



Version 5.10

ERP Interface - Daten und Events via XML Handbuch

Dokument:	Handbuch - ERP Interface - Daten und Events via XML.docx
-----------	--

Erstellt:	24.04.17
-----------	----------

Letzte Änderung:	30.09.19
------------------	----------

Autor:	AEgilmez
--------	----------



Inhaltsverzeichnis

1	Allgemein	3
2	Download Interface	4
2.1	Allgemein.....	4
2.1.1	Implementierte RFC Verbindung SAP (ERP) an FORCAM FORCE™	4
2.2	Download Strukturen	5
2.2.1	Übertragung von Zeitfeldern	5
2.2.2	SAP – Download WSDL (webservice description language)	7
2.2.3	Auftragsdaten	8
2.2.3.1	Auftragskopfdaten.....	8
2.2.3.2	Sachmerkmale des Materials	15
2.2.3.3	Arbeitsvorgangsdaten	19
2.2.3.4	Langtexte zu AVOs.....	30
2.2.3.5	AVO Komponenten.....	33
2.2.3.6	Fertigungshilfsmittel.....	37
2.2.4	Maschinenkapazitäten / Schichtpläne.....	41
2.2.5	Personaldaten (Ministammsätze Personal).....	45
3	Upload Interface	47
3.1	Allgemein.....	47
3.1.1	Anwendung Implementierte RFC Verbindung SAP (ERP)	48
3.2	FORCAM FORCE™ Rückmeldesatz	48
3.3	Rückmeldeereignisse.....	52
3.3.1	Mappinglayer XSLT (XSL Schema Produktstandard).....	53
3.4	Gefüllte Tags im SAP-IDoc abhängig von der Satz-Art.....	54
3.5	Beispiel einer Mengenmeldung	60
3.6	Serialisierung und Korrekturfähigkeit	61
3.6.1	Arbeitsplatz - Serialisierung.....	62
3.6.2	AVO - Serialisierung	62
4	Anhang	63
4.1	Upload XSLT Mapper	63
4.2	Download XSLT Mapper	78
4.2.1	Aufträge	78
4.2.2	Schichten	107
4.2.3	Personal	111

1 Allgemein

In diesem Dokument wird die von FORCAM aktuell implementierte und für einen Upload und Download zur Verfügung gestellte Daten- / Ereignis – und Prozessstruktur zugeschnitten auf SAP als ERP System beschrieben.

Das einem ERP-System unterlagerte FORCAM FORCE™-System stellt für die Kommunikation / Datenaustausch einen Download Service zur Verfügung. Für den Informationsrückfluss in Form eines Rückmeldeprozesses stellt das ERP einen entsprechend Upload-Dienst zur Verfügung. Innerhalb von FORCAM FORCE™ kann durch eine Rule-Engine zu den Ereignissen z.B. zu berechnende Kennwerte hinzugefügt werden (Beispielsweise: Ist-Stückzeit eines Auftrages). Diesen allgemeinen Zusammenhang zeigt Bild 1.

Die aktuell implementierten Download- und Upload-Daten-Konstrukte sind auf das ERP-System SAP ausgelegt und implementiert. Durch Spezifikationen kann diese entsprechend auf andere ERP-Systeme angepasst und verändert werden.

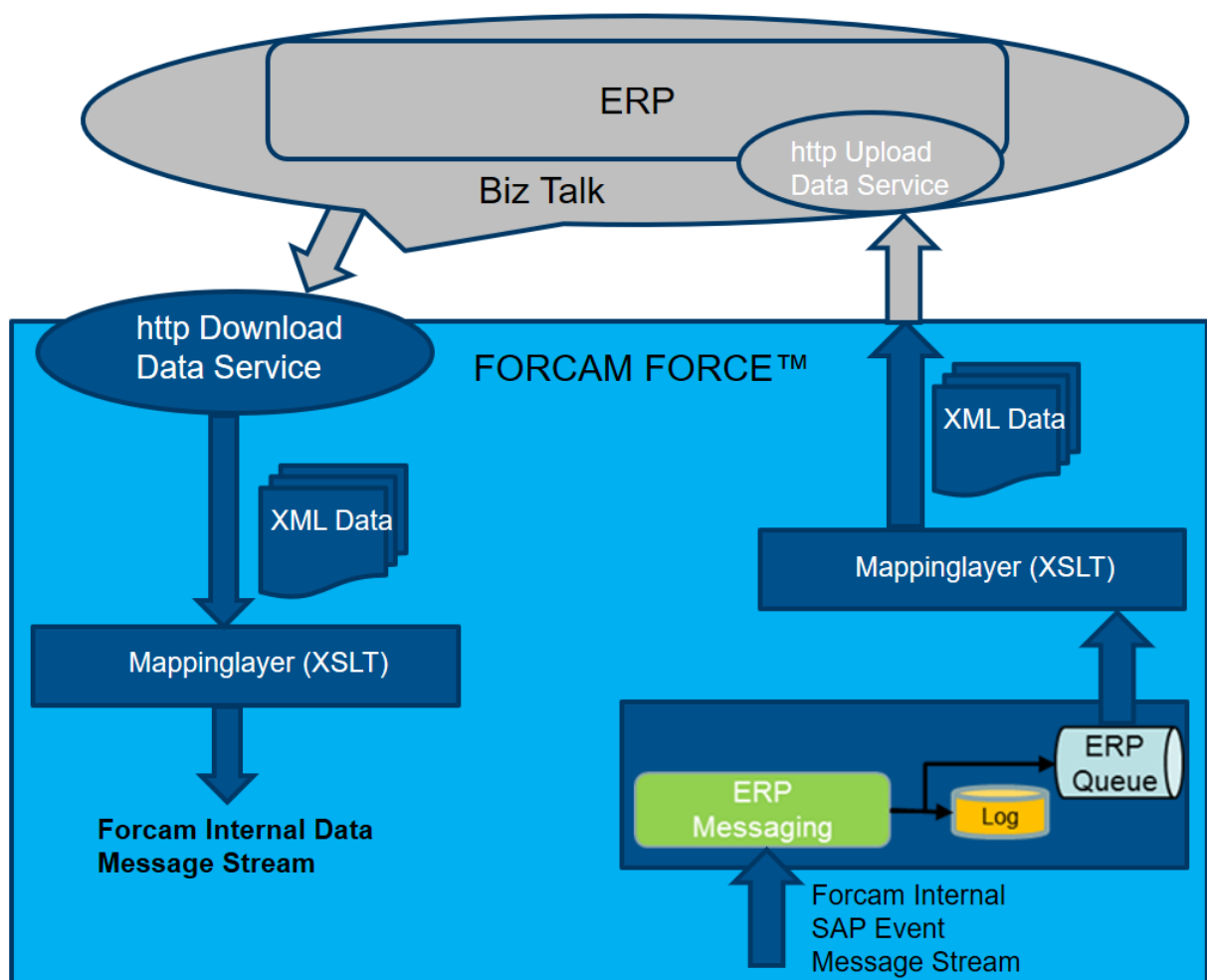


Bild 1: Upload und Download Struktur zwischen ERP und FORCAM FORCE™

2 Download Interface

2.1 Allgemein

Im übergeordneten ERP werden die für den Download notwendigen und definierten Daten in eine definierte Datenstruktur überführt. Der Daten-Download an das unterlagerte FORCAM FORCE™ wird durch Ereignisse im ERP System angestoßen unter Verwendung eines in FORCAM FORCE™ zur Verfügung gestellten Download-Service.

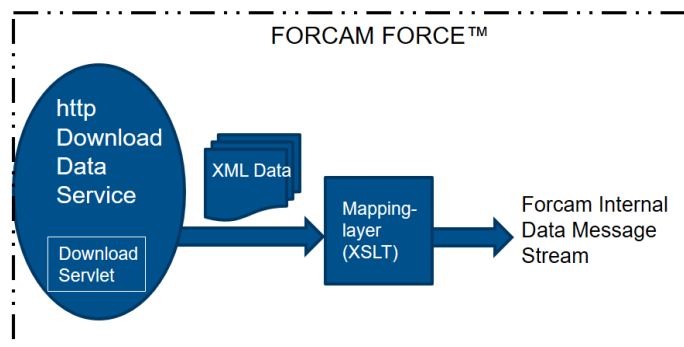


Bild 2: Allgemeine Download Struktur für ERP → FORCAM FORCE™

2.1.1 Implementierte RFC Verbindung SAP (ERP) an FORCAM FORCE™

Die Kommunikationsverbindung zwischen SAP (ERP) und FORCAM FORCE™ erfolgt in der aktuellen Implementierung und Ausprägung immer unter Verwendung einer expliziten RFC – Kommunikation (Destination). Diese Struktur zeigt Bild 3.

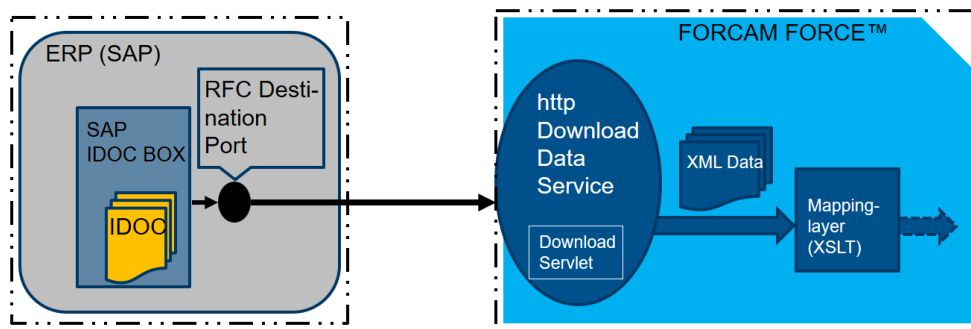


Bild 3: Download-Verbindung von SAP -> FORCAM FORCE™ unter Verwendung RFC

In SAP wird eine XML-HTTP Port mit einer entsprechenden RFC Destination angelegt (content type = text/xml). Die RFC Destination wird im SAP angelegt und besteht aus den Parametern

- http Verbindung zu ext. Server
- Zielmaschine (FORCAM FORCE™ Server)
- Pfadpräfix
- Service Nummer = Kommunikationsport (z.B. 10080)

Das IDOC wird auf Betriebssystemebene über den Port als XML Datei versendet. FORCAM FORCE™ wartet auf eine Übertragung und reicht das XML dann zur Weiterverarbeitung durch.

2.2 Download Strukturen

Dem Download-Prozess werden definierte XML-Datenstrukturen übergeben. Unter Anwendung eines Mappinglayers via XSL werden die Daten interpretiert und in ein FORCAM internes Format umgewandelt.

Aktuell kommen folgenden Abstrakt-Daten-Strukturen im Downloadprozess zur Anwendung:

- Auftragsdaten
 - Auftrag
 - Produktmerkmale sind im Standard nicht enthalten
 - AVO
 - Fertigungshilfsmittel: nur notwendig, wenn zur Anzeige benötigt oder falls TDM Werkzeugdatenverwaltung eingesetzt wird oder falls FORCAM NC-Datenverwaltung verwendet wird.
 - Komponenten: nur notwendig, wenn die zu verbauenden Komponenten angezeigt oder ans ERP zurückgemeldete werden sollen
- Personaldaten können auch in FORCAM FORCE™ gepflegt werden
- Schichtdaten können auch in FORCAM FORCE™ gepflegt werden

i Nur die Auftragsdaten und die zugehörigen Vorgänge werden zwingend benötigt. Sämtliche andere Daten werden nur zu Anzeigezwecken oder für funktionale Erweiterungen wie TDM-Werkzeugdatenverwaltung, NC-Programmverwaltung oder Komponenten-Verprobung (Trace) benötigt.

