M (Material) D (Document) E (Equipment) S (without material)	C (1)	FHART	Type of the production resources and tools: M (Material) D (Document) E (Equipment) S (without material)			type	FR_MD_PRODUCTION_RESOURCE_SOURCE_TOOL	PRT_TYPE
CRVM_A-MATNR or CRVD_A-DOKNR or CRVE_A-EQUNR or CRVS_A-SFHNR	C (18)	MATNR	FHM number			number	FR_MD_PRODUCTION_RESOURCE_SOURCE_TOOL	PRT_NUMBER
MAKT-MAKTX DRAT-DKXTX EQKT-EQKTX CRTX-KTEXT	C (40)	FHTXT	FHM Short description			description	FR_MD_LITERAL_TEXT	TEXT
AFFH-MGVGW (N 9,3)	C (13)	MGVGW	FHM quantity			quantity	FR_MD_PRODUCTION_RESOURCE_SOURCE_TOOL	QUANTITY
								FLOAT

AFFH-MGEINH	C (3)	MGEINH	Quantity unit for display			displayQuantityUnit	FR_MD_PRODUCTION_RE-SOURCE_TOOL	UNIT	VAR-CHAR2(255 CHAR)
CRFH-FGRU1	C (80)	STATU	Order status			orderStatus	FR_MD_PRODUCTION_RE-SOURCE_TOOL	PRT_GROUP	VAR-CHAR2(255 CHAR)
AFVC-PLNFL	C (6)	PLNFL	Planning sequence			-	-	-	-

2.2.4 Machine Capacities / Shift Plans

Shift data for a workplace with n breaks can be transferred to the connected FORCAM FORCE™ with the program **/FFMES/TRANSFER_SHIFT_DATA**. The available capacity of the workplaces is selected. Besides the selection via the time horizon, additional selections via workplace, workplace type, sub-system (BDE group), editor's name and edit date, blocking indicator deletion indicator of the workplace are offered. Also, those shift capacities can be transferred, that overload the plant calendar (capacity on days that are usually not workdays). The reading of a reference offer for the workplace capacity is possible.

One segment for the actual shift data is created (see data structure in Table 9) and n segments for the associated breaks according to the shift definition (see data structure in Table 10).

Table 9: Data structure Shift Segment tools (source SAP (ERP) and target FORCAM FORCE™)

SAP (ERP)			Label	FORCAM FORCE™					
				Mandatory fields		Persist			
SAP field	Type (Length)	IDOC XML SAP		Down-load	Dele-tion record	Internal XML trans-formation	Table	Column	Type
AUFFH	C (5)	SART	Record type (Here: SCHIC)	X		-	-	-	-
S	C (1)	VAKZ	Processing flag (Always S)	X		-	-	-	-
AUFK-MANDT	C (3)	MANDT	Client	X		client	FR_MD_ERP_KEY	CLIENT	VAR-CHAR2(255 CHAR)
AUFK-BUKRS	C (4)	BUKRS	Company code	X		compa-nyCode	FR_MD_ERP_KEY	COM-PANY_CODE	VAR-CHAR2(255 CHAR)
AUFK-WERKS	C (4)	WERKS	Plant	X		plant	FR_MD_ERP_KEY	PLANT	VAR-CHAR2(255 CHAR)
SY-SYSID	C (8)	SYSID	SAP system ID	X		systemId	FR_MD_ERP_KEY	SYSTEM_ID	VAR-CHAR2(255 CHAR)
CRHD-ARBPL	C (8)	ARBPL	Workplace	X		workplace	FR_MD_SHIFT	WORK-PLACE_ID (normalized)	
Date from selection time horizon	C (10) DD.MM.YYYY	VALID_FR	Valid from	X		shiftDate	FR_MD_SHIFT		
KAPA-TPROG	C (4)	KAPTPROG	Shift type	X		shiftTagCode	FR_MD_SHIFT		

			(Shift program, e.g. shift code for early shift etc.)						
Monday, Tuesday, Sunday (written down in the re- spective logon language) determined by FB	C (10)	WEEKDAY	Weekday						
TC37A-BEGZT	C (5) (HH:MM)	SHIFT_STR	Start	X		startTime	FR_MD_SHIFT	START_TS	
TC37A-ENDZT	C (5) (HH:MM)	SHIFT_END	End	X		endTime	FR_MD_SHIFT	END_TS	

Table 10: Data Structure Break Segment (source SAP (ERP) and target FORCAM FORCE™)

SAP (ERP)			Label	FORCAM FORCE™					
				Mandatory fields			Persist		
SAP field	Type (Length)	IDOC XML SAP		Download	Deletion record	Internal XML transformation	Table	Column	Type
	C (5)	SART	Record type (Here: PAUSE)			-	-	-	-
	C (1)	VAKZ	Processing flag (Always I)			-	-	-	-
SY-MANDT	C (3)	MANDT	Client			client	FR_MD_ERP_KEY	CLIENT	VARCHAR2(255 CHAR)
T001K-BUKRS	C (4)	BUKRS	Company code			companyCode	FR_MD_ERP_KEY	COMPANY_CODE	VARCHAR2(255 CHAR)
CRHD-WERKS	C (4)	WERKS	Plant			plant	FR_MD_ERP_KEY	PLANT	VARCHAR2(255 CHAR)
SY-SYSID	C (8)	SYSID	SAP system ID			systemId	FR_MD_ERP_KEY	SYSTEM_ID	VARCHAR2(255 CHAR)
TC37A-BEGDA	C (8)	ARBPL	Workplace			workplace	FR_MD_SHIFT	WORKPLACE_ID (normalized)	
KAPA-TPROG	C (4)	KAPTPROG	Shift type			shiftTagCode	FR_MD_SHIFT		
TC37P-PAUNR	C (3)	AUFNR	Number of the break (pause)			-	-	-	-
TC37P-PAUBEG	C (5) HH:MM	AUFNR	Break (pause) begin	X		startTime			
TC37P-PAUEND	C (5) HH:MM	AUFNR	Break (pause) end	X		endTime			

2.2.5 Personnel Data (Mini Master Data Personnel)

Personnel mini master data can be transferred to the connected FORCAM FORCE™ with the program **/FFMES/TRANSFER_HR_DATA**. In this process a selection via a certain personnel area is possible. All personnel numbers that belong to this personnel area are transferred in a joint IDOC. The transfer of single personnel numbers is not possible.

Data of the following info types are used:

- Info type 0000 Measures
- Info type 0001 Org. assignment
- Info type 0002 Personal data
- Info type 0050 Time recording

One IDOC segment for the personnel data of each one personnel number is created.

The data structure is listed in Table 11.

Table 11: Data structure personnel data segment (source SAP (ERP) and target FORCAM FORCE™)

SAP (ERP)			Label	FORCAM FORCE™					
				Mandatory fields			Persist		
SAP field	Type (Length)	IDOC XML SAP		Download	Dele-tion rec-ord	Internal XML trans-forma-tion	Table	Column	Type
	C (5)	SART	Record type (Here: PERS)	X		-	-	-	-
	C (1)	VAKZ	Processing flag (Always I)	X		-	-	-	-
PA0001-MANDT	C (3)	MANDT	Client	X		client	FR_MD_ERP_KEY	CLIENT	VAR-CHAR2(255 CHAR)
PA0001-BUKRS	C (4)	BUKRS	Company code	X		compa-nyCode	FR_MD_ERP_KEY	COMPANY_CODE	VAR-CHAR2(255 CHAR)
PA0001-WERKS	C (4)	WERKS	Plant (person-nel area)	X		plant	FR_MD_ERP_KEY	PLANT	VAR-CHAR2(255 CHAR)
SY-SYSID	C (8)	SYSID	SAP system ID	X		systemId	FR_MD_ERP_KEY	SYSTEM_ID	VAR-CHAR2(255 CHAR)
PA0001-PERNR	C(8)	PERNR	Personnel number	X		personnel-Number	FR_MD_PERSON	PERSONNEL_NUMBER	VAR-CHAR2(255 CHAR)
PA0050-ZAUSW	C(8)	ZAUSW	Identification number	X		identifica-tionNumber	FR_MD_PERSON	IDENTIFICA-TION_NUMBER	VAR-CHAR2(255 CHAR)

PA0002-VORNA	C(10)	VORNA	First name			firstName	FR_MD_PERSON	FIRST_NAME	VAR-CHAR2(255 CHAR)
PA0002-NACHN	C(20)	NACHN	Last name			lastName	FR_MD_PERSON	LAST_NAME	VAR-CHAR2(255 CHAR)
PA0001-KOSTL	C(10)	KOSTL	Cost center			costCenter	FR_MD_PERSON	COST_CENTER	VAR-CHAR2(255 CHAR)

3 Upload Interface

3.1 General

Feedback events (event sequences) and related structured data are reported back from FORCAM FORCE™ to the ERP with the help of upload processes.

The event triggers are located in the underlying FORCAM FORCE™ system. For a feedback process to an ERP system, an adequate web service has to be provided and used for the communication and data transfer (see Fig. 4).

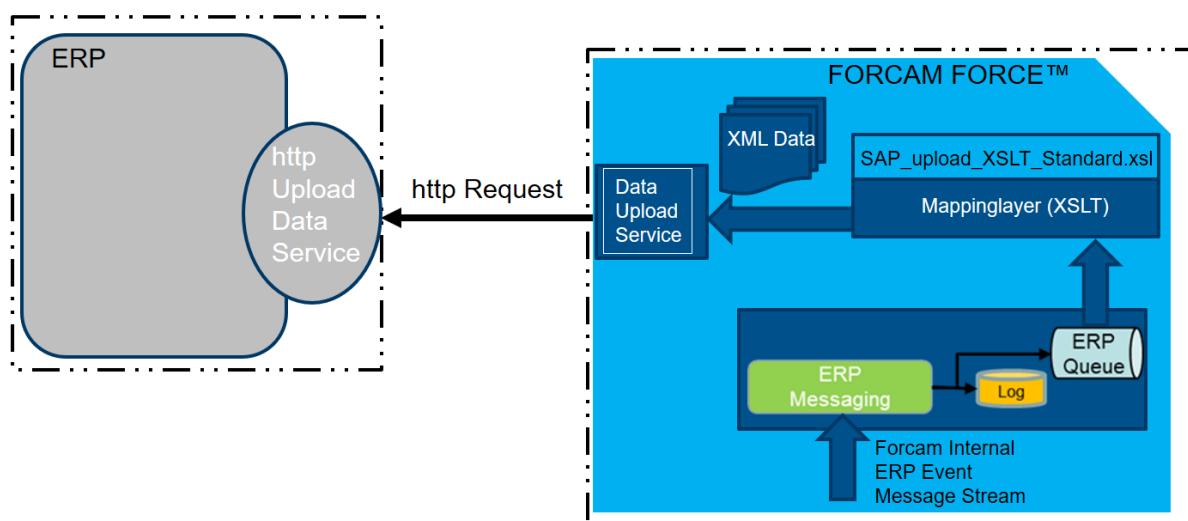


Fig. 4: General data upload principle structure (source FORCAM FORCE™) and target ERP

In FORCAM FORCE™ itself, a “FORCAM internal ERP message stream” is created by trigger events (like OP start, OP end, set-up start, set-up end, corrections) and is supplemented with corresponding event data. Events can also be supplemented with customer specific calculation data (“characteristic values”) via rule engine options.

The events are transmitted to the upload service of the Runtime by a persistent queue. Only confirmations that have been successfully transferred to SAP are acknowledged for the queue. Thus, a fail-over is ensured and failed transfers remain in the queue.

The single events from the “FORCAM internal ERP message stream” are transformed by a XSLT mapping layer into the respective XML telegrams with the help of a XSL scheme file. These XML telegrams are sent to the ERP by using an ERP upload service. In this case the ERP upload service in FORCAM FORCE™ uses the data service provided in the ERP system.

3.1.1 Use of Implemented RFC Connection SAP (ERP)

The communication connection between SAP (ERP) and FORCAM FORCE™ is executed in the current implementation and specification always by using an explicit RFC communication. The data upload process for this process is shown in Fig. 5.

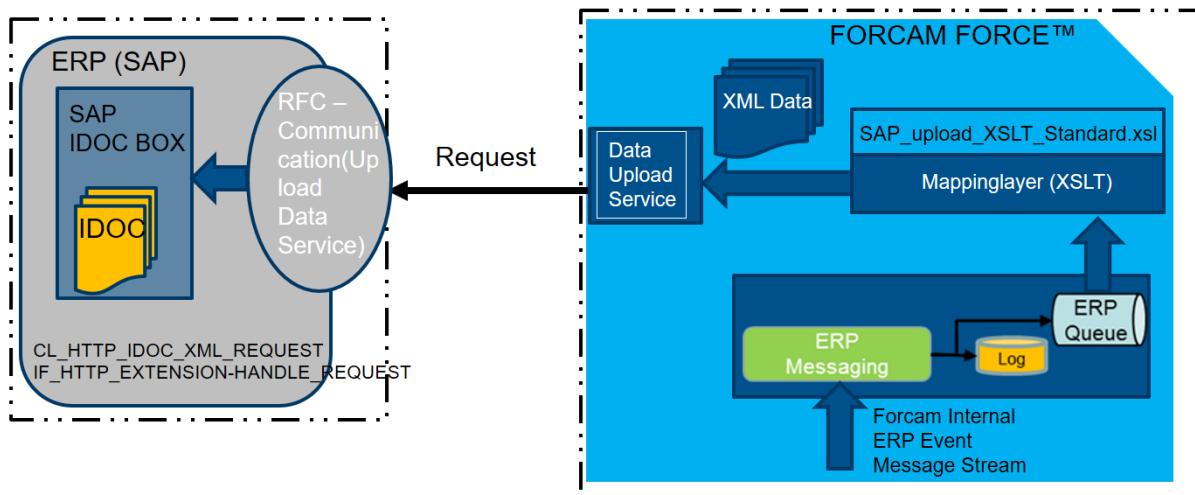


Fig. 5: Upload connection from FORCAM FORCE™ and target SAP (ERP)

The upload data service in SAP uses the IDOC Communication Framework by utilizing ALIAS (TRX SICF) with the current parameters

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

The data upload service of FORCAM FORCE™ logs itself in into the system user at the SAP gateway (secure user server) and the IDOC is transferred.

Each logic message (IDOC basic type) is connected to a feedback code via the partner agreement (transaction WE20). The feedback code is created in SAP and connected to an inbound function module that processes the incoming IDOCs.