2.2.1 Übertragung von Zeitfeldern

Bei der Definition von einem Zeitfeld muss immer zusätzlich ein eigenes separates Feld für die Einheit definiert werden.

Basierend auf der übertragenen Einheit wird der übergebene Wert des Zeitfelds auf einen FORCAM FORCE™ internen einheitlichen Zeitbasiswert umgerechnet. Ohne eine gültige Einheit kann nicht sichergestellt werden, dass der Zeitwert richtig umgerechnet und interpretiert wird.

Folgende Einheiten bei Zeitwerten werden für den ERP-Download unterstützt:

Tabelle 1: Unterstützte Einheiten bei Zeitwerten

Zeittyp	Beschreibung
H	Stunde
STD	Stunde
M_90	Minute / 90
MIN	Minute
MS	Millisekunde
S	Sekunde

Download Interface

TMU	Technische Minute
-----	-------------------

2.2.2 SAP – Download WSDL (webservice description language)

Für den SAP-Download steht ein „Download Servlet“ zu Verfügung unter Anwendung einer Datei vom Typ *.wsdl (webservice description language).

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

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

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

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

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

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

</definitions>
```

2.2.3 Auftragsdaten

2.2.3.1 Auftragskopfdaten

Diese Daten werden im IDOC Segment /FFMES/SAUFTR mit der Satzart **AUFTR** an das Subsystem übergeben.

Die Datenstruktur ist in Tabelle 1 aufgeführt.

Kundenindividuell

- ① Alle **rot markierten Elemente** sind Sonderfelder, welche in der XML-Datenstruktur als leere Elemente generiert werden und sind nicht automatisch Inhalt des Produktstandards FORCAM Adapter für SAP. Sollen diese Felder verwendet werden entstehen Spezifikations- und Customizing Aufwände aus welcher Quelle diese Daten in welcher Form abzuholen sind. Diese Kundenindividuelle Ausprägung kann unter Anwendung von User-Exits implementiert werden.

Tabelle 2: Datenstruktur Auftragskopfdaten (Quelle SAP (ERP) und Ziel FORCAM FORCE™)

SAP (ERP)			Bezeichnung	FORCAM FORCE™					
				Pflichtfelder			Persistieren		
SAP-Feld	Typ (Länge)	IDOC XML SAP		Download	Löschsatz	Interne XML Transformation	Tabelle	Spalte	Typ
	C (5)	SART	Satzart (Hier: AUFTR)			-	-	-	-
	C (1)	VAKZ	Verarbeitungskennzeichen D oder S (D = Löschsatz)			-	-	-	-
AUFK-MANDT	C (3)	MANDT	Mandant			client	FR_MD_ERP_KEY	CLIENT	VAR-CHAR2(255 CHAR
AUFK-BUKRS	C (4)	BUKRS	Buchungskreis			companyCode	FR_MD_ERP_KEY	COMPANY_CODE	VAR-CHAR2(255 CHAR
AUFK-WERKS	C (4)	WERKS	Werk			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

Download Interface

AUFK-AUFNR	C (12)	AUFNR	Auftragsnummer			orderNumber	FR_MD_ORDER	ORDER_NUMBER	VAR-CHAR2(255 CHAR)
	C (1)	ASPLT	Auftrags-split (Default: 0)			orderSplit	FR_MD_ORDER	ORDER_SPLIT	VAR-CHAR2(255 CHAR)
AFPO-POSNR (immer 0001)	N (4)	POSNR	Auftragsposition (Immer 0001)			positionNumber	FR_MD_ORDER	POSITION_NUMBER	VAR-CHAR2(255 CHAR)
AFPO-KNTTP	C (1)	KNTTP	Kontierungstyp			accountingType	FR_MD_ORDER	ACCOUNT_TYPE	VAR-CHAR2(255 CHAR)
AFPO-KDAUF	C (25)	KOOBJ	Adapter: Kontierungsobjekt			accountingObject	FR_MD_ORDER	ACCOUNT_OBJECT	VAR-CHAR2(255 CHAR)
AFPO-PROJN (konv.)	C (40)	ATEXT	Auftragstext			description	FF_MD_LITERAL_TEXT	TEXT	VAR-CHAR2(1024 CHAR)
oder leer	C (18)	MATNR	Materialnummer			material	FR_MD_MATERIAL	MATERIAL_NUMBER	VAR-CHAR2(255 CHAR)
AUFK-KTEXT	C (40)	MTEXT	Kurztext Material			material/description	FF_MD_LITERAL_TEXT	TEXT	VAR-CHAR2(1024 CHAR)
leer	C (20)	KUNNR	Kundennummer			customer-Number	FR_MD_MATERIAL	CUSTOMER_NUMBER	VAR-CHAR2(255 CHAR)
AFKO-GSTRP	DD.MM.YYYY (10)	SSTRT	Sollstarttermin / Eckstart-Termin			targetStart-Date	FR_MD_ORDER	TARGET_START	TI-MESTAMP(6)

Download Interface

AFPO-STRMP	DD.MM.YYYY (10)	PSTRT	Starttermin des Planauftrags			planStartDate	FR_MD_ORDER	PLAN_START	TI-MESTAMP(6)
AFKO-GLTRP	DD.MM.YYYY (10)	SLIEF	Sollliefertermin / Eck-Endtermin			targetEndDate	FR_MD_ORDER	TARGET_END	TI-MESTAMP(6)
AFPO-LTRMP	DD.MM.YYYY (10)	PLIEF	Liefertermin aus Planauftrag			planEndDate	FR_MD_ORDER	PLAN_END	TI-MESTAMP(6)
AFKO-GAMNG	C (17)	AFANZ	Auftragsmenge			targetQuantity	FR_MD_ORDER	TARGET_QUANTITY	VAR-CHAR2(255 CHAR)
AFKO-GMEIN	C (3)	AMEIN	Mengen- einheit für Anzeige (Default: stk)			displayQuantityUnit	FR_MD_ORDER	DISPLAY_QUANTITY_UNIT	VAR-CHAR2(255 CHAR)
TJ02T-TXT04 in DEUTSCH	C (80)	STATU	Auftrags- status Text z.B. FREI SSAP... (SAP Sys- temsta- tus)			ppsStatus	FR_MD_ORDER	ERP_STATUS	VAR-CHAR2(255 CHAR)
TJ02T-ISTAT (Status Codes in- tern)	C (80)	STATU_ID	Status_ID (System- status) z.B. I0002 (Auftrags- kopf)			erpStatusIds	FR_MD_ORDER	ERP_STATUS_ID	VAR-CHAR2(255 CHAR)

Download Interface

TJ02T-TXT04 in der Sprache aus der Tabelle /FFMES/AUFTR	C (80)	STATU_LA	System Statustext in der Auftrags-sprache (Auftragskopf)			erpStatusLa	FR_MD_ORDER	ERP_STATUS_LA	VAR-CHAR2(255 CHAR)
leer oder X (wird auf X gesetzt, wenn Auftragsstatus SPER (Systemstatus) gesetzt ist)	C (1)	SPRKZ	Sperrkennzeichen			lockIndicator	FR_MD_ORDER	LOCK_INDICATOR	VAR-CHAR2(255 CHAR)
AFKO-APRIO	C (1)	APRIO	Auftragspriorität			orderPriority	FR_MD_ORDER	ORDER_PRIORITY	VAR-CHAR2(255 CHAR)
AFKO-FEVR	C (3)	FEVR	Fertigungssteuerer			productionScheduler	FR_MD_ORDER	PRODUCTION_SCHEDULER	VAR-CHAR2(255 CHAR)
AFKO-PLGRP	C (3)	VAGRP	Verantw. Planer-grp/Abteilung			plannerGroupInCharge	FR_MD_ORDER	PLANNER_GROUP_IN_CHARGE	VAR-CHAR2(255 CHAR)
AFKO-FTRMI	DD.MM.YYYY (10)	FRDAT	Freigabedatum			releaseDate	FR_MD_ORDER	RELEASE_DATE	TI-MESTAMP(6)
AFKO-REDKZ	C (1)	REDKZ	Reduzierkennzeichen			reductionIndicator	FR_MD_ORDER	REDUCTION_INDICATOR	VAR-CHAR2(255 CHAR)
leer	C (12)	PRFLS	Prüflosnummer			testBatchNumber	FR_MD_ORDER	TEST_BATCH_NUMBER	VAR-CHAR2(255 CHAR)
leer	C (20)	SERNR	Seriennummer			serialNumber	FR_MD_ORDER	SERIAL_NUMBER	VAR-CHAR2(255 CHAR)

Download Interface

leer	C (80)	ZCHNR	Zeichnungsnummer			drawingNumber	FR_MD_ORDER	DRAWING_NUMBER	VAR-CHAR2(255 CHAR)
leer	C (3)	PROJN	Projekt			project	FR_MD_ORDER	PROJECT	VAR-CHAR2(255 CHAR)
leer	C (6)	MODEL	Modell			model	FR_MD_ORDER	MODEL	VAR-CHAR2(255 CHAR)
leer	C (4)	GERAE	Gerät			device	FR_MD_ORDER	DEVICE	VAR-CHAR2(255 CHAR)
AFPO-LGORT	C (4)	LGORT	Lagerort			storageLocation	FR_MD_ORDER	STORAGE_LOCATION	VAR-CHAR2(255 CHAR)
leer (nur bei REM (Repetitive Manufacturing))	C (4)	VERID	Fertigungsversion (wird bei REM (Repetitive Manufacturing) verwendet)			productionVersion	FR_MD_ORDER	PRODUCTION_VERSION	VAR-CHAR2(255 CHAR)
TJ30T-ASTTX	C (40)	ASTTX	Anwender-status Text in Deutsch (Auftragskopf)			userStatus	FR_MD_ORDER	USER_STATUS	VAR-CHAR2(255 CHAR)
TJ30T-ESTAT (Status Codes extern)	C (40)	ASTTX_ID	Status ID (Userstatus) z.B. E0005			-	-	-	

Download Interface

			(Auftragskopf)						
TJ30T-TXT04 in der Sprache aus der Tabelle /FFMES/AUFTR	C (40)	ASTTX_LA	User Statustext in der Auftragssprache (Auftragskopf)			-	-	-	
AUFG-AUART	C (4)	AUART	Auftragsart						

2.2.3.2 Sachmerkmale des Materials

Diese Daten werden im IDOC Segment /FFMES/SAUFMK mit der **Satzart AUFMK** an das Subsystem übergeben, sofern dies in der Konfiguration so eingestellt wurde. Die Konfiguration wird über die Tabelle /FFMES/AUFTR mit dem Property **AKTIV_MKL** eingestellt.

Der Produktstandard stellt eine allgemeine Datendefinition für diese Daten bereit. Die Ausprägungen von Sachmerkmalen sind jedoch kundenspezifisch definiert und werden nicht im Produktstandard gefüllt. Bei Verwendung entstehen Spezifikations- und Customizing Aufwände aus welcher Quelle diese Daten in welcher Form abzuholen sind. Diese Kundenindividuelle Ausprägung wird durch Implementierung einer Erweiterungsmethode unter Anwendung von User-Exits implementiert werden.

Diese Erweiterungsmethode als Struktur, ist im Unterprogramm zum Füllen des Segments /FFMES/SAUFMK enthalten und wird im Produktstandard mit ausgeliefert. Die Methode heißt **FILL_CLASSIFICATION** und die verwendeten Parameter sind in Tabelle 2 aufgeführt.

Tabelle 3: Verwendete Parameter der Methode FILL_CLASSIFICATION im Adapter

Parameter	Typisierung	Beschreibung
VALUE(IM_AUFK)	TYPE AUFK OPTIONAL	Auftragsstammdaten
VALUE(IM_AFKO)	TYPE AFKO OPTIONAL	Auftragskopfdaten PPS-Aufträge
VALUE(IM_AFPO)	TYPE AFPO OPTIONAL	Auftragsposition
VALUE(STRUCT_AUFTR)	TYPE /FFMES/AUFTR OPTIONAL	FORCAM Adapter für SAP: Parameter: Auftragsversorgung
VALUE(IM_MARA)	TYPE MARA OPTIONAL	Allgemeine Materialdaten
VALUE(IM_MARC)	TYPE MARC OPTIONAL	Werksdaten zum Material
VALUE(RETURN)	TYPE BAPIRET2 OPTIONAL	Returnparameter
VALUE(STRUCT_AUFMK)	TYPE /FFMES/SAUFMK OPTIONAL	Sachmerkmale des Materials
VALUE(STRUCT_ENHDATA)	TYPE /FFMES/ENHDATA OPTIONAL	Erweiterung mit Kundenfeldern

Die Datenstruktur ist in **Tabelle 4** aufgeführt.

Kundenindividuell


-  Alle **rot markierten Elemente** sind Sonderfelder, welche in der XML-Datenstruktur als leere Elemente generiert werden und sind nicht automatisch Inhalt des Produktstandards FORCAM Adapter für SAP. Sollen diese Felder verwendet werden entstehen Spezifikations- und Customizing Aufwände aus welcher Quelle diese Daten in welcher Form abzuholen sind. Diese Kundenindividuelle Ausprägung kann unter Anwendung von User-Exits implementiert werden.

Tabelle 4: Datenstruktur Sachmerkmale Material (Quelle SAP (ERP) und Ziel FORCAM FORCE™)

SAP (ERP)			Bezeichnung	FORCAM FORCE™					
				Pflichtfelder			Persistieren		
SAP-Feld	Typ (Länge)	IDOC XML SAP		Down-load	Lösch-satz	Interne XML Transforma-tion	Tabelle	Spalte	Typ
AUFMK	C (5)	SART	Satzart (Hier: AUFMK)			-	-	-	-
S	C (1)	VAKZ	Verarbei-tungs-kennzei-chen (Immer S)			-	-	-	-
AUFK-MANDT	C (3)	MANDT	Mandant			client	FR_MD_ERP_KEY	CLIENT	VAR-CHAR2(255 CHAR
AUFK-BUKRS	C (4)	BUKRS	Buchungs-kreis			companyCode	FR_MD_ERP_KEY	COMPANY_CODE	VAR-CHAR2(255 CHAR
AUFK-WERKS	C (4)	WERKS	Werk			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	Auftrags-nummer			orderNumber	FR_MD_ORDER	ORDER_NUMBER	VAR-CHAR2(255 CHAR

Download Interface

Immer 0	C (1)	ASPLT	Auftrags-split (Default: 0)			orderSplit	FR_MD_ORDER	ORDER_SPLIT	VAR-CHAR2(255 CHAR)
AFPO-POSNR	N (4)	POSNR	Auftrags-position			orderItem	FR_MD_MATERIAL_CHARACTERISTIC	ORDER_ITEM	VAR-CHAR2(255 CHAR)
leer	C (40)	CLASS	Bezeichnung Klassenart			classType	FR_MD_MATERIAL_CHARACTERISTIC	CLASS_TYPE	VAR-CHAR2(255 CHAR)
Immer ,CLASS'	C (30)	ATNAM	Merkmal-name			characteristicName	FR_MD_MATERIAL_CHARACTERISTIC	CHARACTERISTIC_NAME	VAR-CHAR2(255 CHAR)
leer	C (30)	ATWRT	Merkmal-wert			characteristicValue1	FR_MD_MATERIAL_CHARACTERISTIC	CV01	VAR-CHAR2(255 CHAR)
leer	C (30)	ATW02	Merkmal-wert2			characteristicValue2	FR_MD_MATERIAL_CHARACTERISTIC	CV02	VAR-CHAR2(255 CHAR)
leer	C (30)	ATW03	Merkmal-wert3			characteristicValue3	FR_MD_MATERIAL_CHARACTERISTIC	CV03	VAR-CHAR2(255 CHAR)
leer	C (30)	ATW04	Merkmal-wert4			characteristicValue4	FR_MD_MATERIAL_CHARACTERISTIC	CV04	VAR-CHAR2(255 CHAR)
leer	C (30)	ATW05	Merkmal-wert5			characteristicValue5	FR_MD_MATERIAL_CHARACTERISTIC	CV05	VAR-CHAR2(255 CHAR)
leer	C (30)	ATW06	Merkmal-wert6			characteristicValue6	FR_MD_MATERIAL_CHARACTERISTIC	CV06	VAR-CHAR2(255 CHAR)
leer	C (30)	ATW07	Merkmal-wert7			characteristicValue7	FR_MD_MATERIAL_CHARACTERISTIC	CV07	VAR-CHAR2(255 CHAR)

Download Interface

leer	C (30)	ATW08	Merkmal- wert8			characteristi- cValue8	FR_MD_MATE- RIAL_CHARAC- TERISTIC	CV08	VAR- CHAR2(255 CHAR
leer	C (30)	ATW09	Merkmal- wert9			characteristi- cValue9	FR_MD_MATE- RIAL_CHARAC- TERISTIC	CV09	VAR- CHAR2(255 CHAR
leer	C (30)	ATW10	Merkmal- wert10			characteristi- cValue10	FR_MD_MATE- RIAL_CHARAC- TERISTIC	CV10	VAR- CHAR2(255 CHAR
leer	C (30)	ATW11	Merkmal- wert11			characteristi- cValue11	FR_MD_MATE- RIAL_CHARAC- TERISTIC	CV11	VAR- CHAR2(255 CHAR
leer	C (30)	ATW12	Merkmal- wert12			characteristi- cValue12	FR_MD_MATE- RIAL_CHARAC- TERISTIC	CV12	VAR- CHAR2(255 CHAR
leer	C (30)	ATW13	Merkmal- wert13			characteristi- cValue13	FR_MD_MATE- RIAL_CHARAC- TERISTIC	CV13	VAR- CHAR2(255 CHAR

2.2.3.3 Arbeitsvorgangsdaten

Diese Daten werden im IDOC Segment /FFMES/SAFOLG mit der **Satzart AFOLG** an das Subsystem übergeben, sofern dies in der Konfiguration so eingestellt wurde. Die Konfiguration wird über die Tabelle /FFMES/AUFTR mit dem Flag AKTIV_AVO eingestellt.

Die Datenstruktur ist in **Tabelle 5** aufgeführt.

Kundenindividuell

- ❗ Alle **rot markierten Elemente** sind Sonderfelder, welche in der XML-Datenstruktur als leere Elemente generiert werden und sind nicht automatisch Inhalt des Produktstandards FORCAM Adapter für SAP. Sollen diese Felder verwendet werden entstehen Spezifikations- und Customizing Aufwände aus welcher Quelle diese Daten in welcher Form abzuholen sind. Diese Kundenindividuelle Ausprägung kann unter Anwendung von User-Exits implementiert werden.
- ❗ Speziell für die SAP Felder AFVV-VGW01, AFVV-VGW02, AFVV-VGW03 gibt es die Möglichkeit in der Tabelle STVAL pro Arbeitsplatz zu pflegen welcher Vorgabewert welchem Typ entspricht um der statischen Reihenfolgevorgabe entsprechen zu können.

Tabelle 5: Datenstruktur Arbeitsvorgangsdaten (Quelle SAP (ERP) und Ziel FORCAM FORCE™)

SAP (ERP)			Bezeichnung	FORCAM FORCE™					
				Pflichtfelder			Persistieren		
SAP-Feld	Typ (Länge)	IDOC XML SAP		Down-load	Lösch-satz	Interne XML Transforma-tion	Tabelle	Spalte	Typ
AUFMK	C (5)	SART	Satzart (Hier: AUFMK)			-	-	-	-
S	C (1)	VAKZ	Verarbei-tungskenn-zeichen (Immer S)			-	-	-	-
AUFK-MANDT	C (3)	MANDT	Mandant			client	FR_MD_ERP_KEY	CLIENT	VAR-CHAR2(255 CHAR
AUFK-BUKRS	C (4)	BUKRS	Buchungs-kreis			com-panyCode	FR_MD_ERP_KEY	COMPANY_CODE	VAR-CHAR2(255 CHAR
AUFK-WERKS	C (4)	WERKS	Werk			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	Auftrags-nummer			orderNumber	FR_MD_OPERA-TION	ORDER_NUMBER	VAR-CHAR2(32 CHAR)
Immer 0	C (1)	ASPLT	Auftragssplit			orderSplit	FR_MD_OPERA-TION	ORDER_SPLIT	VARCHAR2(8 CHAR)

Download Interface

			(Immer 0, ist kein SAP Split)						
AFVC-RUECK	N (10)	RUECK	Rückmelde- nummer des Vorgangs			confirmation- Number	FR_MD_OPERA- TION	CONFIRMATION_NUMBER	VAR- CHAR2(255 CHAR)
AFVC-VORNR	C (4)	VORNR	AVO			operation- Number	FR_MD_OPERA- TION	OPERATION_NUMBER	VAR- CHAR2(32 CHAR)
Immer 0	C (1)	VSPLT	AVO Split			operationSplit	FR_MD_OPERA- TION	OPERATION_SPLIT	VARCHAR2(8 CHAR)
AFVC-LTXA1 und AFVC-LTXA2	C (80)	LTXA1	AVO-Kurz- text			description	FF_MD_LITE- RAL_TEXT	TEXT	VAR- CHAR2(1024 CHAR)
AFVV-SSAVD	DD.MM.YYYY (10)	SSAVD	Spätester terminierter Start, Datum Sollstart			planStartDate	FR_MD_OPERA- TION	TARGET_START_DATE	TI- MESTAMP(6)
AFVV-SSAVZ Wenn 24:00 dann wird 23:59 gesetzt, Format HH:MM	HH.MM (5)	SSAVZ	Spätester terminierter Start, Uhr- zeit			planStartTime	FR_MD_OPERA- TION		
AFVV-SSEDD	DD.MM.YYYY (10)	SSEDD	Spätestes terminiertes Ende-Datum Sollende			planEndDate	FR_MD_OPERA- TION	TARGET_END_DATE	TI- MESTAMP(6)
AFVV-SSEDZ Wenn 24:00 dann wird 23:59 gesetzt, Format HH:MM	HH.MM (5)	SSEDZ	Spätestes terminiertes Ende-Uhr- zeit			planEndTime	FR_MD_OPERA- TION		
AFVV-MGVRG Menge in der	C (17)	MGVRG	Vorgabe- menge			targetQuan- tity	FR_MD_OPERA- TION	TARGET_QUANTITY	FLOAT

Download Interface

Einheit AFVV-MEINH									
AFVV-MEINH	C (3)	MEINH	Mengeneinheit			displayQuantityUnit	FR_MD_OPERATION	DISPLAY_QUANTITY_UNIT	VAR-CHAR2(255 CHAR)
AFVV-LMNGA	C (17)	LMNGA	Gemeldete Gutmenge			yield quantity	FR_MD_OPERATION	ERP_YIELD_QUANTITY	FLOAT
AFVV-XMNGA	C (17)	XMNGA	Gemeldete Ausschussmenge			scrap quantity	FR_MD_OPERATION	ERP_SCRAP_QUANTITY	FLOAT
AFVV-RMNGA	C (17)	RMNGA	Gemeldete Nacharbeitsmenge			rework quantity	FR_MD_OPERATION	ERP_REWORK_QUANTITY	FLOAT
errechnet aus AFVV-MGVRG / 100 * AFVC-AUFAK in Einheit AFVV-MEINH	C (17)	AUSSS	Planausschuss			planned scrap quantity	FR_MD_OPERATION	ERP_PLANNED_SCRAP_QUANTITY	FLOAT
AFVV-BMSCH, Einheit AFVV-MEINH	C (17)	BMSCH	Basismenge			piece time factor	FR_MD_OPERATION	TIME_PER_UNIT	NUMBER(19, 0)
AFVV-VGW01	C (17)	VGW01	Rüstzeit Maschine (Vorgabewert 1 in SAP, in SAP frei konfigurierbar d.h. muss keine Rüstzeit sein!)			standard value 1	FR_MD_OPERATION	STANDARD_VALUE_1 / STANDARD_VALUE_1_MS	VAR-CHAR2(255 CHAR)
AFVV-VGE01	C (3)	VGE01	Rüstzeiteinheit (Einheit von SAP Vorgabewert1)			standard unit 1	FR_MD_OPERATION	STANDARD_UNIT_1	VAR-CHAR2(255 CHAR)