3.2 FORCAM FORCE™ Feedback Set

All messages from FORCAM FORCE™ are created in form of a XML feedback set and reported back to the superior ERP.

When using SAP, all messages are transferred in the IDOC segment /FFMES/SRUECK. The corresponding processing program is found in table /FFMES/VERTEILER.

The set type (Feld SART) is used to distinct / define the different messages.

The data structure is listed in Table 12.

Table 12: Data structure feedback set in general for all message types (source FORCAM FORCE™ and SAP (ERP))

Name	Type	Length	XML
Set type	C	5	SART
Client	C	3	MANDT
Accounting area	C	4	BUKRS
Plant	C	4	WERKS
Feedback number	C	10	RUECK
Order number	C	12	AUFNR
OP	C	4	VORNR
Order split	C	3	ASPLT
Workplace	C	8	ARBPL
Yield	C	14	LMNGA
Scrap	C	14	XMNGA
Rework	C	14	RMNGA
Quantity unit	C	3	MEINH
Date of message	C	10	DATUM
Time of message	C	6	EZEIT
Personnel number	C	8	PERNR
Time recording ID card number	C	8	ZAUSW
Time factor	C	6	PROPZ
Break indicator	C	1	PAUSE
Order status	C	2	AUSTA
Machine status	C	3	MASTA
Activity type to be booked	C	30	LSTAR
Target start date	C	10	SOLLS
Target end date	C	10	SOLLE

Wage type	C	4	LOARR
Material number	C	18	MATNR
Cost center	C	10	KOSTL
Team number	C	10	ZTEAM
Shift (S or V)	C	1	ZFLAG
Cost accounting type	C	1	KOTYP
Account assignment object	C	25	KOOBJ
Workplace type	C	2	APTYP
Duration to be booked	C	5	DAUER
Machine hub time	C	14	VGWMZ
Target set-up time	C	14	VGWRZ
Shift day	C	10	SCHIT
Shift indicator	C	1	SCHIK
Cost center machine	C	10	KOSMA
Productive time	C	5	PZEIT
GUID / internal number	C	32	INTNR
Reason for disturbance	C	4	AGRUN
Disturbance initiator	C	10	AVERU
Rework reason	C	4	NGRUN
Rework initiator	C	10	NVERU
Co activity type	C	6	COLAR
Personnel object type	C	2	PEOBJ
GUID of message to be corrected	C	32	KORID
Correction type	C	1	KORTY
Date of correction	C	8	KORDT
Time of correction	C	6	KORZT

Set type of message to be corrected	C	5	KORSA
Time stamp	C	6	TSTMS
First factor	C	14	FAK01
Second factor	C	14	FAK02
Third factor	C	14	FAK03
Additional minutes	C	5	ZUMIN
Scrap reason	C	4	ZFSL1
Plant	C	4	ZBWK1
Cost center	C	10	ZBKS1
Workplace	C	8	ZBAP1
Material number component	C	18	KOMPO
Terminal ID	C	4	TERID
Feedback text	C	40	TEXT1
Version	C	4	VERID
Reservation number	C	10	RSNUM
Reservation position	C	4	RSPOS
Charge	C	10	CHARG
Palette	C	3	PALET
Delivery note date	C	10	LSDAT
Weight	C	14	GWCHT

With each set type, FORCAM FORCE™ sends the necessary fields to SAP that are needed for the particular booking. In most cases a feedback is booked in a production file in SAP with the help of the data sent in the IDOC.

The following fields are filled by default for the single messages and are explained in chapter 3.4 in Table 13.

3.3 Feedback Events

The following feedback events are used in the current FORCAM SAP-oriented ERP interface:

- Status event messages
 - OP start, OP end
- Duration event messages
 - Are generated in FORCAM FORCE™ only at OP end or shift end (OP overlaps shifts).
 - Per activity type: machine production time, machine set-up time
 - If personnel time: Per person and activity type
- Quantity messages
 - Asynchronous or single & || controlled by intervals
 - Separate message per quality type
 - Yield
 - Scrap quantity
 - Rework quantity
- Corrections
 - Delete / Insert
 - Restrictions have to be specified in detail towards NONE SAP and have to be supported by the ERP
 - Only possible with unique message number
- Key figures
 - Have to be defined according to interface characteristic ERP
 - E.g. current job-time calculation → Rule engine use
- Ad hoc maintenance via Shop Floor Terminal
 - Request maintenance
 - Start maintenance
 - Finish maintenance

The data structures of single feedback events are created and expanded following the object & inheritance thought. Each event contains generic standard content and corresponding further process hierarchies as well as a specific content.

The hierarchy of SAP message types is pictured below.

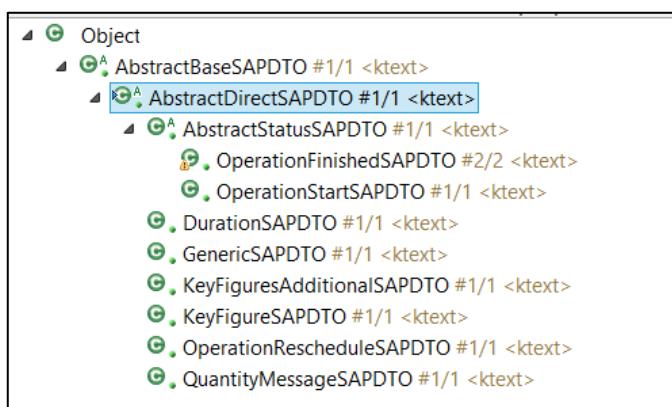


Fig. 6: Hierarchy of SAP message types

3.3.1 Mapping Layer XSLT (XSL Scheme Product Standard)

An XSL file is used for the transformation of data in the mapping layer as scheme for the product standard to SAP (for details see chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**).

3.4 Filled Tags in SAP IDOC Depending on the Set type

Table 13: Filled XML tag elements of the feedback set depending on the sent message type

Tag name	Meaning	Start message (OP-STR)	End message (OPEND)	Quantity message (QTYMG)	Time/duration message (DURAT)	Delete correction (REVMG)	Insert correction EVMG)	Maintainance request	Maintainance start	Maintainance end	
<EDI_DC40>	Header record										
<MANDT>003</MANDT>	Client	X	X	X	X	X	X	X	X	X	
<DOCNUM>-1859907614639058460</DOCNUM>	Document number	X	X	X	X	X	X	X	X	X	
<DIRECT>2</DIRECT>	Direction (Upload = 2)	X	X	X	X	X	X	X	X	X	
<IDOCTYP>/FFMES/R</IDOCTYP>	IDOC Type	X	X	X	X	X	X	X	X	X	
<MESTYP>/FFMES/MES-SAGE</MESTYP>	Message type	X	X	X	X	X	X	X	X	X	
<SNDPOR>FORCAMFF</SNDPOR>	Sender port = port of the FORCAM FORCE™ system via	X	X	X	X	X	X	X	X	X	

	which the data is sent									
<SNDPRT>LS</SNDPRT>	Sender partner (Art) = logical system	X	X	X	X	X	X	X	X	X
<SNDPRN>FORCAMFF</SNDPRN>	Sender partner number = logical system of the sender (FORCAM FORCE™)	X	X	X	X	X	X	X	X	X
<RCVPOR>SAPP01</RCVPOR>	Receiver port = port of the SAP system via which the Date is received	X	X	X	X	X	X	X	X	X
<RCVPRT>LS</RCVPRT>	Receiver partner (Art) = logical system	X	X	X	X	X	X	X	X	X
<RCVPRN>SUB_ALE</RCVPRN>	Receiver partner number = logical system of the receiver (SAP)	X	X	X	X	X	X	X	X	X
<SERIAL>00161321360000000007</SERIAL>	Serialization number	X	X	X	X	X	X	X	X	X

<_FFMES_-SRUECK SEG-MENT="1">		Reporting data									
<SART>QTYMG</SART>		Record type	OPSTR	OPEND	QTYMG	DURAT	REVMG	REVMG	MTREQ	MTSTA	MTFIN
<MANDT>003</MANDT>	Client	X	X	X	X	X	X	X	X	X	X
<BUKRS>7500</BUKRS>	Company code	X	X	X	X	X	X	X	X	X	X
<WERKS>7500</WERKS>	Plant	X	X	X	X	X	X	X	X	X	X
<SYSID>600</SYSID>	System ID	X	X	X	X	X	X	X	X	X	X
<RUECK>0016132136</RUECK>	Operation confirmation number	X	X	X	X	X	X	X	X	X	X
<AUFNR>008500013320</AUFNR>	Operation number	X	X	X	X	X***	X***				
<VORNR>0010</VORNR>	Operation number	X	X	X	X	X***	X***				
<ASPLT>0</ASPLT>	Order split	X	X	X	X	X***	X***				
<ARBPL>84915002</ARBPL>	Workplace	X	X	X	X	X	X				
<DATUM>04.05.2015</DATUM>	Reporting date	X	X	X	X						
<EZEIT>160019</EZEIT>	Reporting time	X	X	X	X						
<SCHIT>2015-05-04</SCHIT>	Shift date (shift day)	X	X	X	X		X				
<SCHIK>F</SCHIK>	Shift indicator	X	X	X	X		X*				
<INTNR>E6276E40F26511E49	Correction GUID	X	X	X	X		X*				

543C30B0AD90108</INTNR>										
<AUSTA></AUSTA>	Order status (phase code)	X	X							
<PERNR>00200038</PERNR>	Personnel number	X	X	X	X		X****			
<LSTAR>2</LSTAR>	Capacity type				X		X*			
<DAUER>30.5</DAUER>	Duration (time)				X		X*			
<ZFSL1></ZFSL1>										
<LMNGA>16.0</LMNGA>	Yield			X			X**			
<XMNGA>0.0</XMNGA>	Scrap quantity			X			X**			
<AGRUN>15</AGRUN>	Scrap reason			X			X**			
<RMNGA>0.0</RMNGA>	Rework quantity			X			X**			
<NGRUN>15</NGRUN>	Rework reason			X			X**			
<MEINH>SZT</MEINH>	Quantity unit			X			X**			
<MATNR>53031208520</MATNR>	Material number			X			X**			
<KORID></KORID>	Correction GUID (GUID of the message to be corrected)									
<KORTY></KORTY>	Correction type (delete, insert)					D	I			

<KORDT></KORDT>	Correction date						X			
<KORZT></KORZT>	Correction time						X			
<KORSA></KORSA>	Correction record type						X			
<CHARG></CHARG>	Batch number									
<NOTIFICATION_TYPE></NOTIFICATION_TYPE>	Message type (for maintenance)							X		
<EQUIPMENT></EQUIPMENT>	Machine number						X			
<REPORTER_ID></REPORTER_ID>	Reported by (personnel number)						X			
<MALFUNCTION_COMMENT></MALFUNCTION_COMMENT>	Message text (for maintenance)						X			
<MALFUNCTION_START_DATE></MALFUNCTION_START_DATE>	Start (datum) of the malfunction						X			
<MALFUNCTION_START_TIME></MALFUNCTION_START_TIME>	Start (time) of the malfunction						X			
<MALFUNCTION_CODE></MALFUNCTION_CODE>	Error code						X			
<MESPMID></MESPMID>	Reference number of						X	X	X	

	the maintenance message									
<MATERIAL_NUMBER></MATERIAL_NUMBER>	Product number of the operation							X		
<MATERIAL_DESCRIPTION></MATERIAL_DESCRIPTION>	Material description in default language							X		
<MAINTENANCE_START_DATE></MAINTENANCE_START_DATE>	Start (date) of the maintenance								X	
<MAINTENANCE_START_TIME></MAINTENANCE_START_TIME>	Start (time) of the maintenance								X	
<MALFUNCTION_END_DATE></MALFUNCTION_END_DATE>	End (date) of the maintenance									X
<MALFUNCTION_END_TIME></MALFUNCTION_END_TIME>	End (time) of the maintenance									X

(* = filled if DURAT correction, ** filled if quantity correction, *** filled if order related, **** filled if personnel related)

- ⓘ Correction messages are only generated for quantity messages and duration messages. In the case of deletion messages, the record to be deleted is referenced by its GUID. In addition, its record type is transmitted. Insert message contains the fields required for the insert type.

3.5 Example of a Quantity Report

```

<?xml version="1.0" encoding="UTF-8"?>
<_FFMES_R>
<IDOC BEGIN="1">
<EDI_DC40 SEGMENT="1">
<TABNAM>EDI_DC40</TABNAM>
<MANDT>003</MANDT>
<DOCTNUM>-1859907614639058460</DOCTNUM>
<DOCREL />
<STATUS />
<DIRECT>2</DIRECT>
<OUTMOD />
<EXPRSS />
<TEST />
<IDOCTYP>FFMES/R</IDOCTYP>
<CIMTYP />
<MESTYP>FFMES/MESSAGE</MESTYP>
<MESCOD />
<MESFCT />
<STD />
<STDVRS />
<STDMES />
<SNDPOR>FORCAMFF</SNDPOR>
<SNDRPT>LS</SNDRPT>
<SNDFC />
<SNDRN>FORCAMFF</SNDRN>
<SNDSAD />
<SNDLAD />
<RCVPOR>SAPP1</RCVPOR>
<RCVPRT>LS</RCVPRT>
<RCVPFC />
<RCVPRN>SUB_ALE</RCVPRN>
<RCVSAD />
<RCVLAD />
<CREDAT />
<CRETIM />
<REFINT />
<REFGRP />
<REFMES />
<ARCKEY />
<SERIAL>00161321360000000007</SERIAL>
</EDI_DC40>
<_FFMES_SRUECK SEGMENT="1">
<SART>QTYMG</SART>
<MANDT>003</MANDT>
<BUKRS>7500</BUKRS>
<WERKS>7500</WERKS>

```

```

<RUECK>0016132136</RUECK>
<AUFNR>008500013320</AUFNR>
<VORNR>0010</VORNR>
<ASPLT>0</ASPLT>
<ARBPL>84915002</ARBPL>
<DATUM>04.05.2015</DATUM>
<EZEIT>160019</EZEIT>
<SCHIT>2015-05-04</SCHIT>
<SCHIK />
<INTNR>E6276E40F26511E49543C30B0AD90108</INTNR>
<AUSTA />
<PERNR>00200038</PERNR>
<LSTAR />
<DAUER />
<ZFSL1 />
<LMNGA>16.0</LMNGA>
<XMNGA>0.0</XMNGA>
<RMNGA>0.0</RMNGA>
<MEINH>SZT</MEINH>
<MATNR>53031208520</MATNR>
<KORID />
<KORTY />
<KORDT />
<KORZT />
<KORSA />
</_-FFMES_-SRUECK>
</IDOC>
</_-FFMES_-R>

```

3.6 Serialization & Correction Capability

To guarantee a sequence order and a clear allocation of messages, a unique serialization is needed.