Download Interface

AFVV-VGW02	C (17)	VGW02	Personal-rüstzeit (Vorgabewert 2 in SAP, in SAP frei konfigurierbar d.h. muss keine Personal-rüstzeit sein!)			standard value 2, piece time	FR_MD_OPERATION	TIME_PER_UNIT / STANDARD_VALUE_2 / STANDARD_VALUE_2_MS	NUMBER(19, 0) / VARCHAR2(255 CHAR)
AFVV-VGE02	C (3)	VGE02	Personal-rüstzeiteinheit (Einheit von SAP Vorgabewert3)			standard unit 2, piece time unit	FR_MD_OPERATION	TIME_PER_UNIT / STANDARD_UNIT_2	NUMBER(19, 0) / VARCHAR2(255 CHAR)
AFVV-VGW03	C (17)	VGW03	Bearbeitungszeit Maschine (Vorgabewert 3 in SAP, in SAP frei konfigurierbar d.h. muss keine Maschinenzeit sein!)			standard value 3	FR_MD_OPERATION	STANDARD_VALUE_3 / STANDARD_VALUE_3_MS	NUMBER(19, 0)
AFVV-VGE03	C (3)	VGE03	Maschinenzeiteinheit (Einheit von SAP Vorgabewert3)			standard unit 3	FR_MD_OPERATION	STANDARD_UNIT_3	VARCHAR2(255 CHAR)

Download Interface

AFVV-VGW04	C (17)	VGW04	Bearbeitungszeit Personal (Vorgabewert 4 in SAP, in SAP frei konfigurierbar d.h. muss keine Bearbeitungszeit Personal sein!)			standard value 4	FR_MD_OPERATION	STANDARD_VALUE_4 / STANDARD_VALUE_4_MS	VAR-CHAR2(255 CHAR)
AFVV-VGE04	C (3)	VGE04	Personalzeiteinheit (Einheit von SAP Vorgabewert4)			standard unit 4	FR_MD_OPERATION	STANDARD_UNIT_4	VAR-CHAR2(255 CHAR)
AFVV-VGW05	C (17)	VGW05	Leistungsart 5 (Vorgabewert 5 in SAP, in SAP frei konfigurierbar welcher Zeit diese entsprechen soll)			standard value 5	FR_MD_OPERATION	STANDARD_VALUE_5 / STANDARD_VALUE_5_MS	VAR-CHAR2(255 CHAR)
AFVV-VGE05	C (3)	VGE05	Zeiteinheit (Einheit von SAP Vorgabewert5)			standard unit 5	FR_MD_OPERATION	STANDARD_UNIT_5	VAR-CHAR2(255 CHAR)
AFVV-VGW06	C (17)	VGW06	Leistungsart 6 (Vorgabewert 6 in			standard value 6	FR_MD_OPERATION	STANDARD_VALUE_6 / STANDARD_VALUE_6_MS	VAR-CHAR2(255 CHAR)

Download Interface

			SAP, in SAP frei konfigurierbar welcher Zeit diese entsprechen soll)						
AFVV-VGE06	C (3)	VGE06	Zeiteinheit (Einheit von SAP Vorgabewert6)			standard unit 6	FR_MD_OPERATION	STANDARD_UNIT_6	VAR-CHAR2(255 CHAR)
CRHD-ARBPL (gelesen mit PLPO-ARBID) wird zurzeit nicht gefüllt. Wäre eigentlich der Arbeitsplatz aus dem Arbeitsplan (der die Vorlage für den AVO des Auftrages war). Kundenspezifisch, wird in Erweiterungsmethoden gefüllt	C (8)	ARBPLP	Planarbeitsplatz (Arbeitsplan ERP) – nicht relevante Information			planned workplace	FR_MD_OPERATION	PLANNED_WORKPLACE_ID	NUMBER(19, 0)
CRHD-ARBPL, mit der AFVC-ARBID wird der Arbeitsplatz (Bezeichnung) aus der Tabelle CRHD gelesen. Ist der echte Arbeitsplatz aus dem AVO	C (8)	ARBPLI	Ist - ERP – Workcenter (Kapazitätsgruppe oder Einzel-Equipment)			target workplace	FR_MD_OPERATION	TARGET_WORKPLACE_ID	NUMBER(19, 0)

Download Interface

CRHD-ARBPL (mit CRHS-OBJID_UP). Mit der AFVC-ARBID wird die Hierarchie Tabelle CRHS gelesen = Übergeordneter Arbeitsplatz aus der ERP-Hierarchie	C (8)	ARBPLG	Ist-Arbeitsplatzgruppe (übergeordnete Hierarchie aus dem ERP) – nicht relevante Information			workplace group	FR_MD_OPERATION	TARGET_WORKGROUP_ID	NUMBER(19,0)
AFVC-LOART	C (4)	LOART	Lohnart			wage type	FR_MD_OPERATION	WAGE_TYPE	VAR-CHAR2(255 CHAR)
AFVC-LOGRP	C (3)	LOGRP	Lohngruppe			wage group	FR_MD_OPERATION	WAGE_GROUP	VAR-CHAR2(255 CHAR)
TJ02T-TXT04 in Deutsch	C (80)	STATU	Auftragsstatus Text Z.B. FREI SSAP... (SAP Systemstatus, AVO)			erp status	FR_MD_OPERATION	ERP_STATUS	VAR-CHAR2(255 CHAR)
TJ02T-ISTAT (System Status Codes intern)	C (80)	STATU_ID	ERP Status_ID Z.B. I0002 (SAP System Status ID AVO)			erp status id	FR_MD_OPERATION	ERP_STATUS_IDS	VAR-CHAR2(255 CHAR)
TJ02T-TXT04 in der Sprache aus der Tabelle /FFMES/AUFTR	C (80)	STATU_LA	ERP Statustext (AVO) in der Auftragsprache (SAP System status)			erp status language	FR_MD_OPERATION	ERP_STATUS_LA	VAR-CHAR2(255 CHAR)

Download Interface

leer oder X, wird auf X gesetzt, wenn Status SPER gesetzt ist	C (1)	SPRKZ	Sperrkennzeichen			lock indicator	FR_MD_OPERATION	LOCK_INDICATOR	VAR-CHAR2(255 CHAR)
berechnet aus AFKO-GAMNG * AFPO-UEBTO / 100 in UOM AFVV-MEINH	C (17)	UELIT	Überlieferungstoleranz			overdelivery quantity	FR_MD_OPERATION	OVERDELIVERY_QUANTITY	FLOAT
Wenn AFPO-UEBTK = 'X' dann N, sonst Y	C (1)	UELIC	Überlieferungsscheck			overdelivery check	FR_MD_OPERATION	OVERDELIVERY_CHECK	NUMBER(1, 0)
berechnet aus AFKO-GAMNG * AFPO-UNTTO / 100, in UOM AFVV-MEINH	C (17)	UNLIT	Unterlieferungstoleranz			underdelivery quantity	FR_MD_OPERATION	UNDERDELIVERY_QUANTITY	FLOAT
Immer N	C (1)	UNLIC	Unterlieferungsscheck			underdelivery check	FR_MD_OPERATION	UNDERDELIVERY_CHECK	NUMBER(1, 0)
leer	C (1)	STORA	Stör AVO Kennzeichen			malfunction operation indicator	FR_MD_OPERATION	MALFUNCTION_OP_INDICATOR	VAR-CHAR2(255 CHAR)
leer	C (4)	STEUS	Steuerschlüssel			control key	FR_MD_OPERATION	CONTROL_KEY	VAR-CHAR2(255 CHAR)
leer	C (1)	RMDKZ	Kennzeichen Rückmeldung vorgesehen			confirmation	FR_MD_OPERATION	CONFIRMATION	VAR-CHAR2(255 CHAR)
leer	C (1)	BEDAV	Kennzeichen Nachbearbeitungsvorgang			rework operation indicator	FR_MD_OPERATION	REWORK_OPERATION_NUMBER	VAR-CHAR2(255 CHAR)

Download Interface

leer	C (4)	MVORN	Vorgangsnummer (Material)			operation material number	FR_MD_OPERATION	OPERATION_MATERIAL_NUMBER	VAR-CHAR2(255 CHAR)
AFPO-MATNR	C (18)	MATNR	Materialnummer			material number	FR_MD_MATERIAL	MATERIAL_NUMBER	VAR-CHAR2(255 CHAR)
MAKT-MAKTX oder AUFK-KTEXT	C (40)	MTEXT	Materialkurztext			material description	FR_MD_LITERAL_TEXT	TEXT	VAR-CHAR2(1024 CHAR)
leer	C (1)	PALKZ	Paletten-schein-ken-nung			pallet note identifier	FR_MD_OPERATION	PALLET_NOTE_IDENTIFIER	VAR-CHAR2(255 CHAR)
leer	C (2)	VASTA	Status Vorgänger AVO			status predecessor operation	FR_MD_OPERATION	STATUS_PREDECESSOR_OPERATION	VAR-CHAR2(255 CHAR)
Nur bei Serienfertigung verwendet, sonst immer leer	C (4)	VERID	Fertigungs-version			production version	FR_MD_OPERATION	PRODUCTION_VERSION	VAR-CHAR2(255 CHAR)
leer	C (22)	ZEINR	Dokumentnummer			document number	FR_MD_OPERATION	DOCUMENT_NUMBER	VAR-CHAR2(255 CHAR)
leer	C (3)	ZEIAR	Dokumenten-tart			document type	FR_MD_OPERATION	DOCUMENT_TYPE	VAR-CHAR2(255 CHAR)
leer	C (2)	ZEIVR	Dokumentversion			document version	FR_MD_OPERATION	DOCUMENT_VERSION	VAR-CHAR2(255 CHAR)
AFVC-PLNFL	C (6)	PLNFL	Folge			sequence	FR_MD_OPERATION	OPERATION_SEQUENCE	VAR-CHAR2(255 CHAR)
wenn AFVC-PLNFL = '000000' dann leer sonst 1	C (1)	FLGAT	Folgenart			sequence category	FR_MD_OPERATION	SEQUENCE_CATEGORY	VAR-CHAR2(255 CHAR)

Download Interface

TJ30T-TXT04 (in Deutsch)	C (40)	ASTTX	Anwender-status (Text), AVO			user status	FR_MD_OPERATION	USER_STATUS	VAR-CHAR2(255 CHAR)
leer	C (4)	EVORN	Einsprung AVO für alternative Folge			alternate operation number	FR_MD_OPERATION	ALTERNATE_OPERATION_NUMBER	VAR-CHAR2(255 CHAR)
leer	C (4)	RVORN	Rücksprung AVO für alternative Folge			confirmation operation number	FR_MD_OPERATION	CONFIRM_OPERATION_NUMBER	VAR-CHAR2(255 CHAR)
AUFK-AUART	C (4)	AUART	Auftragsart			order type	FR_MD_OPERATION	ORDER_TYPE	VAR-CHAR2(255 CHAR)
TJ30T-ESTAT	C (40)	ASTTX_ID	Anwender-status Codes (intern) z.B. E00001, AVO			-	-	-	-
TJ30T-TXT04 in Auftragssprache (aus der Tabelle /FFMES/AUFTR)	C (40)	ASTTX_LA	Anwender-status Text in der Auftragsprache, AVO			-	-	-	-

2.2.3.4 Langtexte zu AVOs

Diese Daten werden im IDOC Segment /FFMES/SAFOTX mit der **Satzart AFOTX** an das Subsystem übergeben, sofern dies in der Konfiguration so eingestellt wurde. Die Konfiguration wird über die Tabelle /FFMES/AUFTR mit dem Flag **AKTIV_TXT** eingestellt.

Die Datenstruktur ist in **Tabelle 6** aufgeführt.

Tabelle 6: Datenstruktur AVO Langtexte (Quelle SAP (ERP) und Ziel FORCAM FORCE™)

SAP (ERP)			Bezeichnung	FORCAM FORCE™					
				Pflichtfelder			Persistieren		
SAP-Feld	Typ (Länge)	IDOC XML SAP		Down-load	Lösch-satz	Interne XML Trans-formation	Tabelle	Spalte	Typ
AFOTX	C (5)	SART	Satzart (Hier: AFOTX)			-	-	-	-
S	C (1)	VAKZ	Verarbeitungs- kennzeichen (Immer S)			-	-	-	-
/FFMES/AUFTR- SPRAS	C (1)	SPRAS	Sprachenschlüs- sel aus der Ta- belle /FFMES/AUFTR			language (Hinweis: Wird nicht benötigt & befüllt, da vom XSL vor- gegeben)	-	-	
wird hochgezählt	N (4)	LFDNR	Laufende Num- mer			(Hinweis: wird nicht benötigt & befüllt, da die Texte konkateniert werden)	-	-	
Langtext zum AVO: Text ID = AVOT	C (80)	AFOTXT	AFO Text (Textzeile) (wird konkateniert)			operati- onText	FR_MD_LITE- RAL_TEXT	TEXT	VAR- CHAR2(1024 CHAR)

Download Interface

Textname = AFVC-MANDT + AFVC-AUFPL + AFVC-APLZL Textobjekt = AUFK									
--	--	--	--	--	--	--	--	--	--

2.2.3.5 AVO Komponenten

Diese Daten werden im IDOC Segment /FFMES/SAFOKO mit der **Satzart AFOKO** an das Subsystem übergeben, sofern dies in der Konfiguration so eingestellt wurde. Die Konfiguration wird über die Tabelle /FFMES/AUFTR mit dem Flag **AKTIV_KOM** eingestellt. Es werden standardmäßig keine Dummy-Baugruppen (RESB-DUMPS = X) und auch keine Textpositionen (TXTPS = X) erzeugt. Bei Kuppelprodukt wird ebenfalls kein Satz gesendet (RESB-KZKUP = X)

Die Datenstruktur ist in **Tabelle 7** aufgeführt.

Tabelle 7: Datenstruktur AVO Komponente (Quelle SAP (ERP) und Ziel FORCAM FORCE™)

SAP (ERP)			Bezeichnung	FORCAM FORCE™					
				Pflichtfelder			Persistieren		
SAP-Feld	Typ (Länge)	IDOC XML SAP		Down-load	Lösch-satz	Interne XML Trans-formation	Tabelle	Spalte	Typ
AFOKO	C (5)	SART	Satzart (Hier: AFOKO)			-	-	-	-
wenn RESB-XLOEK = 'X' dann D, sonst S	C (1)	VAKZ	Verarbeitungs-kennzeichen			-	-	-	-
AUFK-MANDT	C (3)	MANDT	Mandant			client	FR_MD_ERP_KEY	CLIENT	VAR-CHAR2(255 CHAR
AUFK-BUKRS	C (4)	BUKRS	Buchungskreis			com-panyCode	FR_MD_ERP_KEY	COMPANY_CODE	VAR-CHAR2(255 CHAR
AUFK-WERKS	C (4)	WERKS	Werk			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	Auftragsnum-mer			orderNum-ber	FR_MD_OPERA-TION	ORDER_NUMBER	VARCHAR2(32 CHAR)
Immer 0	C (1)	ASPLT	Auftragssplit (Immer 0, ist kein SAP Split)			orderSplit	FR_MD_OPERA-TION	ORDER_SPLIT	VARCHAR2(8 CHAR)

Download Interface

AFVC-RUECK	N (10)	RUECK	Rückmelde- nummer des Vorgangs			confirma- tionNumber	FR_MD_OPERA- TION	CONFIRMATION_NUM- BER	VAR- CHAR2(255 CHAR)
AFVC-VORNR	C (4)	VORNR	AVO			operation- Number	FR_MD_OPERA- TION	OPERATION_NUMBER	VARCHAR2(32 CHAR)
Immer 0	C (1)	VSPLT	AVO Split			operationS- plit	FR_MD_OPERA- TION	OPERATION_SPLIT	VARCHAR2(8 CHAR)
RESB-RSNUM	C (10)	RSNUM	Nummer der Reservierung / des Sekundär- bedarfs			reservation- Number		RESERVATION_NUMBER	VAR- CHAR2(255 CHAR)
RESB-RSPOS	C (4)	RSPOS	Positionsnum- mer der Reser- vierung / des Sekundärbe- darfs			position- Number		POSITION_NUMBER	VAR- CHAR2(255 CHAR)
RESB-MATNR	C (18)	MATNR	Komponenten- nummer			component- Number		COMPONENT_NUMBER	VAR- CHAR2(255 CHAR)
RESB-WERKS	C (4)	MWERK	Werk aus der Reservierung			reservation- Plant		RESERVATION_PLANT	VAR- CHAR2(255 CHAR)
RESB-LGORT	C (4)	LGORT	Lagerort			storageLoca- tion		STORAGE_LOCATION	VAR- CHAR2(255 CHAR)
RESB-CHARG	C (10)	CHARG	Charge			batchNum- ber		BATCH_NUMBER	VAR- CHAR2(255 CHAR)
RESB-SOBKZ	C (1)	SOBKZ	Sonderbe- standkennzei- chen			specialStock- Indicator		SPECIAL_STOCK_INDI- CATOR	VAR- CHAR2(255 CHAR)
RESB-BDTER	TT.MM.JJJJ	BDTER	Bedarfstermin der Kompo- nente			require- mentsDate		REQUIREMENTS_DATE	TI- MESTAMP(6)

Download Interface

RESB-BDMNG	C (15)	BDMNG	Bedarfsmenge			requirement-Quantity		REQUIREMENT_QUANTITY	FLOAT
RESB-MEINS	C (3)	MEINS	Basismengeneinheit			baseUnitOf-Measure		BASE_UNIT_OF_MEASURE	VAR-CHAR2(255 CHAR)
RESB-SERNR	C (8)	SERNR	Seriennummer			serialNumber		SERIAL_NUMBER	VAR-CHAR2(255 CHAR)
MAKT-MAKTX für RESB-MATNR in Sprache aus Tabelle /FFMES/AUFTR	C (40)	KTX01	Bezeichnung technisches Objekt (Artikeltext)			description		TEXT	VAR-CHAR2(1024 CHAR)
RESB-LGNUM	C (3)	LGNUM	Lagernummer / Lagerkomplex			warehouseNumber		WAREHOUSE_NUMBER	VAR-CHAR2(255 CHAR)
RESB-LGTYP	C (3)	LGTYP	Lagertyp			warehouse-Type		WAREHOUSE_TYPE	VAR-CHAR2(255 CHAR)
RESB-LGPLA	C (10)	LGPLA	Lagerplatz			storageBin		STORAGE_BIN	VAR-CHAR2(255 CHAR)
MARA-MTART	C (4)	MTART	Materialart			materialType		MATERIAL_TYPE	VAR-CHAR2(255 CHAR)
MARC-RGEKZ	C (1)	RGEKZ	Kennzeichen Retrograde Entnahme			retrograde-Withdrawal-indicator		RETROGRD_WITHDRAWAL_INDICATOR	VAR-CHAR2(255 CHAR)
MARC-SCHGT	C (1)	SCHGT	Kennzeichen Schüttgut			bulkMaterial-Indicator		BULK_MATERIAL_INDICATOR	VAR-CHAR2(255 CHAR)
MARA-MATKL	C (9)	MATKL	Warengruppe			material-Group		MATERIAL_GROUP	VAR-CHAR2(255 CHAR)
AFVC-APLFL	C (6)	PLNFL	Planfolge			-	-	-	-

2.2.3.6 Fertigungshilfsmittel

Diese Daten werden im IDOC Segment **/FFMES/SAUFFH** mit der **Satzart AUFFH** an das Subsystem übergeben, sofern dies in der Konfiguration so eingestellt wurde. Die Konfiguration wird über die Tabelle **/FFMES/AUFTR** mit dem **Flag AKTIV_FHM** eingestellt.

Die Datenstruktur ist in **Tabelle 8** aufgeführt.

Tabelle 8: Datenstruktur Fertigungshilfsmittel (Quelle SAP (ERP) und Ziel FORCAM FORCE™)

SAP (ERP)			Bezeichnung	FORCAM FORCE™					
				Pflichtfelder			Persistieren		
SAP-Feld	Typ (Länge)	IDOC XML SAP		Down-load	Lösch-satz	Interne XML Transforma-tion	Tabelle	Spalte	Typ
AUFFH	C (5)	SART	Satzart (Hier: AFOKO)			-	-	-	-
Wenn AFFH-LOEKZ = '' dann S, sonst D	C (1)	VAKZ	Verarbeitungs-kennzeichen			-	-	-	-
AUFK-MANDT	C (3)	MANDT	Mandant			client	FR_MD_ERP_KEY	CLIENT	VAR-CHAR2(255 CHAR
AUFK-BUKRS	C (4)	BUKRS	Buchungskreis			companyCode	FR_MD_ERP_KEY	COM-PANY_CODE	VAR-CHAR2(255 CHAR
AUFK-WERKS	C (4)	WERKS	Werk			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	Auftragsnum-mer			orderNumber	FR_MD_OPERATION	ORDER_NUM-BER	VARCHAR2(32 CHAR)
Immer 0	C (1)	ASPLT	Auftragssplit (Immer 0,			orderSplit	FR_MD_OPERATION	ORDER_SPLIT	VARCHAR2(8 CHAR)

Download Interface

			Split im FORCAM Sprachgebrauch kein SAP Split)						
AFVC-RUECK	N (10)	RUECK	Rückmeldenummer des Vorgangs			Wird nicht mehr gemappt			
AFVC-VORNR	C (4)	VORNR	AVO			operation-Number	FR_MD_OPERATION	OPERATION_NUMBER	VARCHAR2(32 CHAR)
Immer 0	C (1)	VSPLT	AVO Split			operationSplit	FR_MD_OPERATION	OPERATION_SPLIT	VARCHAR2(8 CHAR)
AFFH-PSNFH	N (4)	LFDNR	Laufende Nummer der Langtext Pos.			sequence	FR_MD_PRODUCTION_RE-SOURCE_TOOL	PRT_SEQUENCE	VARCHAR2(255 CHAR)
M (Material), D (Dokument), E (Equipment), S (ohne Material)	C (1)	FHART	Art des Fertigungshilfsmittels			type	FR_MD_PRODUCTION_RE-SOURCE_TOOL	PRT_TYPE	VARCHAR2(255 CHAR)
CRVM_A-MATNR oder CRVD_A-DOKNR oder CRVE_A-EQUNR oder CRVS_A-SFHNR	C (18)	MATNR	FHM Nummer			number	FR_MD_PRODUCTION_RE-SOURCE_TOOL	PRT_NUMBER	VARCHAR2(255 CHAR)
MAKT-MAKTX DRAT-DKTXT EQKT-EQKTX CRTX-KTEXT	C (40)	FHTXT	FHM Kurztext			description	FR_MD_LITERAL_TEXT	TEXT	VARCHAR2(1024 CHAR)
AFFH-MGVGW (N 9,3)	C (13)	MGVGW	FHM Menge			quantity	FR_MD_PRODUCTION_RE-SOURCE_TOOL	QUANTITY	FLOAT
AFFH-MGEINH	C (3)	MGEINH	Mengeneinheit für Anzeige			displayQuantityUnit	FR_MD_PRODUCTION_RE-SOURCE_TOOL	UNIT	VARCHAR2(255 CHAR)
CRFH-FGRU1	C (80)	STATU	Auftragsstatus			orderStatus	FR_MD_PRODUCTION_RE-SOURCE_TOOL	PRT_GROUP	VARCHAR2(255 CHAR)

Download Interface

AFVC-PLNFL	C (6)	PLNFL	Planfolge			-	-	-	-
------------	-------	-------	-----------	--	--	---	---	---	---

2.2.4 Maschinenkapazitäten / Schichtpläne

Schichtdaten für einen Arbeitsplatz mit n Pausen können mit dem Programm **/FFMES/TRANSFER_SHIFT_DATA** an das angebundene FORCAM FORCE™ übertragen werden. Es wird das Kapazitätsangebot der Arbeitsplätze selektiert.

Es werden neben der Selektion über den Zeithorizont zusätzliche Selektionen über Arbeitsplatz, Arbeitsplatzart, Subsystem (BDE Gruppe), Name des Ändernden und Datum der Änderung, Sperrkennzeichen und Löschvormerkung des Arbeitsplatzes angeboten. Es können auch Schichtkapazitäten übertragen werden, die den Fabrikkalender übersteuern (Kapazität an Tagen, die normalerweise keine Arbeitstage sind). Das Lesen eines Referenzangebotes für die Arbeitsplatzkapazität ist möglich.

Es wird ein Segment für die eigentlichen Schichtdaten erstellt (vgl. Datenstruktur **Tabelle 9**) und n Segmente für die dazugehörigen Pausen entsprechend der Schichtdefinition (vgl. Datenstruktur **Tabelle 10**).

Tabelle 9: Datenstruktur Schichtsegment (Quelle SAP (ERP) und Ziel FORCAM FORCE™)

SAP (ERP)			Bezeichnung	FORCAM FORCE™					
				Pflichtfelder			Persistieren		
SAP-Feld	Typ (Länge)	IDOC XML SAP		Down-load	Lösch-satz	Interne XML Transforma-tion	Tabelle	Spalte	Typ