Correction capability can only be guaranteed, if each message has its own unique message number and if the superior ERP also allows cancellation processes (deletion / insertion of messages). A not-SAP correction process has to be completely specified for each ERP.

3.6.1 Workplace – Serialization

The SAP standard provides a 10-digit number = 4 digits channel number + 6 digits channel counter

- 4 digits channel number => 9999 workplaces
- 6 digits channel counter => counter to serialize reports up to 999999 reports

The channel numbers are managed in table BDRGIN (SAP standard table). FORCAM FORCE™ maintains one channel per workplace. The current channel statuses are requested from SAP per SOAP request at FORCAM FORCE™ (FFRuntime) system start, to initialize the APL serialization. To execute this alignment, a function module is called-up in SAP, which back-reports the current counter readings from table BDR-GIN.

⚠ This type of serialization should no longer be used if possible. As soon as an error occurs in the update chain for this workplace, all messages to be posted are also piled up for different operations on a workplace.

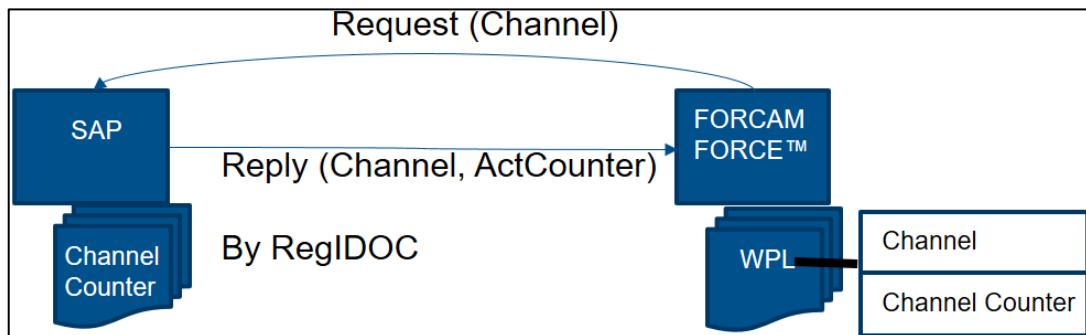


Fig. 7: Process of the workplace serialization

3.6.2 OP-Serialization

Not a synchronization with SAP channel registry table (BDRGIN), counter per feedback number is operated in FORCAM FORCE™ in DB table. The parameter CHECK_IDOC_PRED is active in SAP in table /FFMES/CONTROL.

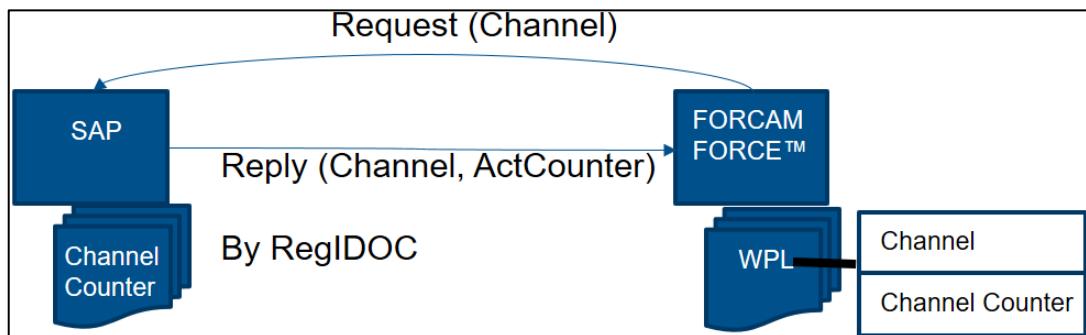


Fig. 8: OP-Serialization

4 Appendix

4.1 Upload XSLT Mapper

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- XSL for transforming the internal erpUpload format to an SAP IDOC XML. -->
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
    <xsl:output method="xml" indent="yes"/>
    <xsl:strip-space elements="*"/>

    <xsl:template match="ERPUpload">
        <_FFMES_R>
            <IDOC BEGIN="1">
                <EDI_DC40 SEGMENT="1">
                    <TABNAM>EDI_DC40</TABNAM>

                    <xsl:choose>
                        <xsl:when test="EDI_DC40_MANDT">
                            <MANDT><xsl:value-of select="EDI_DC40_MANDT"/></MANDT>
                        </xsl:when>
                        <xsl:otherwise>
                            <MANDT></MANDT>
                        </xsl:otherwise>
                    </xsl:choose>

                    <xsl:choose>
                        <xsl:when test="DOCNUM">
                            <DOCNUM><xsl:value-of select="DOCNUM"/></DOCNUM>
                        </xsl:when>
                        <xsl:otherwise>
                            <DOCNUM></DOCNUM>
                        </xsl:otherwise>
                    </xsl:choose>

                    <DOCREL></DOCREL>
                    <STATUS></STATUS>
                    <DIRECT>2</DIRECT>
                    <OUTMOD></OUTMOD>
                    <EXPRSS></EXPRSS>
                    <TEST></TEST>

                    <xsl:choose>
                        <xsl:when test="IDOCTYP">
                            <IDOCTYP><xsl:value-of select="IDOCTYP"/></IDOCTYP>
                        </xsl:when>

```

```

<xsl:otherwise>
    <IDOCTYP></IDOCTYP>
</xsl:otherwise>
</xsl:choose>

<CIMTYP></CIMTYP>

<xsl:choose>
    <xsl:when test="MESTYP">
        <MESTYP><xsl:value-of select="MES-
TYP"/></MESTYP>
    </xsl:when>
    <xsl:otherwise>
        <MESTYP></MESTYP>
    </xsl:otherwise>
</xsl:choose>

<MESCOD></MESCOD>
<MESFCT></MESFCT>
<STD></STD>
<STDVRS></STDVRS>
<STDMES></STDMES>

<xsl:choose>
    <xsl:when test="SNDPOR">
        <SNDPOR><xsl:value-of se-
lect="SNDPOR"/></SNDPOR>
    </xsl:when>
    <xsl:otherwise>
        <SNDPOR></SNDPOR>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="SNDPRT">
        <SNDPRT><xsl:value-of se-
lect="SNDPRT"/></SNDPRT>
    </xsl:when>
    <xsl:otherwise>
        <SNDPRT></SNDPRT>
    </xsl:otherwise>
</xsl:choose>

<SNDPFC></SNDPFC>

<xsl:choose>
    <xsl:when test="SNDPRN">
        <SNDPRN><xsl:value-of se-
lect="SNDPRN"/></SNDPRN>
    </xsl:when>
    <xsl:otherwise>

```

```

        <SNDPRN></SNDPRN>
    </xsl:otherwise>
</xsl:choose>

<SNDSAD></SNDSAD>
<SNDLAD></SNDLAD>

<xsl:choose>
    <xsl:when test="RCVPOR">
        <RCVPOR><xsl:value-of se-
lect="RCVPOR"/></RCVPOR>
    </xsl:when>
    <xsl:otherwise>
        <RCVPOR></RCVPOR>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="RCVPRT">
        <RCVPRT><xsl:value-of se-
lect="RCVPRT"/></RCVPRT>
    </xsl:when>
    <xsl:otherwise>
        <RCVPRT></RCVPRT>
    </xsl:otherwise>
</xsl:choose>

<RCVPFC></RCVPFC>

<xsl:choose>
    <xsl:when test="RCVPRN">
        <RCVPRN><xsl:value-of se-
lect="RCVPRN"/></RCVPRN>
    </xsl:when>
    <xsl:otherwise>
        <RCVPRN></RCVPRN>
    </xsl:otherwise>
</xsl:choose>

<RCVSAD></RCVSAD>
<RCVLAD></RCVLAD>
<CREDAT></CREDAT>
<CRETIM></CRETIM>
<REFINT></REFINT>
<REFGRP></REFGRP>
<REFMES></REFMES>
<ARCKEY></ARCKEY>

<xsl:choose>
    <xsl:when test="SERIAL">

```

```

        <SERIAL><xsl:value-of select="SERIAL"/></SE-
RIAL>
        </xsl:when>
        <xsl:otherwise>
            <SERIAL></SERIAL>
        </xsl:otherwise>
    </xsl:choose>

    </EDI_DC40>
    <_FFMES_-SRUECK SEGMENT="1">

    <xsl:choose>
        <xsl:when test="SART">
            <SART><xsl:value-of select="SART"/></SART>
        </xsl:when>
        <xsl:otherwise>
            <SART></SART>
        </xsl:otherwise>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="RUECK_MANDT">
            <MANDT><xsl:value-of se-
lect="RUECK_MANDT"/></MANDT>
        </xsl:when>
        <xsl:otherwise>
            <MANDT></MANDT>
        </xsl:otherwise>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="BUKRS">
            <BUKRS><xsl:value-of se-
lect="BUKRS"/></BUKRS>
        </xsl:when>
        <xsl:otherwise>
            <BUKRS></BUKRS>
        </xsl:otherwise>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="WERKS">
            <WERKS><xsl:value-of se-
lect="WERKS"/></WERKS>
        </xsl:when>
        <xsl:otherwise>
            <WERKS></WERKS>
        </xsl:otherwise>
    </xsl:choose>

    <xsl:choose>

```

```

<xsl:when test="SYSID">
    <SYSID><xsl:value-of select="SYSID"/></SYSID>
</xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="RUECK">
        <RUECK><xsl:value-of se-
lect="RUECK"/></RUECK>
    </xsl:when>
    <xsl:otherwise>
        <RUECK></RUECK>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="AUFNR">
        <AUFNR><xsl:value-of se-
lect="AUFNR"/></AUFNR>
    </xsl:when>
    <xsl:otherwise>
        <AUFNR></AUFNR>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="VORNR">
        <VORNR><xsl:value-of se-
lect="VORNR"/></VORNR>
    </xsl:when>
    <xsl:otherwise>
        <VORNR></VORNR>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="ASPLT">
        <ASPLT><xsl:value-of se-
lect="ASPLT"/></ASPLT>
    </xsl:when>
    <xsl:otherwise>
        <ASPLT></ASPLT>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="ARBPL">
        <ARBPL><xsl:value-of select="ARBPL"/></AR-
BPL>
    </xsl:when>
    <xsl:otherwise>

```

```

        <ARBPL></ARBPL>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="DATUM">
        <DATUM><xsl:value-of select="DA-
TUM"/></DATUM>
    </xsl:when>
    <xsl:otherwise>
        <DATUM></DATUM>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="EZEIT">
        <EZEIT><xsl:value-of se-
lect="EZEIT"/></EZEIT>
    </xsl:when>
    <xsl:otherwise>
        <EZEIT></EZEIT>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="SCHIT">
        <SCHIT><xsl:value-of se-
lect="SCHIT"/></SCHIT>
    </xsl:when>
    <xsl:otherwise>
        <SCHIT></SCHIT>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="SCHIK">
        <SCHIK><xsl:value-of se-
lect="SCHIK"/></SCHIK>
    </xsl:when>
    <xsl:otherwise>
        <SCHIK></SCHIK>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="KOSMA">
        <KOSMA><xsl:value-of se-
lect="KOSMA"/></KOSMA>
    </xsl:when>
    <xsl:otherwise>
        <KOSMA></KOSMA>
    </xsl:otherwise>
</xsl:choose>

```

```

<xsl:choose>
    <xsl:when test="PZEIT">
        <PZEIT><xsl:value-of se-
lect="PZEIT"/></PZEIT>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="INTNR">
        <INTNR><xsl:value-of se-
lect="INTNR"/></INTNR>
    </xsl:when>
    <xsl:otherwise>
        <INTNR></INTNR>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="AGRUN">
        <AGRUN><xsl:value-of se-
lect="AGRUN"/></AGRUN>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="AVERU">
        <AVERU><xsl:value-of se-
lect="AVERU"/></AVERU>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="NGRUN">
        <NGRUN><xsl:value-of se-
lect="NGRUN"/></NGRUN>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="NVERU">
        <NVERU><xsl:value-of se-
lect="NVERU"/></NVERU>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="COLAR">
        <COLAR><xsl:value-of select="CO-
LAR"/></COLAR>
    </xsl:when>
</xsl:choose>

```

```

<xsl:choose>
    <xsl:when test="PEOBJ">
        <PEOBJ><xsl:value-of se-
lect="PEOBJ"/></PEOBJ>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="AUSTA">
        <AUSTA><xsl:value-of se-
lect="AUSTA"/></AUSTA>
    </xsl:when>
    <xsl:otherwise>
        <AUSTA></AUSTA>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="MASTA">
        <MASTA><xsl:value-of se-
lect="MASTA"/></MASTA>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="PERNR">
        <PERNR><xsl:value-of se-
lect="PERNR"/></PERNR>
    </xsl:when>
    <xsl:otherwise>
        <PERNR></PERNR>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="ZAUSW">
        <ZAUSW><xsl:value-of se-
lect="ZAUSW"/></ZAUSW>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="PROPZ">
        <PROPZ><xsl:value-of se-
lect="PROPZ"/></PROPZ>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="PAUSE">

```

```

        <PAUSE><xsl:value-of se-
lect="PAUSE"/></PAUSE>
        </xsl:when>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="LSTAR">
            <LSTAR><xsl:value-of se-
lect="LSTAR"/></LSTAR>
        </xsl:when>
        <xsl:otherwise>
            <LSTAR></LSTAR>
        </xsl:otherwise>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="SOLLS">
            <SOLLS><xsl:value-of se-
lect="SOLLS"/></SOLLS>
        </xsl:when>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="SOLLE">
            <SOLLE><xsl:value-of se-
lect="SOLLE"/></SOLLE>
        </xsl:when>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="LOARR">
            <LOARR><xsl:value-of se-
lect="LOARR"/></LOARR>
        </xsl:when>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="DAUER">
            <DAUER><xsl:value-of se-
lect="DAUER"/></DAUER>
        </xsl:when>
        <xsl:otherwise>
            <DAUER></DAUER>
        </xsl:otherwise>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="ZFS1">
            <ZFS1><xsl:value-of se-
lect="ZFS1"/></ZFS1>
        </xsl:when>
    