AUFFH	C (5)	SART	Satzart (Hier: SCHIC)	X		-	-	-	-
S	C (1)	VAKZ	Verarbeitungs-kennzeichen (Immer S)	X		-	-	-	-
AUFK-MANDT	C (3)	MANDT	Mandant	X		client	FR_MD_ERP_KEY	CLIENT	VAR-CHAR2(255 CHAR
AUFK-BUKRS	C (4)	BUKRS	Buchungskreis	X		companyCode	FR_MD_ERP_KEY	COM-PANY_CODE	VAR-CHAR2(255 CHAR
AUFK-WERKS	C (4)	WERKS	Werk	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	Arbeitsplatz	X		workplace	FR_MD_SHIFT	WORK-PLACE_ID (normalized)	
Datum aus Selektionszeit-horizont	C (10) DD.MM.YYYY	VALID_FR	Gültig ab	X		shiftDate	FR_MD_SHIFT		
KAPA-TPROG	C (4)	KAPTPROG	Schicht-Typ	X		shiftTagCode	FR_MD_SHIFT		

Download Interface

			(Schichtprogramm (Schichtcode für Frühschicht, etc.)						
Montag, Dienstag, ..., Sonntag (ausgeschrieben in der jeweiligen Anmeldesprache) wird über FB ermittelt	C (10)	WEEKDAY	Wochentag						
TC37A-BEGZT	C (5) (HH:MM)	SHIFT_STR	Start	X		startTime	FR_MD_SHIFT	START_TS	
TC37A-ENDZT	C (5) (HH:MM)	SHIFT_END	Ende	X		endTime	FR_MD_SHIFT	END_TS	

Tabelle 10: Datenstruktur Pausensegment innerhalb der Schichtversorgung (Quelle SAP (ERP) und Ziel FORCAM FORCE™)

SAP (ERP)			Bezeichnung	FORCAM FORCE™					
				Pflichtfelder			Persistieren		
SAP-Feld	Typ (Länge)	IDOC XML SAP		Down-load	Lösch-satz	Interne XML Trans-formation	Tabelle	Spalte	Typ
	C (5)	SART	Satzart (Hier: PAUSE)			-	-	-	-
	C (1)	VAKZ	Verarbeitungs- kennzeichen (Immer I)			-	-	-	-
SY-MANDT	C (3)	MANDT	Mandant			client	FR_MD_ERP_KEY	CLIENT	VARCHAR2(255 CHAR
T001K-BUKRS	C (4)	BUKRS	Buchungskreis			companyCode	FR_MD_ERP_KEY	COM-PANY_CODE	VARCHAR2(255 CHAR
CRHD-WERKS	C (4)	WERKS	Werk			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	Arbeitsplatz			workplace	FR_MD_SHIFT	WORK-PLACE_ID (normalized)	
KAPA-TPROG	C (4)	KAPTPROG	Schicht-Typ			shiftTagCode	FR_MD_SHIFT		
TC37P-PAUNR	C (3)	AUFNR	Nummer der Pause			-	-	-	-
TC37P-PAU-BEG	C (5) HH:MM	AUFNR	Pausen Beginn	X		startTime			
TC37P-PAU-END	C (5) HH:MM	AUFNR	Pausen Ende	X		endTime			

2.2.5 Personaldaten (Ministammsätze Personal)

Personal Ministammsätze können mit dem Programm **/FFMES/TRANSFER_HR_DATA** an das angebundene FORCAM FORCE™ übertragen werden. Dabei ist eine Selektion über einen bestimmten Personalbereich möglich. Alle Personalnummern, die zu diesem Personalbereich gehören, werden in einem gemeinsamen IDOC übertragen. Eine Übertragung einzelner Personalnummern ist nicht möglich.

Es werden Daten aus den folgenden Infotypen verwendet:

- Infotyp 0000 Maßnahmen
- Infotyp 0001 Org. Zuordnung
- Infotyp 0002 Daten zur Person
- Infotyp 0050 Zeiterfassung

Es wird jeweils ein IDOC Segment für die Personaldaten einer Personalnummer erstellt.

Die Datenstruktur ist in **Tabelle 11** aufgeführt.

Tabelle 11: Datenstruktur Personaldatensegment (Quelle SAP (ERP) und Ziel FORCAM FORCE™)

SAP (ERP)			Bezeichnung	FORCAM FORCE™					
				Pflichtfelder			Persistieren		
SAP-Feld	Typ (Länge)	IDOC XML SAP		Down-load	Lösch-satz	Interne XML Transformation	Tabelle	Spalte	Typ
	C (5)	SART	Satzart (Hier: PERS)	X		-	-	-	-
	C (1)	VAKZ	Verarbeitungs-kennzeichen (Immer I)	X		-	-	-	-
PA0001-MANDT	C (3)	MANDT	Mandant	X		client	FR_MD_ERP_KEY	CLIENT	VARCHAR2(255 CHAR
PA0001-BUKRS	C (4)	BUKRS	Buchungskreis	X		companyCode	FR_MD_ERP_KEY	COMPANY_CODE	VARCHAR2(255 CHAR
PA0001-WERKS	C (4)	WERKS	Werk (Personal-bereich)	X		plant	FR_MD_ERP_KEY	PLANT	VARCHAR2(255 CHAR
SY-SYSID	C (8)	SYSID	SAP System ID	X		systemId	FR_MD_ERP_KEY	SYSTEM_ID	VARCHAR2(255 CHAR
PA0001-PERNR	C(8)	PERNR	Personalnum-mer	X		personnelNumber	FR_MD_PERSON	PERSONNEL_NUM-BER	VARCHAR2(255 CHAR)
PA0050-ZAUSW	C(8)	ZAUSW	Ausweisnummer	X		identificationNum-ber	FR_MD_PERSON	IDENTIFICA-TION_NUMBER	VARCHAR2(255 CHAR)
PA0002-VORNA	C(10)	VORNA	Vorname			firstName	FR_MD_PERSON	FIRST_NAME	VARCHAR2(255 CHAR)
PA0002-NACHN	C(20)	NACHN	Nachname			lastName	FR_MD_PERSON	LAST_NAME	VARCHAR2(255 CHAR)
PA0001-KOSTL	C(10)	KOSTL	Kostenstelle			costCenter	FR_MD_PERSON	COST_CENTER	VARCHAR2(255 CHAR)

3 Upload Interface

3.1 Allgemein

Mit Hilfe des Upload-Prozesses werden Rückmelde-Ereignisse (Ereignissesequenzen) und zugehörige strukturierte Daten von FORCAM FORCE™ an das ERP zurückgemeldet.

Die Ereignis-Trigger befinden sich im unterlagerten FORCAM FORCE™ -System. Für einen Rückmeldeprozess in ein ERP-System ist dort ein entsprechender Web-Service für die Kommunikation und Datentransfer bereitzustellen und anzuwenden.

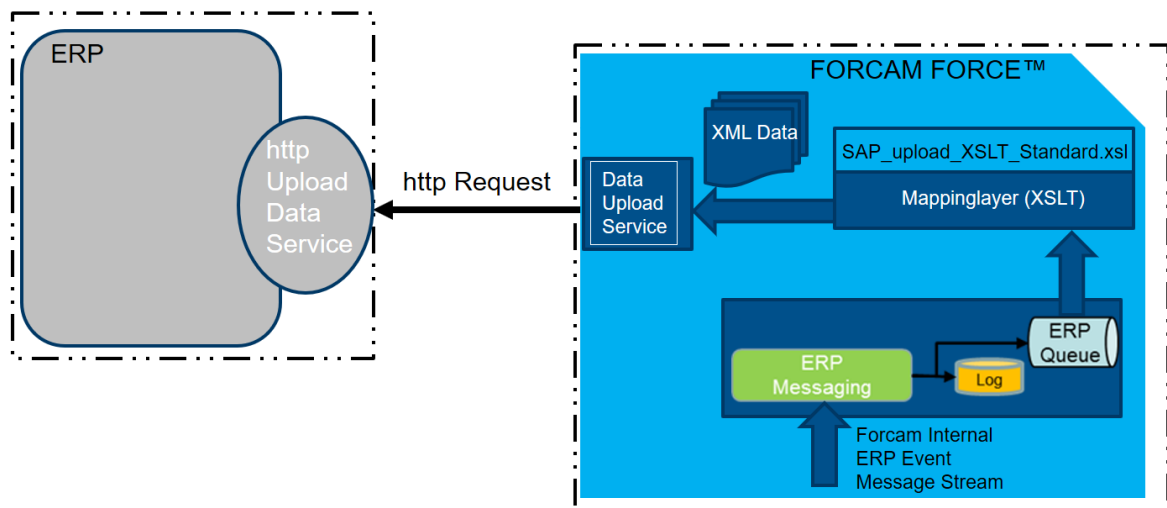


Bild 4: Allgemeine Data Upload Prinzip-Struktur (Quelle FORCAM FORCE™ und Ziel ERP)

In FORCAM FORCE™ selbst werden durch Trigger-Ereignisse wie (AVO–Start, AVO–Ende, Rüst-Start, Rüst-Ende, Korrekturen) ein „Forcam Internal ERP Message Stream“ erzeugt und mit entsprechenden Ereignisdaten angereichert. Über Rule-Engine Möglichkeiten können Ereignisse noch mit kundenspezifischen Berechnungsdaten („Kennwerte“) angereichert werden.

Die Ereignisse werden durch eine persistente Queue an den Upload-Service von der Runtime übermittelt. Ausschließlich erfolgreich an SAP übertragene Rückmeldungen werden zur Queue hin quittiert. Dadurch ist eine Ausfallsicherheit gewährleistet und fehlgeschlagene Übertragungen verbleiben in der Queue.

Vor der Übertragung wird durch ein XSLT Mapping Layer die Einzelereignisse aus dem „Forcam internal Message Stream“ mit Hilfe eines XSL Schema Files in die jeweiligen XML-Telegramme transformiert. Diese XML – Telegramme werden unter Verwendung eines ERP – Upload-Dienstes an das ERP gesendet. Der ERP – Upload-Dienst im FORCAM FORCE™ nutzt in diesem Fall den im ERP System bereitgestellten Data Service.

3.1.1 Anwendung Implementierte RFC Verbindung SAP (ERP)

Die Kommunikationsverbindung zwischen SAP (ERP) und FORCAM FORCE™ erfolgt in der aktuellen Implementierung und Ausprägung immer unter Verwendung einer expliziten RFC – Kommunikation. Den Data-Upload-Prozess für diesen Prozess zeigt **Bild 5**.

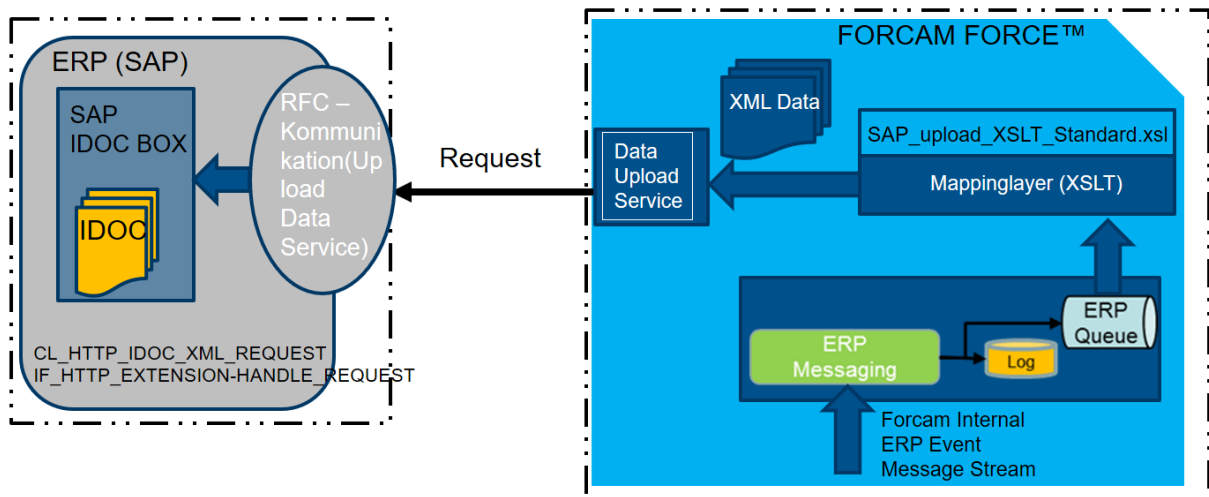


Bild 5: Upload-Verbindung von FORCAM FORCE™ und Ziel SAP (ERP)

Der Upload Data Service in SAP nutzt das IDOC Communication Framework unter Verwendung ALIAS (TRX SICF) mit den aktuellen Parametern

- HTTP Port über SMICM,
- Spring – Services (8000)
- System User (Kommunikation)

Der Data Upload Service von FORCAM FORCE™ meldet sich dem System User am SAP Gateway (Sicherer Anwendungsserver) an und das IDOC wird übergeben.

Jede logische Nachricht (IDOC Basistyp) ist über die Partnervereinbarung (Transaktion WE20) mit einem Rückmeldecode verbunden. Der Rückmeldecode wird dabei in SAP angelegt und mit einem Eingangsfunktionsbaustein verbunden, der die dafür eingehenden IDOCS verarbeitet.

3.2 FORCAM FORCE™ Rückmeldesatz

Alle Meldungen vom FORCAM FORCE™ werden in Form eines XML-Rückmeldesatzes erzeugt und an das überlagerte ERP zurückgemeldet.

Bei Verwendung von SAP werden alle Meldungen im IDOC Segment **/FFMES/SRUECK** übertragen. In der **Tabelle /FFMES/VERTEILER** wird das entsprechende Verarbeitungsprogramm gefunden. Zur Unterscheidung / Definition der verschiedenen Meldungen wird die Satzart benutzt (**Feld SART**).

Die Datenstruktur ist in **Tabelle 12** aufgeführt.

Tabelle 12: Datenstruktur Rückmeldesatz allgemein für alle Nachrichtentypen (Quelle FORCAM FORCE™ und SAP (ERP))

Bezeichnung	Typ	Länge	XML
Satzart	C	5	SART
Mandant	C	3	MANDT
Buchungskreis	C	4	BUKRS
Werk	C	4	WERKS
Rückmeldenummer	C	10	RUECK
Auftragsnummer	C	12	AUFNR
AVO	C	4	VORNR
Auftragssplit	C	3	ASPLT
Arbeitsplatz	C	8	ARBPL
Gutmenge	C	14	LMNGA
Ausschuss	C	14	XMNGA
Nacharbeit	C	14	RMNGA
Mengeneinheit	C	3	MEINH
Datum der Meldung	C	10	DATUM
Uhrzeit der Meldung	C	6	EZEIT
Personalnummer	C	8	PERNR
Zeitausweis	C	8	ZAUSW
Zeitfaktor	C	6	PROPZ
Pausenkennzeichen	C	1	PAUSE
Auftragsstatus	C	2	AUSTA
Maschinenstatus	C	3	MASTA
Zu buchende Leistungsart	C	30	LSTAR
Sollstart Datum	C	10	SOLLS
Sollende Datum	C	10	SOLLE

Upload Interface

Lohnart	C	4	LOARR
Materialnummer	C	18	MATNR
Kostenstelle	C	10	KOSTL
Teamnummer	C	10	ZTEAM
Schicht (S oder V)	C	1	ZFLAG
Kostenrechnungstyp	C	1	KOTYP
Kontierungsobjekt	C	25	KOOBJ
Arbeitsplatztyp	C	2	APTYP
Zu buchende Dauer	C	5	DAUER
Maschinentaktzeit	C	14	VGWMZ
Sollrüstzeit	C	14	VGWRZ
Schichttag	C	10	SCHIT
Schichtkennung	C	1	SCHIK
Kostenstelle Maschine	C	10	KOSMA
Produktivzeit	C	5	PZEIT
GUID / Interne Nummer	C	32	INTNR
Ausfallgrund	C	4	AGRUN
Ausfallverursacher	C	10	AVERU
Nacharbeitsgrund	C	4	NGRUN
Nacharbeits-verursacher	C	10	NVERU
Co Leistungsart	C	6	COLAR
Personalobjekttyp	C	2	PEOBJ
GUID der zu korrigierenden Meldung	C	32	KORID
Korrekturtyp	C	1	KORTY
Datum der Korrektur	C	8	KORDT

Upload Interface

Uhrzeit der Korrektur	C	6	KORZT
Satzart der zu korrigierenden Meldung	C	5	KORSA
Zeitstempel	C	6	TSTMS
Erster Faktor	C	14	FAK01
Zweiter Faktor	C	14	FAK02
Dritter Faktor	C	14	FAK03
Zuschlagsminuten	C	5	ZUMIN
Ausschussgrund	C	4	ZFSL1
Werk	C	4	ZBWK1
Kostenstelle	C	10	ZBKS1
Arbeitsplatz	C	8	ZBAP1
Materialnummer Komponente	C	18	KOMPO
Terminal ID	C	4	TERID
Rückmeldetext	C	40	TEXT1
Version	C	4	VERID
Nummer der Reservierung	C	10	RSNUM
Reservierungsposition	C	4	RSPOS
Charge	C	10	CHARG
Palette	C	3	PALET
Lieferscheindatum	C	10	LSDAT
Gewicht	C	14	GWCHT

Von FORCAM FORCE™ werden mit jeder Satzart die notwendigen Felder an SAP gesendet, die für die jeweilige Buchung erforderlich sind. In den meisten Fällen wird mit Hilfe der im IDOC gesendeten Daten eine Rückmeldung auf einen Fertigungsauftrag in SAP gebucht.

Für die einzelnen Meldungen sind folgende Felder im Standard gefüllt und in Abschnitt 3.4 in Tabelle 12 beschrieben.

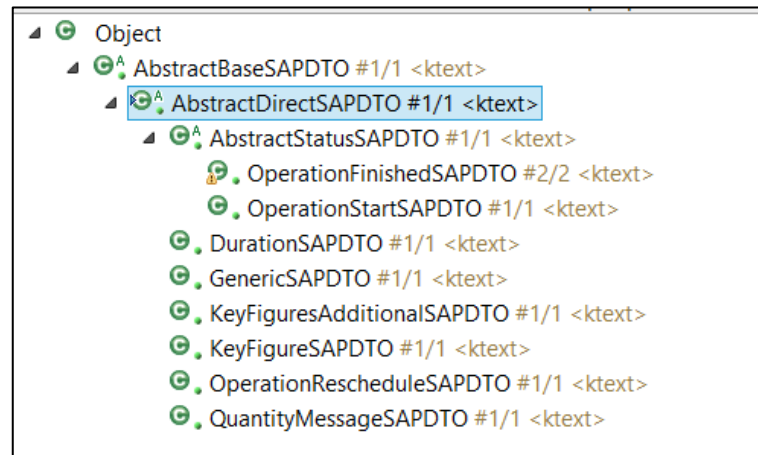
3.3 Rückmeldeereignisse

In der aktuellen Forcam – SAP – ausgeprägten ERP-Schnittstelle kommen folgende Rückmeldeereignisse zur Anwendung:

- Status – Ereignismeldungen
 - AVO Start, AVO - Ende
- Dauer – Ereignismeldungen
 - Werden in FORCAM FORCE™ nur bei AVO – Ende oder Schichtende (AVO läuft Schicht-übergreifend) generiert
 - Pro Leistungsart: Maschinenproduktionszeit, Maschinerüstzeit
 - Bei Personalzeit: Pro Person und Leistungsart
- Mengenmeldungen
 - Asynchron oder Einzel & | | gesteuert nach Intervallen
 - Pro Mengentyp eine separate Meldung
 - Gutmenge
 - Ausschussmenge
 - Nacharbeitsmenge
- Korrekturen
 - Delete / Insert
 - Restriktionen sind gegenüber NONE SAP im Detail zu spezifizieren und muss durch das ERP unterstützt werden
 - Nur möglich mit eindeutiger Meldungsnummer
- Kennzahlen
 - Sind nach Schnittstellenausprägung ERP zu definieren
 - Z.B. Ist-Stückzeit Berechnung → Rule Engine Anwendung
- Ad-hoc-Wartung über das Shop Floor Terminal
 - Ad-hoc-Wartung anfordern (Request Maintenance)
 - Ad-hoc-Wartung Start (Start Maintenance)
 - Ad-hoc-Wartung beendet (Finish Maintenance)

Die Datenstrukturen der Einzelmeldeereignisse werden nach dem Objekt & Vererbungsgedanken aufgebaut und angereichert. Jedes Ereignis enthält generischen Standardinhalt und entsprechend weitere Vererbungshierarchien sowie einen spezifischen Inhalt.

Die Hierarchie der SAP-Nachrichten-Typen ist nachfolgend dargestellt.



3.3.1 Mappinglayer XSLT (XSL Schema Produktstandard)

Für die Transformation der Daten im Mappinglayer wird eine XSL Datei als Schema für den Produktstandard der Rückmeldung an SAP verwendet (Details siehe hierzu **Abschnitt 4**).

3.4 Gefüllte Tags im SAP-IDoc abhängig von der Satz-Art

Tabelle 13: Gefüllte XML-Tag-Elemente des Rückmeldesatzes abhängig von der gesendeten Meldungsart

Tag Name	Bedeutung	Start-Mel- dung (OP- STR)	Ende Mel- dung (O- PEND)	Mengen- meldun- gen (QTYMG)	Zeit/Dauer- Meldung (DURAT)	Kor- rektur lö- schen (RE- VMG)	Korrektur Ein- fügen (RE- VMG)	War- tungs- Anfor- derung	War- tungs- Start	War- tungs- Ende
<EDI_DC40>	Kopfsatz									
<MANDT>003</MANDT>	Mandant	X	X	X	X	X	X	X	X	X
<DOCNUM>- 1859907614639058460</DOCNUM>	Dokumenten- nummer	X	X	X	X	X	X	X	X	X
<DIRECT>2</DIRECT>	Richtung (Up- load = 2)	X	X	X	X	X	X	X	X	X
<IDOCTYP>/FFMES/R</IDOCTYP>	IDOC-Typ	X	X	X	X	X	X	X	X	X
<MESTYP>/FFMES/MESSAGE</MES- TYP>	Meldungstyp	X	X	X	X	X	X	X	X	X
<SNDPOR>FORCAMFF</SNDPOR>	Absender - Port = Port des FORCAM FORCE™ Sys- tems über den	X	X	X	X	X	X	X	X	X

Upload Interface

	die Daten gesendet werden									
<SNDPRT>LS</SNDPRT>	Absender-partner (Art) = Logisches System	X	X	X	X	X	X	X	X	X
<SNDPRN>FORCAMFF</SNDPRN>	Absender Partner Nummer = Logisches System des Absenders (FORCAM FORCE™)	X	X	X	X	X	X	X	X	X
<RCVPOR>SAPPU1</RCVPOR>	Empfänger-Port = Port des SAP Systems über den die Daten empfangen werden	X	X	X	X	X	X	X	X	X
<RCVPRT>LS</RCVPRT>	Empfänger-partner (Art) = Logisches System	X	X	X	X	X	X	X	X	X
<RCVPRN>SUB_ALE</RCVPRN>	Empfänger Partner Nummer = Logisches System des Empfängers (SAP)	X	X	X	X	X	X	X	X	X
<SE-RIAL>00161321360000000007</SE-RIAL>	Serialisierungsnummer	X	X	X	X	X	X	X	X	X

Upload Interface

<_FFMES_-SRUECK SEGMENT="1">	Meldedaten									
<SART>QTYMG</SART>	Satzart	OPSTR	OPEND	QTYMG	DURAT	REVMG	REVMG	MTREQ	MTSTA	MTFIN
<MANDT>003</MANDT>	Mandant	X	X	X	X	X	X	X	X	X
<BUKRS>7500</BUKRS>	Buchkreis	X	X	X	X	X	X	X	X	X
<WERKS>7500</WERKS>	Werk	X	X	X	X	X	X	X	X	X
<SYSID>600</SYSID>	System ID	X	X	X	X	X	X	X	X	X
<RUECK>0016132136</RUECK>	AVO-Rückmel- denummer	X	X	X	X	X	X	X	X	X
<AUFNR>008500013320</AUFNR>	Auftrags-Num- mer	X	X	X	X	X***	X***			
<VORNR>0010</VORNR>	AVO-Nummer	X	X	X	X	X***	X***			
<ASPLT>0</ASPLT>	Auftrags-Split	X	X	X	X	X***	X***			
<ARBPL>84915002</ARBPL>	Arbeitsplatz	X	X	X	X	X	X			
<DATUM>04.05.2015</DATUM>	Melde-Datum	X	X	X	X					
<EZEIT>160019</EZEIT>	Melde-Uhrzeit	X	X	X	X					
<SCHIT>2015-05-04</SCHIT>	Schichtdatum (Schichttag)	X	X	X	X		X			
<SCHIK>F</SCHIK>	Schichtkenn- zeichen	X	X	X	X		X*			
<INTNR>E6276E40F26511E49 543C30B0AD90108</INTNR>	Korrektur- GUID	X	X	X	X		X*			

Upload Interface

<AUSTA></AUSTA>	Auftragsstatus(Phasen-code)	X	X							
<PERNR>00200038</PERNR>	Personalnummer	X	X	X	X		X****			
<LSTAR>2</LSTAR>	Leistungsart				X		X*			
<DAUER>30.5</DAUER>	Dauer (Zeit)				X		X*			
<ZFSL1></ZFSL1>										
<LMNGA>16.0</LMNGA>	Gutmenge			X			X**			
<XMNGA>0.0</XMNGA>	Ausschussmenge			X			X**			
<AGRUN>15</AGRUN>	Ausschussgrund			X			X**			
<RMNGA>0.0</RMNGA>	Nacharbeitsmenge			X			X**			
<NGRUN>15</NGRUN>	Nacharbeitsgrund			X			X**			
<MEINH>SZT</MEINH>	Mengeneinheit			X			X**			
<MATNR>53031208520</MATNR>	Materialnummer			X			X**			
<KORID></KORID>	Korrektur-GUID (GUID der zu korrigierenden Meldung)									
<KORTY></KORTY>	Korrektur-Typ (Delete, Insert)					D	I			

Upload Interface

<KORDT></KORDT>	Korrektur-Datum						X			
<KORZT></KORZT>	Korrektur-Zeit						X			
<KORSA></KORSA>	Korrektur-Satzart						X			
<CHARG></CHARG>	Chargennummer									
<NOTIFICATION_TYPE></NOTIFICATION_TYPE>	Meldungs-Typ (für Wartung)							X		
<EQUIPMENT></EQUIPMENT>	Maschinen-Nummer							X		
<REPORTER_ID></REPORTER_ID>	gemeldet von (Personalnummer)							X		
<MALFUNCTION_COMMENT></MALFUNCTION_COMMENT>	Meldungs-Text (für Wartung)							X		
<MALFUNCTION_START_DATE></MALFUNCTION_START_DATE>	Start (Datum) der Störung							X		
<MALFUNCTION_START_TIME></MALFUNCTION_START_TIME>	Start (Uhrzeit) der Störung							X		
<MALFUNCTION_CODE></MALFUNCTION_CODE>	Fehlercode							X		
<MESPMID></MESPMID>	Referenznummer der Wartungsmeldung							X	X	X

Upload Interface

<MATERIAL_NUMBER></MATERIAL_NUMBER>	Produktnummer des AVOs							X		
<MATERIAL_DESCRIPTION></MATERIAL_DESCRIPTION>	Material-Beschreibung in Standard-Sprache							X		
<MAINTENANCE_START_DATE></MAINTENANCE_START_DATE>	Start (Datum) der Wartung								X	
<MAINTENANCE_START_TIME></MAINTENANCE_START_TIME>	Start (Uhrzeit) der Wartung								X	
<MALFUNCTION_END_DATE></MALFUNCTION_END_DATE>	Ende (Datum) der Wartung									X
<MALFUNCTION_END_TIME></MALFUNCTION_END_TIME>	Ende (Uhrzeit) der Wartung									X

(* = befüllt bei DURAT Korrektur, ** befüllt bei Mengenkorrektur, *** befüllt bei Auftragsbezug, **** befüllt bei Personenbezug)

- i** Korrekturmeldungen werden nur für Mengenmeldungen und Dauermeldungen erzeugt. Bei Lösch-Meldung wird der zu löschende Satz über seine GUID referenziert. Zusätzlich wird seine Satzart übermittelt. Einfüge-Meldung enthält die für den Einfüge-Typ notwendigen Felder.

3.5 Beispiel einer Mengenmeldung