```

```

<xsl:otherwise>
    <ZFSL1></ZFSL1>
</xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="LMNGA">
        <LMNGA><xsl:value-of se-
lect="LMNGA"/></LMNGA>
    </xsl:when>
    <xsl:otherwise>
        <LMNGA></LMNGA>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="XMNGA">
        <XMNGA><xsl:value-of se-
lect="XMNGA"/></XMNGA>
    </xsl:when>
    <xsl:otherwise>
        <XMNGA></XMNGA>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="RMNGA">
        <RMNGA><xsl:value-of se-
lect="RMNGA"/></RMNGA>
    </xsl:when>
    <xsl:otherwise>
        <RMNGA></RMNGA>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="MEINH">
        <MEINH><xsl:value-of se-
lect="MEINH"/></MEINH>
    </xsl:when>
    <xsl:otherwise>
        <MEINH></MEINH>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="MATNR">
        <MATNR><xsl:value-of se-
lect="MATNR"/></MATNR>
    </xsl:when>
    <xsl:otherwise>

```

```

        <MATNR></MATNR>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="KOSTL">
        <KOSTL><xsl:value-of se-
lect="KOSTL"/></KOSTL>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="ZTEAM">
        <ZTEAM><xsl:value-of se-
lect="ZTEAM"/></ZTEAM>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="ZFLAG">
        <ZFLAG><xsl:value-of se-
lect="ZFLAG"/></ZFLAG>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="KOTYP">
        <KOTYP><xsl:value-of select="KOTYP"/></KO-
TYP>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="KOOBJ">
        <KOOBJ><xsl:value-of se-
lect="KOOBJ"/></KOOBJ>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="APTYP">
        <APTRYP><xsl:value-of select="APTRYP"/></AP-
TYP>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="VGWMZ">
        <VGWMZ><xsl:value-of se-
lect="VGWMZ"/></VGWMZ>
    </xsl:when>
</xsl:choose>

```

```

        </xsl:choose>

        <xsl:choose>
            <xsl:when test="VGWRZ">
                <VGWRZ><xsl:value-of se-
lect="VGWRZ"/></VGWRZ>
            </xsl:when>
        </xsl:choose>

        <xsl:choose>
            <xsl:when test="KORID">
                <KORID><xsl:value-of se-
lect="KORID"/></KORID>
            </xsl:when>
            <xsl:otherwise>
                <KORID></KORID>
            </xsl:otherwise>
        </xsl:choose>

        <xsl:choose>
            <xsl:when test="KORTY">
                <KORTY><xsl:value-of se-
lect="KORTY"/></KORTY>
            </xsl:when>
            <xsl:otherwise>
                <KORTY></KORTY>
            </xsl:otherwise>
        </xsl:choose>

        <xsl:choose>
            <xsl:when test="KORDT">
                <KORDT><xsl:value-of se-
lect="KORDT"/></KORDT>
            </xsl:when>
            <xsl:otherwise>
                <KORDT></KORDT>
            </xsl:otherwise>
        </xsl:choose>

        <xsl:choose>
            <xsl:when test="KORZT">
                <KORZT><xsl:value-of se-
lect="KORZT"/></KORZT>
            </xsl:when>
            <xsl:otherwise>
                <KORZT></KORZT>
            </xsl:otherwise>
        </xsl:choose>

        <xsl:choose>
            <xsl:when test="KORSA">

```

```

        <KORSA><xsl:value-of se-
lect="KORSA"/></KORSA>
        </xsl:when>
        <xsl:otherwise>
            <KORSA></KORSA>
        </xsl:otherwise>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="TSTMS">
            <TSTMS><xsl:value-of se-
lect="TSTMS"/></TSTMS>
            </xsl:when>
        </xsl:choose>

        <xsl:choose>
            <xsl:when test="FAK01">
                <FAK01><xsl:value-of se-
lect="FAK01"/></FAK01>
                </xsl:when>
            </xsl:choose>

            <xsl:choose>
                <xsl:when test="FAK02">
                    <FAK02><xsl:value-of se-
lect="FAK02"/></FAK02>
                    </xsl:when>
                </xsl:choose>

                <xsl:choose>
                    <xsl:when test="FAK03">
                        <FAK03><xsl:value-of se-
lect="FAK03"/></FAK03>
                        </xsl:when>
                    </xsl:choose>

                    <xsl:choose>
                        <xsl:when test="ZUMIN">
                            <ZUMIN><xsl:value-of se-
lect="ZUMIN"/></ZUMIN>
                            </xsl:when>
                        </xsl:choose>

                        <xsl:choose>
                            <xsl:when test="ZZFSL">
                                <ZZFSL><xsl:value-of se-
lect="ZZFSL"/></ZZFSL>
                                </xsl:when>
                            </xsl:choose>

                            <xsl:choose>

```

```

<xsl:when test="ZZBWK">
    <ZZBWK><xsl:value-of se-
lect="ZZBWK"/></ZZBWK>
        </xsl:when>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="ZBKS">
            <ZBKS><xsl:value-of se-
lect="ZBKS"/></ZBKS>
                </xsl:when>
            </xsl:choose>

            <xsl:choose>
                <xsl:when test="ZBAP">
                    <ZBAP><xsl:value-of se-
lect="ZBAP"/></ZBAP>
                        </xsl:when>
                    </xsl:choose>

                    <xsl:choose>
                        <xsl:when test="KOMPO">
                            <KOMPO><xsl:value-of se-
lect="KOMPO"/></KOMPO>
                                </xsl:when>
                            </xsl:choose>

                            <xsl:choose>
                                <xsl:when test="TERID">
                                    <TERID><xsl:value-of se-
lect="TERID"/></TERID>
                                        </xsl:when>
                                    </xsl:choose>

                                    <xsl:choose>
                                        <xsl:when test="TEXT1">
                                            <TEXT1><xsl:value-of se-
lect="TEXT1"/></TEXT1>
                                                </xsl:when>
                                            </xsl:choose>

                                            <xsl:choose>
                                                <xsl:when test="VERID">
                                                    <VERID><xsl:value-of se-
lect="VERID"/></VERID>
                                                        </xsl:when>
                                                    </xsl:choose>

                                                    <xsl:choose>
                                                        <xsl:when test="RSNUM">

```

```

        <RSNUM><xsl:value-of se-
lect="RSNUM"/></RSNUM>
            </xsl:when>
        </xsl:choose>

        <xsl:choose>
            <xsl:when test="RSPOS">
                <RSPOS><xsl:value-of se-
lect="RSPOS"/></RSPOS>
                    </xsl:when>
            </xsl:choose>

            <xsl:choose>
                <xsl:when test="CHARG">
                    <CHARG><xsl:value-of se-
lect="CHARG"/></CHARG>
                        </xsl:when>
                </xsl:choose>

                <xsl:choose>
                    <xsl:when test="PALET">
                        <PALET><xsl:value-of se-
lect="PALET"/></PALET>
                            </xsl:when>
                    </xsl:choose>

                    <xsl:choose>
                        <xsl:when test="LSDAT">
                            <LSDAT><xsl:value-of se-
lect="LSDAT"/></LSDAT>
                                </xsl:when>
                        </xsl:choose>

                        <xsl:choose>
                            <xsl:when test="GWCHT">
                                <GWCHT><xsl:value-of se-
lect="GWCHT"/></GWCHT>
                                    </xsl:when>
                            </xsl:choose>

                            <xsl:choose>
                                <xsl:when test="ADDITIONAL">
                                    <xsl:copy-of select="ADDITIONAL/*"/>
                                        </xsl:when>
                                </xsl:choose>
                                </_FFMES_-SRUECK>
                            </IDOC>
                        </_FFMES_-R>
                    </xsl:template>
    </xsl:stylesheet>
</xsl:when>

```

```

        </xsl:choose>

        <xsl:choose>
            <xsl:when test="GWCHT">
                <GWCHT><xsl:value-of se-
lect="GWCHT"/></GWCHT>
            </xsl:when>
        </xsl:choose>

        <xsl:choose>
            <xsl:when test="ADDITIONAL">
                <xsl:copy-of select="ADDITIONAL/*"/>
            </xsl:when>
        </xsl:choose>
        </_-FFMES_-SRUECK>
    </IDOC>
</_-FFMES_-R>
</xsl:template>
</xsl:stylesheet>
<xsl:when test="PALET">
    <PALET><xsl:value-of se-
lect="PALET"/></PALET>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="LSDAT">
        <LSDAT><xsl:value-of se-
lect="LSDAT"/></LSDAT>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="GWCHT">
        <GWCHT><xsl:value-of se-
lect="GWCHT"/></GWCHT>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="ADDITIONAL">
        <xsl:copy-of select="ADDITIONAL/*"/>
    </xsl:when>
</xsl:choose>
        </_-FFMES_-SRUECK>
    </IDOC>
</_-FFMES_-R>
</xsl:template>
</xsl:stylesheet>

```

4.2 Download XSLT Mapper

4.2.1 Orders

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="com.forcam.erp.download.common" xmlns:ord="com.forcam.erp.download.order" version="1.0">

    <!-- A XML output is generated. -->
    <xsl:output method="xml" indent="yes" />

    <!-- Remove all spaces (trim) in all elements. -->
    <xsl:strip-space elements="*" />

    <!-- Start point of processing the IDOC-XML. -->
    <xsl:template match="__FFMES__F">
        <xsl:apply-templates/>
    </xsl:template>

    <xsl:template match="IDOC">
        <xsl:apply-templates/>
    </xsl:template>

    <!-- Discard all data of the EDI_DC40 tag. -->
    <xsl:template match="EDI_DC40"/>

    <!-- Discard all data of the __FFMES__SVART tag. -->
    <xsl:template match="__FFMES__SVSART"/>

    <!-- Discard all data of the __FFMES__CAFOLG tag. -->
    <xsl:template match="__FFMES__CAFOLG"/>

    <!-- Prevent that the content of all XML tags is inserted under the operations tag. -->
    <xsl:template match="*"/>

    <!-- Template for the processing of the operation texts. -->
    <xsl:template match="__FFMES__SAFOTX">
        <xsl:value-of select="AFOTXT"/><xsl:value-of select="
" />
    </xsl:template>

    <!-- Variable that contains the language code from the field REGIO. If this field isn't available,
then de-DE is used as default language. -->
    <xsl:variable name="languageCode">
        <xsl:choose>
            <xsl:when test="__FFMES__F/IDOC/__FFMES__SAUFTR/REGIO">
                <xsl:value-of select="__FFMES__F/IDOC/__FFMES__SAUFTR/RE-
GIO"/>
            </xsl:when>
            <xsl:otherwise>de-DE</xsl:otherwise>
        </xsl:choose>
    </xsl:variable>

```

```

</xsl:variable>

    
<xsl:template name="editTimeUnits">
    <xsl:param name="originalTimeUnit"/>
    <xsl:choose>
        <xsl:when test="$originalTimeUnit='SEK' or $originalTimeU-
nit='SEC'">
            <xsl:text>S</xsl:text>
        </xsl:when>
        <xsl:otherwise>
            <xsl:value-of select="$originalTimeUnit"/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

    
<xsl:template name="removeLeadingZeros">
    <xsl:param name="originalString"/>
    <xsl:choose>
        <xsl:when test="starts-with($originalString,'0')">
            <xsl:call-template name="removeLeadingZeros">
                <xsl:with-param name="originalString">
                    <xsl:value-of select="substring-after($originalString,'0')"/>
                </xsl:with-param>
            </xsl:call-template>
        </xsl:when>
        <xsl:otherwise>
            <xsl:value-of select="$originalString"/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

    
<xsl:template name="transformDateToTimestamp">
    <xsl:param name="originalDate"/>
    <xsl:param name="originalTime" select="\"00:00\"/>
    <xsl:param name="timezone"/>

    <xsl:choose>
        <!-- en-US => mm/dd/yyyy -->
        <xsl:when test="$languageCode='en-US'">
            <xsl:variable name="year" select="substring($originalDate, 7, 4)"/>
            <xsl:variable name="month" select="substring($originalDate, 1, 2)"/>
            <xsl:variable name="day" select="substring($originalDate, 4, 2)"/>
                <!-- Concat all values to the timestamp with 00:00:00 as time. -->
                <xsl:value-of select="concat($year, '-', $month, '-', $day, 'T', $original-
Time, ':00', $timezone)"/>
        </xsl:when>
    </xsl:choose>

```

```

<!-- hu-HU => yyyy.mm.dd, jp-JP => yyyy/mm/dd -->
<xsl:when test="$languageCode='hu-HU' or $languageCode='jp-JP'">
    <xsl:variable name="year" select="substring($originalDate, 1, 4)"/>
    <xsl:variable name="month" select="substring($originalDate, 6, 2)"/>
    <xsl:variable name="day" select="substring($originalDate, 9, 2)"/>
        <!-- Concat all values to the timestamp with 00:00:00 as time. -->
        <xsl:value-of select="concat($year, '-', $month, '-', $day, 'T', $original-
Time, ':00', $timezone)"/>
    </xsl:when>
    <!-- dd.mm.yyyy, dd/mm/yyyy => de-DE, en-GB, zh-CN, fr-FR, es-ES, bg-BG, hr-HR,
cs-CZ, da-DK, ro-RO, nl-NL, hi-IN, it-IT, pl-PL, pt-PT, ru-RU, sv-SE, tr-TR and br-BR -->
    <xsl:otherwise>
        <xsl:variable name="year" select="substring($originalDate, 7, 4)"/>
        <xsl:variable name="month" select="substring($originalDate, 4, 2)"/>
        <xsl:variable name="day" select="substring($originalDate, 1, 2)"/>
            <!-- Concat all values to the timestamp with 00:00:00 as time. -->
            <xsl:value-of select="concat($year, '-', $month, '-', $day, 'T', $original-
Time, ':00', $timezone)"/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

<!--
    Transform number delimiter . and , to the de-DE format. This means , as delimiter
and no thousand delimiter.
    en-US, en-GB: 1,000.56 => 1000,56
-->
<xsl:template name="transformNumberDelimiters">
    <xsl:param name="number"/>

    <xsl:choose>
        <xsl:when test="$languageCode='en-US' or $languageCode='en-GB' or $lan-
guageCode='hi-IN' or $languageCode='jp-JP'>
            <xsl:value-of select="translate($number, ',', ',')"/>
        </xsl:when>
        <!-- de-DE, zh-CN, fr-FR, es-ES, bg-BG, hr-HR, cs-CZ, da-DK, ro-RO, nl-NL, hu-
HU, it-IT, pl-PL, pt-PT, ru-RU, sv-SE, tr-TR und br-BR. -->
        <xsl:otherwise>
            <xsl:value-of select="translate(translate($number, '.', ','), ',', ',')"/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

<!-- Template for the order. -->
<xsl:template match="____FFMES____SAUFTR">

    <xsl:choose>
        <!-- If an order is deleted, then only the order id and the erp key is important.
-->
        <xsl:when test="VAKZ = 'D'">
            <xsl:variable name="ORDER_NUMBER">

```

```

        <xsl:call-template name="removeLeadingZeros">
            <xsl:with-param name="originalString" se-
lect="AUFNR"/>
        </xsl:call-template>
    </xsl:variable>

<ord:OrderType xmlns="com.forcam.erp.download.common" xmlns:ord="com.for-
cam.erp.download.order" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLo-
cation="com.forcam.erp.download.order ERPDownloadOrder.xsd" action="DELETE">
    <ord:OrderERPKey>
        <Client><xsl:value-of select="MANDT"/></Client>
        <CompanyCode><xsl:value-of select="BUKRS"/></CompanyCode>
        <Plant><xsl:value-of select="WERKS"/></Plant>

        <xsl:if test="SYSID">
            <SystemId><xsl:value-of select="SYSID"/></SystemId>
        </xsl:if>
    </ord:OrderERPKey>
    <ord:OrderNumber><xsl:value-of select="$ORDER_NUMBER"/></ord:OrderNumber>
    <ord:OrderSplit><xsl:value-of select="ASPLT"/></ord:OrderSplit>
        <ord:ERPStatusIds><xsl:value-of select="STATU_ID"/></ord:ERPStatusIds>
        <ord:ERPStatusCodes><xsl:value-of select="STATU"/></ord:ERPStatusCodes>
        <ord:ERPStatusCodesTranslation><xsl:value-of select="STATU_LA"/></ord:ERPStatus-
CodesTranslation>
    </ord:OrderType>
    </xsl:when>

    <!-- Otherwise a new order is created or an existing order is updated. -->
    <xsl:otherwise>
        <xsl:variable name="ORDER_ACTION">
            <xsl:choose>
                <!-- A NOP is set -> this means, that the order itself is not to
be processed only the childs -> operations. -->
                <xsl:when test="VAKZ = 'X'">NOP</xsl:when>
                <!-- Order has to be processed. -->
                <xsl:otherwise>ADD</xsl:otherwise>
            </xsl:choose>
        </xsl:variable>

        <!-- Create a variable with the order number with no leading zeros. -->
    >
        <xsl:variable name="ORDER_NUMBER">
            <xsl:call-template name="removeLeadingZeros">
                <xsl:with-param name="originalString" se-
lect="AUFNR"/>
            </xsl:call-template>
        </xsl:variable>

        <ord:OrderType xmlns="com.forcam.erp.download.common" xmlns:ord="com.for-
cam.erp.download.order" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLo-
cation="com.forcam.erp.download.order ERPDownloadOrder.xsd" action="{{$ORDER_ACTION}}">

```

```

<ord:OrderERPKey>
<Client><xsl:value-of select="MANDT"/></Client>
<CompanyCode><xsl:value-of select="BUKRS"/></CompanyCode>
<Plant><xsl:value-of select="WERKS"/></Plant>

<xsl:if test="SYSID">
    <SystemId><xsl:value-of select="SYSID"/></SystemId>
</xsl:if>
</ord:OrderERPKey>
    <ord:OrderNumber><xsl:value-of select="$ORDER_NUMBER"/></ord:OrderNumber>
    <ord:OrderSplit><xsl:value-of select="ASPLT"/></ord:OrderSplit>

        <ord:ERPStatusIds><xsl:value-of select="STATU_ID"/></ord:ERPStatusIds>
<ord:ERPStatusCodes><xsl:value-of select="STATU"/></ord:ERPStatusCodes>
<ord:ERPStatusCodesTranslation><xsl:value-of select="STATU_LA"/></ord:ERPStatus-
CodesTranslation>

<xsl:if test="APRIO">
    <ord:OrderPriority><xsl:value-of se-
lect="APRIO"/></ord:OrderPriority>
    </xsl:if>

    <!-- Create release timestamp in format dd.MM.yyyyT00:00:00 -->
    <ord:ReleaseDateTime>
        <xsl:call-template name="transformDate-
ToTimestamp">
            <xsl:with-param name="originalDate" se-
lect="FRDAT"/>
        </xsl:call-template>
    </ord:ReleaseDateTime>

    <!-- Create target start timestamp in format dd.MM.yyyyT00:00:00 -->
    <ord:TargetStartDateTime>
        <xsl:call-template name="transformDate-
ToTimestamp">
            <xsl:with-param name="originalDate" se-
lect="SSTRT"/>
        </xsl:call-template>
    </ord:TargetStartDateTime>

    <!-- Create target end timestamp in format dd.MM.yyyyT00:00:00 -->
    <ord:TargetEndDateTime>
        <xsl:call-template name="transformDate-
ToTimestamp">
            <xsl:with-param name="originalDate" se-
lect="SLIEF"/>
        </xsl:call-template>
    </ord:TargetEndDateTime>

        <ord:TargetQuantity>

```

```

        <xsl:call-template name="transformNumberDelime-
ters">
            <xsl:with-param name="number" se-
lect="AFANZ"/>
        </xsl:call-template>
    </ord:TargetQuantity>

    <!-- If no quantity unit is available, the unit piece (stk) is set. -->
    <xsl:choose>
        <xsl:when test="AMEIN">
            <ord:DisplayQuantityUnit><xsl:value-of se-
lect="AMEIN"/></ord:DisplayQuantityUnit>
        </xsl:when>
        <xsl:otherwise>
            <ord:DisplayQuantityUnit>stk</ord:Dis-
playQuantityUnit>
        </xsl:otherwise>
    </xsl:choose>

    <xsl:if test="VERID">
        <ord:ProductionVersion><xsl:value-of select="VERID"/></ord:ProductionVersion>
    </xsl:if>

    <xsl:if test="VAGRP">
        <ord:PlannerGroupInCharge><xsl:value-of select="VAGRP"/></ord:Planner-
GroupInCharge>
    </xsl:if>

    <ord:Description>
        <Translation>
            <Language><xsl:value-of select="$languageCode"/></Language>
            <Text><xsl:value-of select="ATEXT"/></Text>
        </Translation>
    </ord:Description>

    <!-- FLS specific fields for the order. -->
    <xsl:if test="GSTRS">
        <!-- Create scheduled start timestamp in format yyyy-MM-ddT00:00:00 -->
        <ord:ScheduledStartDate>
            <xsl:call-template name="transformDateToTimestamp">
                <xsl:with-param name="originalDate" select="GSTRS"/>
            </xsl:call-template>
        </ord:ScheduledStartDate>
    </xsl:if>
    <xsl:if test="GLTRS">
        <!-- Create scheduled end timestamp in format yyyy-MM-ddT00:00:00 -->
        <ord:ScheduledEndDate>
            <xsl:call-template name="transformDateToTimestamp">
                <xsl:with-param name="originalDate" select="GLTRS"/>
            </xsl:call-template>
    </xsl:if>

```

```

        </ord:ScheduledEndDate>
    </xsl:if>

        <!-- If characteristics of materials are available, then add
these characteristics to the ERP-XML. -->
        <xsl:if test="____FFMES____SAUFMK">
            <!-- Template for processing the characteristics of
materials. -->
            <ord:MaterialCharacteristics>
                <xsl:for-each se-
lect="____FFMES____SAUFMK">
                    <ord:MaterialCharacteristic>
                        <ord:orderItem><xsl:value-of select="POSNR"/></ord:orderItem>
                        <ord:classType><xsl:value-of select="CLASS"/></ord:classType>
                        <ord:characteristicName><xsl:value-of select="ATNAM"/></ord:characteristic-
Name>
                        <ord:characteristicValue1><xsl:value-of select="ATWRT"/></ord:characteris-
ticValue1>
                        <ord:characteristicValue2><xsl:value-of select="ATW02"/></ord:characteris-
ticValue2>
                        <ord:characteristicValue3><xsl:value-of select="ATW03"/></ord:characteris-
ticValue3>
                        <ord:characteristicValue4><xsl:value-of select="ATW04"/></ord:characteris-
ticValue4>
                        <ord:characteristicValue5><xsl:value-of select="ATW05"/></ord:characteris-
ticValue5>
                        <ord:characteristicValue6><xsl:value-of select="ATW06"/></ord:characteris-
ticValue6>
                        <ord:characteristicValue7><xsl:value-of select="ATW07"/></ord:characteris-
ticValue7>
                        <ord:characteristicValue8><xsl:value-of select="ATW08"/></ord:characteris-
ticValue8>
                        <ord:characteristicValue9><xsl:value-of select="ATW09"/></ord:characteris-
ticValue9>
                        <ord:characteristicValue10><xsl:value-of select="ATW10"/></ord:characteris-
ticValue10>
                        <ord:characteristicValue11><xsl:value-of select="ATW11"/></ord:characteris-
ticValue11>
                        <ord:characteristicValue12><xsl:value-of select="ATW12"/></ord:characteris-
ticValue12>
                        <ord:characteristicValue13><xsl:value-of select="ATW13"/></ord:characteris-
ticValue13>
                    </ord:MaterialCharacteristic>
                </xsl:for-each>
            </ord:MaterialCharacteristics>
        </xsl:if>

        <!-- Template for processing customer specific Order userFields -->
        <!-- processing disabled in default xsl of order download
        <xsl:if test="orderUserFields">
            <xsl:call-template name="processOrderUserFields"/>

```

```

</xsl:if>
-->

<ord:Operations>
    <xsl:call-template name="operations"/>
</ord:Operations>

<ord:Material>
    <Description>
        <Translation>
            <Language><xsl:value-of select="$languageCode"/></Language>
            <Text><xsl:value-of select="MTEXT"/></Text>
        </Translation>
    </Description>
    <Number><xsl:value-of select="MATNR"/></Number>
</ord:Material>
</ord:OrderType>
    </xsl:otherwise>
</xsl:choose>
</xsl:template>

<!--
Template for processing customer specific ORDER userFields.
Delete all unused UFxx fields and replace the '...' of the used fields with the customer specific
value.
50 userfields from UF01 - UF50 are available (only the first then userfields are in the sample be-
low).
-->
<!--
Please enter the name of the xml node containing the order user fields here.
<xsl:template name="processOrderUserFields">
    <xsl:for-each select="orderUserFields">
        <ord:UserDataFields>
            <xsl:if test="...">
                <UF1><xsl:value-of select="..."/></UF1>
            </xsl:if>
            <xsl:if test="...">
                <UF2><xsl:value-of select="..."/></UF2>
            </xsl:if>
            <xsl:if test="...">
                <UF3><xsl:value-of select="..."/></UF3>
            </xsl:if>
            <xsl:if test="...">
                <UF4><xsl:value-of select="..."/></UF4>
            </xsl:if>
            <xsl:if test="...">
                <UF5><xsl:value-of select="..."/></UF5>
            </xsl:if>
            <xsl:if test="...">
                <UF6><xsl:value-of select="..."/></UF6>
            </xsl:if>
        </ord:UserDataFields>
    </xsl:for-each>
</xsl:template>

```

```

<xsl:if test="...">
    <UF7><xsl:value-of select="..."/></UF7>
</xsl:if>
<xsl:if test="...">
    <UF8><xsl:value-of select="..."/></UF8>
</xsl:if>
<xsl:if test="...">
    <UF9><xsl:value-of select="..."/></UF9>
</xsl:if>
<xsl:if test="...">
    <UF10><xsl:value-of select="..."/></UF10>
</xsl:if>
</ord:UserDataFields>
</xsl:for-each>
</xsl:template>
-->

<!-- Template that add all operations with the corresponding action to the ERP-XML. -->
<xsl:template name="operations">
    <xsl:if test="__FFMES__SAFOLG[VAKZ != 'D']">
        <xsl:call-template name="addOperations"/>
    </xsl:if>

    <xsl:if test="__FFMES__SAFOLG[VAKZ = 'D']">
        <xsl:call-template name="deleteOperations"/>
    </xsl:if>
</xsl:template>

<!-- Template for adding all operations to the ERP-XML (add, update and nop operation). -->
<xsl:template name="addOperations">
    <xsl:for-each select="__FFMES__SAFOLG[VAKZ != 'D']">

        <xsl:variable name="OPERATION_ACTION">
            <xsl:choose>
                <!-- A NOP is set -> this means, that the operations itself is
not to be processed only the childs -> operations components and/or production resource tools. -->
                <xsl:when test="VAKZ = 'X'">NOP</xsl:when>
                <!-- Operation has to be processed. -->
                <xsl:otherwise>ADD</xsl:otherwise>
            </xsl:choose>
        </xsl:variable>

        <!-- Create ERP-XML for the operation. -->
        <ord:Operation action="${OPERATION_ACTION}">
            <ord:OrderERPKey>
                <Client><xsl:value-of select="MANDT"/></Client>
                <CompanyCode><xsl:value-of select="BUKRS"/></CompanyCode>
                <Plant><xsl:value-of select="WERKS"/></Plant>

                <xsl:if test="SYSID">
                    <SystemId><xsl:value-of select="SYSID"/></SystemId>

```

```

        </xsl:if>
    </ord:OrderERPKey>
    <ord:OperationNumber><xsl:value-of select="VORNR"/></ord:OperationNumber>
    <ord:OperationSplit><xsl:value-of select="VSPLT"/></ord:OperationSplit>

        <ord:ConfirmationNumber><xsl:value-of select="RUECK"/></ord:ConfirmationNumber>

        <ord:ERPYieldQuantity>
            <xsl:call-template name="transformNumberDelimeters">
                <xsl:with-param name="number" select="LMNGA"/>
            </xsl:call-template>
        </ord:ERPYieldQuantity>

        <ord:ERPReworkQuantity>
            <xsl:call-template name="transformNumberDelimeters">
                <xsl:with-param name="number" select="RMNGA"/>
            </xsl:call-template>
        </ord:ERPReworkQuantity>

        <ord:ERPScrapQuantity>
            <xsl:call-template name="transformNumberDelimeters">
                <xsl:with-param name="number" select="XMNGA"/>
            </xsl:call-template>
        </ord:ERPScrapQuantity>

            <ord:ERPPlannedScrapQuantity><xsl:value-of select="AUSSS"/></ord:ERPPlannedScrapQuantity>
            <ord:ERPStatusIds><xsl:value-of select="STATU_ID"/></ord:ERPStatusIds>
            <ord:ERPStatusCodes><xsl:value-of select="STATU"/></ord:ERPStatusCodes>
            <ord:ERPStatusCodesTranslation><xsl:value-of select="STATU_LA"/></ord:ERPStatusCodesTranslation>

        <ord:Description>
            <Translation>
                <Language><xsl:value-of select="$languageCode"/></Language>
                <Text><xsl:value-of select="LTXA1"/></Text>
            </Translation>
        </ord:Description>

        <!-- If operation texts are available, then add the texts to the ERP-->
    XML. -->
        <xsl:if test="____FFMES____SAFOTX">
            <!-- This variable contains all operation texts in one string. -->
            <xsl:variable name="concatenatedText">
                <xsl:apply-templates select="____FFMES____SAFOTX"/>
            </xsl:variable>
            <ord:OperationText>

```

```

<Translation>
    <Language><xsl:value-of select="$language-
Code"/></Language>
    <Text><xsl:value-of select="$concatenat-
edText"/></Text>
</Translation>
</ord:OperationText>
</xsl:if>

<ord:ControlKey><xsl:value-of select="STEUS"/></ord:ControlKey>
<ord:OrderType><xsl:value-of select="AUART"/></ord:OrderType>
<ord:FunctionType>1</ord:FunctionType>
<ord:LeadingOperation>true</ord:LeadingOperation>

<xsl:if test="PLNFL">
    <ord:OperationSequence><xsl:value-of select="PLNFL"/></ord:OperationSequence>
</xsl:if>

<ord:TargetWorkplace><xsl:value-of select="ARBPLI"/></ord:TargetWorkplace>

<xsl:if test="ARBPLG">
    <ord:WorkplaceGroup><xsl:value-of select="ARBPLG"/></ord:WorkplaceGroup>
</xsl:if>

<!-- Create target start timestamp in format yyyy-MM-ddTHH:mm:00 -->
<xsl:choose>
    <xsl:when test="__FFMES__SOPPLD/SSAVD">
        <ord:TargetStartDateTime>
            <xsl:call-template name="transformDate-
ToTimestamp">
                <xsl:with-param name="originalDate"-
select=__FFMES__SOPPLD/SSAVD"/>
                <xsl:with-param name="originalTime"-
select=__FFMES__SOPPLD/SSAVZ"/>
                <xsl:with-param name="timezone"-
select="UTCOFF"/>
                </xsl:call-template>
            </ord:TargetStartDateTime>
        </xsl:when>
        <xsl:otherwise>
            <ord:TargetStartDateTime>
                <xsl:call-template name="transformDate-
ToTimestamp">
                    <xsl:with-param name="originalDate"-
select="SSAVD"/>
                    <xsl:with-param name="originalTime"-
select="SSAVZ"/>
                    <xsl:with-param name="timezone"-
select="UTCOFF"/>
                    </xsl:call-template>
            </ord:TargetStartDateTime>
        </xsl:otherwise>
    </xsl:when>
</xsl:choose>

```

```

        </xsl:otherwise>
    </xsl:choose>

    <!-- Create target end timestamp in format yyyy-MM-ddTHH:mm:00 -->
    <xsl:choose>
        <xsl:when test="__FFMES__SOPPLD/SSEDD">
            <ord:TargetEndDateTime>
                <xsl:call-template name="transformDate-
ToTimestamp">
                    <xsl:with-param name="originalDate" />
                    <xsl:with-param name="originalTime" />
                    <xsl:with-param name="timezone" />
                </xsl:call-template>
            </ord:TargetEndDateTime>
        </xsl:when>
        <xsl:otherwise>
            <ord:TargetEndDateTime>
                <xsl:call-template name="transformDate-
ToTimestamp">
                    <xsl:with-param name="originalDate" />
                    <xsl:with-param name="originalTime" />
                    <xsl:with-param name="timezone" />
                </xsl:call-template>
            </ord:TargetEndDateTime>
        </xsl:otherwise>
    </xsl:choose>

    <ord:DefaultStrokeFactor>1</ord:DefaultStrokeFactor>
    <ord:DefaultPieceTimeFactor>
        <xsl:call-template name="transformNumberDelimeters">
            <xsl:with-param name="number" select="BMSCH"/>
        </xsl:call-template>
    </ord:DefaultPieceTimeFactor>

    <!-- If no quantity unit is available, the unit piece (stk) is set. -->
    <xsl:choose>
        <xsl:when test="MEINH">
            <ord:DisplayQuantityUnit><xsl:value-of se-
lect="MEINH"/></ord:DisplayQuantityUnit>
            <xsl:when>
                <xsl:otherwise>
                    <ord:DisplayQuantityUnit>stk</ord:DisplayQuant-
ityUnit>
                </xsl:otherwise>
            </xsl:when>
        </xsl:otherwise>
    </xsl:choose>

```

```

        </xsl:choose>

<ord:TargetQuantity>
    <xsl:call-template name="transformNumberDelimeters">
        <xsl:with-param name="number" select="MGVRG"/>
    </xsl:call-template>
</ord:TargetQuantity>

<xsl:choose>
    <xsl:when test="__FFMES__SOPPLD/RUEST">
        <ord:StandardValue1>
            <xsl:call-template name="transform-
NumberDelimeters">
                <xsl:with-param name="number" se-
lect=__FFMES__SOPPLD/RUEST"/>
            </xsl:call-template>
        </ord:StandardValue1>
    </xsl:when>
    <xsl:otherwise>
        <ord:StandardValue1>
            <xsl:call-template name="transform-
NumberDelimeters">
                <xsl:with-param name="number" se-
lect="VGW01"/>
            </xsl:call-template>
        </ord:StandardValue1>
    </xsl:otherwise>
</xsl:choose>

<xsl:choose>
    <xsl:when test="__FFMES__SOPPLD/RSTZE">
        <ord:StandardUnit1>
            <xsl:call-template name="editTimeU-
nits">
                <xsl:with-param
name="originalTimeUnit" select=__FFMES__SOPPLD/RSTZE"/>
            </xsl:call-template>
        </ord:StandardUnit1>
    </xsl:when>
    <xsl:otherwise>
        <ord:StandardUnit1>
            <xsl:call-template name="editTimeU-
nits">
                <xsl:with-param
name="originalTimeUnit" select="VGE01"/>
            </xsl:call-template>
        </ord:StandardUnit1>
    </xsl:otherwise>
</xsl:choose>

<ord:StandardValue2>

```

```

        <xsl:call-template name="transformNumberDelimeters">
            <xsl:with-param name="number" select="VGW02"/>
        </xsl:call-template>
    </ord:StandardValue2>

    <!-- If no piece time unit is available, then the unit min is set. -->
    <xsl:choose>
        <xsl:when test="VGE02">
            <ord:StandardUnit2><xsl:value-of select="VGE02"/></ord:StandardUnit2>
        </xsl:when>
        <xsl:otherwise>
            <ord:StandardUnit2>MIN</ord:StandardUnit2>
        </xsl:otherwise>
    </xsl:choose>

    <xsl:if test="EVORN">
        <ord:AlternateOperationNumber><xsl:value-of select="EVORN"/></ord:AlternateOperationNumber>
    </xsl:if>

    <xsl:if test="VERID">
        <ord:ProductionVersion><xsl:value-of select="VERID"/></ord:ProductionVersion>
    </xsl:if>

    <ord:CounterNumber>0</ord:CounterNumber>

    <xsl:if test="ASTTX">
        <ord:UserStatus><xsl:value-of select="ASTTX"/></ord:UserStatus>
    </xsl:if>

    <xsl:choose>
        <xsl:when test="UELIC='Y'">
            <ord:OverdeliveryCheck>true</ord:OverdeliveryCheck>
        </xsl:when>
        <xsl:otherwise>
            <ord:OverdeliveryCheck>false</ord:OverdeliveryCheck>
        </xsl:otherwise>
    </xsl:choose>

    <ord:OverdeliveryQuantity>
        <xsl:call-template name="transformNumberDelimeters">
            <xsl:with-param name="number" select="UELIT"/>
        </xsl:call-template>
    </ord:OverdeliveryQuantity>

    <xsl:choose>
        <xsl:when test="UNLIC='Y'">

```

```

        <ord:UnderdeliveryCheck>true</ord:UnderdeliveryCheck>
        </xsl:when>
        <xsl:otherwise>
            <ord:UnderdeliveryCheck>false</ord:UnderdeliveryCheck>
            </xsl:otherwise>
        </xsl:choose>

        <ord:UnderdeliveryQuantity>
            <xsl:call-template name="transformNumberDelimeters">
                <xsl:with-param name="number" select="UNLIT"/>
            </xsl:call-template>
        </ord:UnderdeliveryQuantity>

        <ord:DefaultTransportQuantity>0</ord:DefaultTransportQuantity>

        <!-- FLS specific fields for operation. -->
        <xsl:choose>
            <xsl:when test="__FFMES__SOPPLD/FSAVD">
                <!-- Create earliest start timestamp in format yyyy-MM-ddT00:00:00 -->
                <ord:EarliestStartDate>
                    <xsl:call-template name="transformDateToTimestamp">
                        <xsl:with-param name="originalDate" select="__FFMES__SOPPLD/FSAVD"/>
                        <xsl:with-param name="timezone" select="UTCOFF"/>
                    </xsl:call-template>
                </ord:EarliestStartDate>
                <xsl:when>
                    <xsl:when test="FSAVD">
                        <!-- Create earliest start timestamp in format yyyy-MM-ddT00:00:00 -->
                        <ord:EarliestStartDate>
                            <xsl:call-template name="transformDateToTimestamp">
                                <xsl:with-param name="originalDate" select="FSAVD"/>
                                <xsl:with-param name="timezone" select="UTCOFF"/>
                            </xsl:call-template>
                        </ord:EarliestStartDate>
                    </xsl:when>
                </xsl:when>
            </xsl:when>
            <xsl:when test="__FFMES__SOPPLD/SSAVD">
                <!-- Create latest start timestamp in format yyyy-MM-ddT00:00:00 -->
            </xsl:when>
        </xsl:choose>
    
```

```

<ord:LatestStartDate>
    <xsl:call-template name="transformDate-
ToTimestamp">
        <xsl:with-param name="originalDate"-
select=__FFMES__SOPPLD/SSAVD"/>
        <xsl:with-param name="timezone"-
select="UTCOFF"/>
        </xsl:call-template>
    </ord:LatestStartDate>
</xsl:when>
<xsl:when test="SSAVD">
    <!-- Create latest start timestamp in format yyyy-
MM-ddT00:00:00 -->
    <ord:LatestStartDate>
        <xsl:call-template name="transformDate-
ToTimestamp">
            <xsl:with-param name="originalDate"-
select="SSAVD"/>
            <xsl:with-param name="timezone"-
select="UTCOFF"/>
            </xsl:call-template>
        </ord:LatestStartDate>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="__FFMES__SOPPLD/FSEDD">
        <!-- Create earliest end timestamp in format yyyy-
MM-ddT00:00:00 -->
        <ord:EarliestEndDate>
            <xsl:call-template name="transformDate-
ToTimestamp">
                <xsl:with-param name="originalDate"-
select=__FFMES__SOPPLD/FSEDD"/>
                <xsl:with-param name="timezone"-
select="UTCOFF"/>
                </xsl:call-template>
            </ord:EarliestEndDate>
        </xsl:when>
        <xsl:when test="FSEDD">
            <!-- Create earliest end timestamp in format yyyy-
MM-ddT00:00:00 -->
            <ord:EarliestEndDate>
                <xsl:call-template name="transformDate-
ToTimestamp">
                    <xsl:with-param name="originalDate"-
select="FSEDD"/>
                    <xsl:with-param name="timezone"-
select="UTCOFF"/>
                    </xsl:call-template>
                </ord:EarliestEndDate>
            </xsl:when>
        </xsl:choose>
    
```

```

        </xsl:when>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="__FFMES__SOPPLD/SSEDD">
            <!-- Create latest end timestamp in format yyyy-MM-
ddT00:00:00 -->
            <ord:LatestEndDate>
                <xsl:call-template name="transformDate-
ToTimestamp">
                    <xsl:with-param name="originalDate" />
                    <xsl:with-param name="timezone" />
                </xsl:call-template>
            </ord:LatestEndDate>
        </xsl:when>
        <xsl:when test="SSEDD">
            <!-- Create latest end timestamp in format yyyy-MM-
ddT00:00:00 -->
            <ord:LatestEndDate>
                <xsl:call-template name="transformDate-
ToTimestamp">
                    <xsl:with-param name="originalDate" />
                    <xsl:with-param name="timezone" />
                </xsl:call-template>
            </ord:LatestEndDate>
        </xsl:when>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="__FFMES__SOPPLD/EPANF">
            <!-- Create scheduled start timestamp in format yyyy-
MM-ddT00:00:00 -->
            <ord:ScheduledStartDate>
                <xsl:call-template name="transformDate-
ToTimestamp">
                    <xsl:with-param name="originalDate" />
                    <xsl:with-param name="timezone" />
                </xsl:call-template>
            </ord:ScheduledStartDate>
        </xsl:when>
        <xsl:when test="EPANF">
            <!-- Create scheduled start timestamp in format yyyy-
MM-ddT00:00:00 -->
            <ord:ScheduledStartDate>

```

```

        <xsl:call-template name="transformDate-
ToTimestamp">
            <xsl:with-param name="originalDate" />
            <xsl:with-param name="timezone" />
        </xsl:call-template>
    </ord:ScheduledStartDate>
</xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="__FFMES__SOPPLD/EPEND">
        <!-- Create scheduled end timestamp in format yyyy-
MM-ddT00:00:00 -->
        <ord:ScheduledEndDate>
            <xsl:call-template name="transformDate-
ToTimestamp">
                <xsl:with-param name="originalDate" />
                <xsl:with-param name="timezone" />
            </xsl:call-template>
        </ord:ScheduledEndDate>
</xsl:when>
<xsl:when test="EPEND">
        <!-- Create scheduled end timestamp in format yyyy-
MM-ddT00:00:00 -->
        <ord:ScheduledEndDate>
            <xsl:call-template name="transformDate-
ToTimestamp">
                <xsl:with-param name="originalDate" />
                <xsl:with-param name="timezone" />
            </xsl:call-template>
        </ord:ScheduledEndDate>
</xsl:when>
</xsl:choose>

<!-- FLS specific fields. -->
<xsl:choose>
    <xsl:when test="__FFMES__SOPPLD/ZWNOR">
        <ord:TargetQueueTime>
            <xsl:call-template name="transform-
NumberDelimeters">
                <xsl:with-param name="number" se-
lect="__FFMES__SOPPLD/ZWNOR"/>
            </xsl:call-template>
        </ord:TargetQueueTime>
</xsl:when>

```

```

<xsl:when test="ZWNOR">
    <ord:TargetQueueTime>
        <xsl:call-template name="transform-
NumberDelimeters">
            <xsl:with-param name="number" se-
lect="ZWNOR"/>
        </xsl:call-template>
    </ord:TargetQueueTime>
</xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="__FFMES__SOPPLD/ZEIWN">
        <ord:TargetQueueTimeUnit><xsl:value-of se-
lect="__FFMES__SOPPLD/ZEIWN"/></ord:TargetQueueTimeUnit>
    </xsl:when>
    <xsl:when test="ZEIWN">
        <ord:TargetQueueTimeUnit><xsl:value-of se-
lect="ZEIWN"/></ord:TargetQueueTimeUnit>
    </xsl:when>
    <xsl:choose>
        <xsl:when test="__FFMES__SOPPLD/BEARZ">
            <ord:TargetProcessingTime>
                <xsl:call-template name="transform-
NumberDelimeters">
                    <xsl:with-param name="number" se-
lect="__FFMES__SOPPLD/BEARZ"/>
                </xsl:call-template>
            </ord:TargetProcessingTime>
        </xsl:when>
        <xsl:when test="BEARZ">
            <ord:TargetProcessingTime>
                <xsl:call-template name="transform-
NumberDelimeters">
                    <xsl:with-param name="number" se-
lect="BEARZ"/>
                </xsl:call-template>
            </ord:TargetProcessingTime>
        </xsl:when>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="__FFMES__SOPPLD/BEAZE">
            <ord:TargetProcessingTimeUnit><xsl:value-of se-
lect="__FFMES__SOPPLD/BEAZE"/></ord:TargetProcessingTimeUnit>
        </xsl:when>
    </xsl:choose>

```

```

        </xsl:when>
        <xsl:when test="BEAZE">
            <ord:TargetProcessingTimeUnit><xsl:value-of se-
lect="BEAZE"/></ord:TargetProcessingTimeUnit>
            </xsl:when>
        </xsl:choose>

        <xsl:choose>
            <xsl:when test="__FFMES__SOPPLD/ABRUE">
                <ord:TargetTeardownTime>
                    <xsl:call-template name="transform-
NumberDelimeters">
                        <xsl:with-param name="number" se-
lect="__FFMES__SOPPLD/ABRUE"/>
                        </xsl:call-template>
                </ord:TargetTeardownTime>
            </xsl:when>
            <xsl:when test="ABRUE">
                <ord:TargetTeardownTime>
                    <xsl:call-template name="transform-
NumberDelimeters">
                        <xsl:with-param name="number" se-
lect="ABRUE"/>
                        </xsl:call-template>
                </ord:TargetTeardownTime>
            </xsl:when>
        </xsl:choose>

        <xsl:choose>
            <xsl:when test="__FFMES__SOPPLD/ARUZE">
                <ord:TargetTeardownTimeUnit><xsl:value-of se-
lect="__FFMES__SOPPLD/ARUZE"/></ord:TargetTeardownTimeUnit>
                </xsl:when>
                <xsl:when test="ARUZE">
                    <ord:TargetTeardownTimeUnit><xsl:value-of se-
lect="ARUZE"/></ord:TargetTeardownTimeUnit>
                </xsl:when>
            </xsl:choose>

            <xsl:choose>
                <xsl:when test="__FFMES__SOPPLD/ZLMAX">
                    <ord:TargetWaitTime>
                        <xsl:call-template name="transform-
NumberDelimeters">
                            <xsl:with-param name="number" se-
lect="__FFMES__SOPPLD/ZLMAX"/>
                            </xsl:call-template>
                    </ord:TargetWaitTime>
                </xsl:when>
            </xsl:choose>

```

```

        </xsl:when>
        <xsl:when test="ZLMAX">
            <ord:TargetWaitTime>
                <xsl:call-template name="transform-
NumberDelimeters">
                    <xsl:with-param name="number" se-
lect="ZLMAX"/>
                </xsl:call-template>
            </ord:TargetWaitTime>
        </xsl:when>
    </xsl:choose>
<xsl:choose>
    <xsl:when test="__FFMES__SOPPLD/ZEILM">
        <ord:TargetWaitTimeUnit><xsl:value-of se-
lect="__FFMES__SOPPLD/ZEILM"/></ord:TargetWaitTimeUnit>
        <xsl:when>
            <xsl:when test="ZEILM">
                <ord:TargetWaitTimeUnit><xsl:value-of se-
lect="ZEILM"/></ord:TargetWaitTimeUnit>
            </xsl:when>
        </xsl:choose>

        <xsl:choose>
            <xsl:when test="__FFMES__SOPPLD/ZTNOR">
                <ord:TargetMoveTime>
                    <xsl:call-template name="transform-
NumberDelimeters">
                        <xsl:with-param name="number" se-
lect="__FFMES__SOPPLD/ZTNOR"/>
                    </xsl:call-template>
                </ord:TargetMoveTime>
            </xsl:when>
            <xsl:when test="ZTNOR">
                <ord:TargetMoveTime>
                    <xsl:call-template name="transform-
NumberDelimeters">
                        <xsl:with-param name="number" se-
lect="ZTNOR"/>
                    </xsl:call-template>
                </ord:TargetMoveTime>
            </xsl:when>
        </xsl:choose>
        <xsl:choose>
            <xsl:when test="__FFMES__SOPPLD/ZEITN">
                <ord:TargetMoveTimeUnit><xsl:value-of se-
lect="__FFMES__SOPPLD/ZEITN"/></ord:TargetMoveTimeUnit>
                <xsl:when>
                    <xsl:when test="ZEITN">
                        <ord:TargetMoveTimeUnit><xsl:value-of se-
lect="EITN"/></ord:TargetMoveTimeUnit>
                    </xsl:when>
                </xsl:choose>
            </xsl:when>
        </xsl:choose>
    </xsl:when>

```

```

        </xsl:choose>

        <xsl:choose>
            <xsl:when test="__FFMES__SOPPLD/MINWE">
                <ord:MinimumSendAheadQuantity>
                    <xsl:call-template name="transform-
NumberDelimeters">
                        <xsl:with-param name="number" se-
lect="__FFMES__SOPPLD/MINWE"/>
                    </xsl:call-template>
                </ord:MinimumSendAheadQuantity>
            </xsl:when>
            <xsl:when test="MINWE">
                <ord:MinimumSendAheadQuantity>
                    <xsl:call-template name="transform-
NumberDelimeters">
                        <xsl:with-param name="number" se-
lect="MINWE"/>
                    </xsl:call-template>
                </ord:MinimumSendAheadQuantity>
            </xsl:when>
        </xsl:choose>

        <xsl:choose>
            <xsl:when test="__FFMES__SOPPLD/ZMINU">
                <ord:MinimumOverlapTime>
                    <xsl:call-template name="transform-
NumberDelimeters">
                        <xsl:with-param name="number" se-
lect="__FFMES__SOPPLD/ZMINU"/>
                    </xsl:call-template>
                </ord:MinimumOverlapTime>
            </xsl:when>
            <xsl:when test="ZMINU">
                <ord:MinimumOverlapTime>
                    <xsl:call-template name="transform-
NumberDelimeters">
                        <xsl:with-param name="number" se-
lect="ZMINU"/>
                    </xsl:call-template>
                </ord:MinimumOverlapTime>
            </xsl:when>
        </xsl:choose>
        <xsl:choose>
            <xsl:when test="__FFMES__SOPPLD/ZEIMU">
                <ord:MinimumOverlapTimeUnit><xsl:value-of se-
lect="__FFMES__SOPPLD/ZEIMU"/></ord:MinimumOverlapTimeUnit>
            </xsl:when>
            <xsl:when test="ZEIMU">
                <ord:MinimumOverlapTimeUnit><xsl:value-of se-
lect="ZEIMU"/></ord:MinimumOverlapTimeUnit>
            </xsl:when>
        </xsl:choose>
    
```

```

        </xsl:when>
    </xsl:choose>
<!-- End of FLS specific fields. -->

<!-- Template for processing customer specific Operation userFields -->
<!-- processing disabled in default xsl of order download
<xsl:if test="operationUserFields">
    <xsl:apply-templates select="processOperationUserFields"/>
</xsl:if>
-->

        <!-- If operation components are available, then add these to the
ERP-XML. -->
<xsl:if test="__FFMES__SAFOKO">
    <ord:OperationComponents>
        <xsl:if test="__FFMES__SAFOKO[VAKZ != 'D']">
            <xsl:call-template name="addOperationCom-
ponents"/>
        </xsl:if>

        <xsl:if test="__FFMES__SAFOKO[VAKZ='D']">
            <xsl:call-template name="deleteOpera-
tionComponents"/>
        </xsl:if>
    </ord:OperationComponents>
</xsl:if>

        <!-- If production resource tools are available, then add these to the
ERP-XML. -->
<xsl:if test="__FFMES__SAUFFH">
    <ord:ProductionResourceTools>
        <xsl:if test="__FFMES__SAUFFH[VAKZ != 'D']">
            <xsl:call-template name="addProductionRe-
sourceTools"/>
        </xsl:if>

        <xsl:if test="__FFMES__SAUFFH[VAKZ = 'D']">
            <xsl:call-template name="deleteProduction-
ResourceTools"/>
        </xsl:if>
    </ord:ProductionResourceTools>
</xsl:if>

<ord:Material>
    <Description>
        <Translation>
            <Language><xsl:value-of select="$languageCode"/></Language>
            <Text><xsl:value-of select="MTEXT"/></Text>
        </Translation>
    </Description>
    <Number><xsl:value-of select="MATNR"/></Number>

```

```

</ord:Material>
</ord:Operation>
    </xsl:for-each>
</xsl:template>

<!--
    Template for processing customer specific OPERATION userFields.
    Delete all unused UFxx fields and replace the '...' of the used fields with the customer specific
value.
    50 userfields from UF01 - UF50 are available (only the first then userfields are in the sample be-
low).
-->
<!--
Please enter the name of the xml node containing the operation user fields here.
<xsl:template name="processOperationUserFields">
    <xsl:for-each select="operationUserFields">
        <ord:UserFields>
            <xsl:if test="...">
                <UF1><xsl:value-of select="..."/></UF1>
            </xsl:if>
            <xsl:if test="...">
                <UF2><xsl:value-of select="..."/></UF2>
            </xsl:if>
            <xsl:if test="...">
                <UF3><xsl:value-of select="..."/></UF3>
            </xsl:if>
            <xsl:if test="...">
                <UF4><xsl:value-of select="..."/></UF4>
            </xsl:if>
            <xsl:if test="...">
                <UF5><xsl:value-of select="..."/></UF5>
            </xsl:if>
            <xsl:if test="...">
                <UF6><xsl:value-of select="..."/></UF6>
            </xsl:if>
            <xsl:if test="...">
                <UF7><xsl:value-of select="..."/></UF7>
            </xsl:if>
            <xsl:if test="...">
                <UF8><xsl:value-of select="..."/></UF8>
            </xsl:if>
            <xsl:if test="...">
                <UF9><xsl:value-of select="..."/></UF9>
            </xsl:if>
            <xsl:if test="...">
                <UF10><xsl:value-of select="..."/></UF10>
            </xsl:if>
        </xsl:for-each>
    </ord:UserFields>
</xsl:template>
-->

```

```

<!-- Template for adding operations to the ERP-XML that have to be deleted. -->
<xsl:template name="deleteOperations">
    <xsl:for-each select="____FFMES____SAFOLG[VAKZ = 'D']">
        <xsl:variable name="ORDER_NUMBER">
            <xsl:call-template name="removeLeadingZeros">
                <xsl:with-param name="originalString" select="AUFNR"/>
            </xsl:call-template>
        </xsl:variable>

        <ord:Operation action="DELETE">
            <ord:OrderERPKey>
                <Client><xsl:value-of select="MANDT"/></Client>
                <CompanyCode><xsl:value-of select="BUKRS"/></CompanyCode>
                <Plant><xsl:value-of select="WERKS"/></Plant>

                <xsl:if test="SYSID">
                    <SystemId><xsl:value-of select="SYSID"/></SystemId>
                </xsl:if>
            </ord:OrderERPKey>
            <ord:OperationNumber><xsl:value-of select="VORNR"/></ord:OperationNumber>
            <ord:OperationSplit><xsl:value-of select="VSPLT"/></ord:OperationSplit>

            <ord:ConfirmationNumber><xsl:value-of select="RUECK"/></ord:ConfirmationNumber>
            <ord:ERPStatusIds><xsl:value-of select="STATU_ID"/></ord:ERPStatusIds>
            <ord:ERPStatusCodes><xsl:value-of select="STATU"/></ord:ERPStatusCodes>
            <ord:ERPStatusCodesTranslation><xsl:value-of select="STATU_LA"/></ord:ERPStatusCodesTranslation>
        </ord:Operation>
    </xsl:for-each>
</xsl:template>

<!-- Template for adding the production resource tools to the ERP-XML that have to be deleted. -->
<xsl:template name="deleteProductionResourceTools">
    <xsl:for-each select="____FFMES____SAUFFH[VAKZ = 'D']">
        <ord:ProductionResourceTool action="DELETE">
            <ord:Number><xsl:value-of select="MATNR"/></ord:Number>
            <ord>Type><xsl:value-of select="FHART"/></ord>Type>
        </ord:ProductionResourceTool>
    </xsl:for-each>
</xsl:template>

<!-- Template for adding the production resource tools to the ERP-XML that have to be added or updated. -->
<xsl:template name="addProductionResourceTools">
    <xsl:for-each select="____FFMES____SAUFFH[VAKZ != 'D']">
        <xsl:variable name="PRODUCTION_RESOURCE_TOOL_ACTION">
            <xsl:choose>

```

```

        <!-- NOP --> no processing of the production resource tool. -->
        <xsl:when test="VAKZ = 'X'">NOP</xsl:when>
        <!-- Process production resource tool. -->
        <xsl:otherwise>ADD</xsl:otherwise>
    </xsl:choose>
</xsl:variable>

        <!-- Create ERP-XML for a production resource tool. -->
<ord:ProductionResourceTool action="${$PRODUCTION_RESOURCE_TOOL_ACTION}">
    <ord:Number><xsl:value-of select="MATNR"/></ord:Number>
    <ord:Type><xsl:value-of select="FHART"/></ord:Type>
    <ord:Sequence><xsl:value-of select="LFDNR"/></ord:Sequence>
    <ord:Quantity>
        <xsl:call-template name="transformNumberDelimiters">
            <xsl:with-param name="number" se-
lect="MGVGW"/>
        </xsl:call-template>
    </ord:Quantity>
    <ord:Unit><xsl:value-of select="MGEINH"/></ord:Unit>
    <ord:Group><xsl:value-of select="STATU"/></ord:Group>
    <ord:Description>
        <Translation>
            <Language><xsl:value-of select="$language-
Code"/></Language>
            <Text><xsl:value-of select="FHTXT"/></Text>
        </Translation>
    </ord:Description>
</ord:ProductionResourceTool>
</xsl:for-each>
</xsl:template>

        <!-- Template for adding the operation components to the ERP-XML that have to be added or
updated. -->
<xsl:template name="addOperationComponents">
    <xsl:for-each select="____FFMES____SAFOKO[VAKZ != 'D']">

        <xsl:variable name="OPERATION_COMPONENT_ACTION">
            <xsl:choose>
                <!-- NOP --> no processing of the operation component. -->
                <xsl:when test="VAKZ = 'X'">NOP</xsl:when>
                <!-- Process operation component. -->
                <xsl:otherwise>ADD</xsl:otherwise>
            </xsl:choose>
        </xsl:variable>

        <ord:OperationComponent action="${$OPERATION_COMPONENT_ACTION}">
            <ord:ComponentNumber><xsl:value-of se-
lect="MATNR"/></ord:ComponentNumber>
            <ord:PositionNumber><xsl:value-of select="RSPOS"/></ord:Position-
Number>

```

```

<ord:BaseUnitOfMeasure><xsl:value-of select="MEINS"/></ord:Ba-
seUnitOfMeasure>

    <xsl:if test="CHARG">
        <ord:BatchNumber><xsl:value-of select="CHARG"/></ord:Batch-
Number>
    </xsl:if>

        <ord:ConfirmationNumber><xsl:value-of se-
lect="RUECK"/></ord:ConfirmationNumber>

            <!-- If no description of the operation component is available, then
don't add an empty description. -->
            <xsl:if test="KTX01">
                <ord:Description>
                    <Translation>
                        <Language><xsl:value-of select="$languageCode"/></Language>
                        <Text><xsl:value-of select="KTX01"/></Text>
                    </Translation>
                </ord:Description>
            </xsl:if>

            <xsl:if test="MATKL">
                <ord:MaterialGroup><xsl:value-of select="MATKL"/></ord:Materi-
alGroup>
            </xsl:if>

            <xsl:if test="MTART">
                <ord:MaterialType><xsl:value-of select="MTART"/></ord:Material-
Type>
            </xsl:if>

            <!-- Create requirement date in format yyyy-MM-ddTHH:mm:00 -->
            <ord:RequirementDate>
                <xsl:call-template name="transformDateToTimestamp">
                    <xsl:with-param name="originalDate" se-
lect="BDTER"/>
                </xsl:call-template>
            </ord:RequirementDate>

            <ord:RequirementQuantity>
                <xsl:call-template name="transformNumberDelimiters">
                    <xsl:with-param name="number" select="BDMNG"/>
                </xsl:call-template>
            </ord:RequirementQuantity>
            <ord:ReservationNumber><xsl:value-of select="RSNUM"/></ord:Res-
ervationNumber>

            <xsl:if test="LGPLA">
                <ord:StorageBin><xsl:value-of select="LGPLA"/></ord:StorageBin>
            </xsl:if>

```

```
<xsl:if test="LGORT">
    <ord:StorageLocation><xsl:value-of select="LGORT"/></ord:StorageLocation>
</xsl:if>
    </ord:OperationComponent>
</xsl:for-each>
</xsl:template>

<!-- Template for deleting the operation components to the ERP-XML that have to be de-
leted. -->
<xsl:template name="deleteOperationComponents">
    <xsl:for-each select="____FFMES____SAFOKO[VAKZ = 'D']">
        <ord:OperationComponent action="DELETE">
            <ord:ComponentNumber><xsl:value-of se-
lect="MATNR"/></ord:ComponentNumber>
            <ord:PositionNumber><xsl:value-of select="RSPOS"/></ord:Position-
Number>
            </ord:OperationComponent>
        </xsl:for-each>
    </xsl:template>
</xsl:stylesheet>
```

4.2.2 Shifts

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="com.forcam.erp.download.shifts"
    xmlns:ct="com.forcam.erp.download.common" version="2.0">
    <!-- A XML output is generated. -->
    <xsl:output method="xml" indent="yes" />

    <!-- Remove all spaces (trim) in all elements. -->
    <xsl:strip-space elements="*" />

    <!-- Start point of processing the IDOC-XML. -->
    <xsl:template match="__FFMES__S">
        <xsl:apply-templates/>
    </xsl:template>

        <!-- Discard all data of the EDI_DC40 tag. -->
    <xsl:template match="EDI_DC40"/>

        <xsl:template match="IDOC">
            <!-- Shift supply is always a total shift supply -> add/update -->
            <ShiftsType xmlns="com.forcam.erp.download.shifts" xmlns:ct="com.for-
cam.erp.download.common" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ac-
tion="ADD" xsi:schemaLocation="com.forcam.erp.download.shifts ERPDownloadShifts.xsd">
                <xsl:apply-templates/>
            </ShiftsType>
        </xsl:template>

    <!--
        Convert the SAP-Timestamp to the XSL timestamp.
        YYYYMMDDHHMMSS -> YYYY-MM-DDTHH:MM:SS+Timezone
    -->
    <xsl:template name="transformSAPToXSLTimestamp">
        <xsl:param name="sapTimestamp"/>
        <xsl:param name="sapTimezone"/>

        <xsl:value-of select="concat(substring($sapTimestamp, 1, 4), '-',
            substring($sapTimestamp, 5, 2), '-',
            substring($sapTimestamp, 7, 2), 'T',
            substring($sapTimestamp, 9, 2), ':',
            substring($sapTimestamp, 11, 2), ':',
            substring($sapTimestamp, 13, 2), $sapTimezone)" />
    </xsl:template>

    <xsl:template match="__FFMES__SHIFT">
        <Shift>
            <ERPKey>
                <ct:Client><xsl:value-of select="MANDT"/></ct:Client>
                <ct:CompanyCode><xsl:value-of select="BUKRS"/></ct:CompanyCode>
                <ct:Plant><xsl:value-of select="WERKS"/></ct:Plant>

```

```

<xsl:if test="SYSID">
    <ct:SystemId><xsl:value-of select="SYSID"/></ct:SystemId>
</xsl:if>
</ERPKey>

<Workplace><xsl:value-of select="ARBPL"/></Workplace>

<!-- Convert sap shift code to shift type. -->
<xsl:choose>
    <!-- All types of early shift. -->
    <xsl:when test="KAPTPROG='EM'">
        <ShiftTypeCode>11</ShiftTypeCode>
    </xsl:when>
    <xsl:when test="KAPTPROG='EMP'">
        <ShiftTypeCode>11</ShiftTypeCode>
    </xsl:when>
    <xsl:when test="KAPTPROG='EMPT'">
        <ShiftTypeCode>11</ShiftTypeCode>
    </xsl:when>
    <xsl:when test="KAPTPROG='EMS'">
        <ShiftTypeCode>11</ShiftTypeCode>
    </xsl:when>
    <xsl:when test="KAPTPROG='EMV'">
        <ShiftTypeCode>11</ShiftTypeCode>
    </xsl:when>
    <xsl:when test="KAPTPROG='EJ'">
        <ShiftTypeCode>11</ShiftTypeCode>
    </xsl:when>
    <xsl:when test="KAPTPROG='2S-A'">
        <ShiftTypeCode>11</ShiftTypeCode>
    </xsl:when>

    <!-- All types of late shift. -->
    <xsl:when test="KAPTPROG='EA'">
        <ShiftTypeCode>12</ShiftTypeCode>
    </xsl:when>
    <xsl:when test="KAPTPROG='EAP'">
        <ShiftTypeCode>12</ShiftTypeCode>
    </xsl:when>
    <xsl:when test="KAPTPROG='EAPT'">
        <ShiftTypeCode>12</ShiftTypeCode>
    </xsl:when>
    <xsl:when test="KAPTPROG='EAS'">
        <ShiftTypeCode>12</ShiftTypeCode>
    </xsl:when>
    <xsl:when test="KAPTPROG='EAV'">
        <ShiftTypeCode>12</ShiftTypeCode>
    </xsl:when>
    <xsl:when test="KAPTPROG='2S-B'">
        <ShiftTypeCode>12</ShiftTypeCode>
    </xsl:when>

```

```

        </xsl:when>

        <!-- All types of night shift. -->
        <xsl:when test="KAPTPROG='EN'">
            <ShiftTypeCode>13</ShiftTypeCode>
        </xsl:when>
        <xsl:when test="KAPTPROG='END'">
            <ShiftTypeCode>13</ShiftTypeCode>
        </xsl:when>
        <xsl:when test="KAPTPROG='ENPT'">
            <ShiftTypeCode>13</ShiftTypeCode>
        </xsl:when>
        <xsl:when test="KAPTPROG='ENV'">
            <ShiftTypeCode>13</ShiftTypeCode>
        </xsl:when>
        <xsl:when test="KAPTPROG='ENVS'">
            <ShiftTypeCode>13</ShiftTypeCode>
        </xsl:when>

        <!-- Unknown shift tag -> set to -1. Would be removed by processing. -->
        <xsl:otherwise>
            <ShiftTypeCode>-1</ShiftTypeCode>
        </xsl:otherwise>
    </xsl:choose>

    <!-- Extract the day, month and year from shift day. -->
    <xsl:variable name="shiftStartYear" select="substring(VALID_FR, 7, 4)" />
    <xsl:variable name="shiftStartMonth" select="substring(VALID_FR, 4, 2)" />
    <xsl:variable name="shiftStartDay" select="substring(VALID_FR, 1, 2)" />

    <!-- Format the associated day: YYYY-MM-DD. -->
    <xsl:variable name="shiftAssociatedDay" select="concat($shiftStartYear, '-', $shiftStartMonth,
    '-', $shiftStartDay)" />

    <ShiftDate><xsl:value-of select="$shiftAssociatedDay" /></ShiftDate>

    <xsl:variable name="sapTimezone" select="UTCOFF" />

    <!-- Create timestamp for shift start and shift end in format YYYY-MM-
DDTHH:mm:00+HH:MM. -->
    <!-- Sample: 20111010045600, Offset: 02:00 -> 2011-10-10T04:56:00+02:00 -->
    <StartTime>
        <xsl:call-template name="transformSAPToXSLTimestamp">
            <xsl:with-param name="sapTimestamp" select="SHIFT_STRT" />
            <xsl:with-param name="sapTimezone" select="UTCOFF" />
        </xsl:call-template>
    </StartTime>

    <EndTime>
        <xsl:call-template name="transformSAPToXSLTimestamp">
            <xsl:with-param name="sapTimestamp" select="SHIFT_ENDE" />

```

```

<xsl:with-param name="sapTimezone" select="UTCOFF"/>
</xsl:call-template>
</EndDateTime>

    <!-- Shift break available? -->
    <xsl:if test="__FFMES__BREAK">
        <Breaks>
            <!-- Process all shift breaks of the shift. -->
            <xsl:for-each select="__FFMES__BREAK">
                <Break>
                    <StartDateTime>
                        <xsl:call-template name="transformSAPToXSLTimestamp">
                            <xsl:with-param name="sapTimestamp" select="BREAK_STRT"/>
                            <xsl:with-param name="sapTimezone" select="UTCOFF"/>
                        </xsl:call-template>
                    </StartDateTime>

                    <EndDateTime>
                        <xsl:call-template name="transformSAPToXSLTimestamp">
                            <xsl:with-param name="sapTimestamp" select="BREAK_ENDE"/>
                            <xsl:with-param name="sapTimezone" select="UTCOFF"/>
                        </xsl:call-template>
                    </EndDateTime>
                </Break>
            </xsl:for-each>
            </Breaks>
        </xsl:if>
    </Shift>
</xsl:template>
</xsl:stylesheet>

```

4.2.3 Personnel

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="com.forcam.erp.download.persons" version="2.0">
    <!-- A XML output is generated. -->
    <xsl:output method="xml" indent="yes" />

    <!-- Remove all spaces (trim) in all elements. -->
    <xsl:strip-space elements="*" />

    <!-- Start point of processing the IDOC-XML. -->
    <xsl:template match="__FFMES__H">
        <xsl:apply-templates/>
    </xsl:template>

    <!-- Daten aus EDI_DC40 verwerfen -->
    <xsl:template match="EDI_DC40"/>

    <xsl:template match="IDOC">
        <PersonsType xmlns="com.forcam.erp.download.persons"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="com.forcam.erp.download.persons ERPDownloadPersons.xsd" action="ADD">
            <xsl:apply-templates/>
        </PersonsType>
    </xsl:template>

    <xsl:template match="__FFMES__HRDATA">
        <Person>
            <PersonERPKey>
                <Client><xsl:value-of select="MANDT"/></Client>
                <CompanyCode><xsl:value-of select="BUKRS"/></CompanyCode>
                <PersonnelArea><xsl:value-of select="WERKS"/></PersonnelArea>

                <xsl:if test="SYSID">
                    <SystemId><xsl:value-of select="SYSID"/></SystemId>
                </xsl:if>
            </PersonERPKey>
            <PersonNumber><xsl:value-of select="PERNR"/></PersonNumber>
            <IdentificationNumber><xsl:value-of select="ZAUSW"/></Identification-
Number>
                <FirstName><xsl:value-of select="VORNA"/></FirstName>
                <LastName><xsl:value-of select="NACHN"/></LastName>
                <CostCenter><xsl:value-of select="KOSTL"/></CostCenter>
            </Person>
        </xsl:template>
    </xsl:stylesheet>

```