```
<?xml version="1.0" encoding="UTF-8"?>
<_FFMES_R>
  <IDOC BEGIN="1">
    <EDI_DC40 SEGMENT="1">
      <TABNAM>EDI_DC40</TABNAM>
      <MANDT>003</MANDT>
      <DOCNUM>-1859907614639058460</DOCNUM>
      <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>
      <SNDPRT>LS</SNDPRT>
      <SNDPFC />
      <SNDPRN>FORCAMFF</SNDPRN>
      <SNDSAD />
      <SNDLAD />
      <RCVPOR>SAPPU1</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 Serialisierung und Korrekturfähigkeit

Eine eindeutige Serialisierung ist notwendig um Sequenzreihenfolgen und eindeutige Zuordnungen von Meldungen zu gewährleisten.

Korrekturfähigkeit kann nur sichergestellt werden, wenn jede einzelne Meldung mit einer überall eindeutigen Nachrichtennummer versehen ist und das übergeordnete ERP ebenfalls Stornierungsprozesse (Delete / Insert von Nachrichten) zulässt. Ein nicht SAP – Korrekturprozess muss für jedes ERP vollständig spezifiziert werden.

3.6.1 Arbeitsplatz - Serialisierung

Der SAP Standard bietet eine 10-stellige Nummer = 4 Stellen Kanalnummer + 6 Stellen Kanalzähler

- 4 Stellen Kanalnummer => 9999 Arbeitsplätze
- 6 Stellen Kanalzähler => Zähler zur Serialisierung der Meldungen bis 999999 Meldungen

Die Kanalnummern werden in der Tabelle BDRGIN verwaltet (SAP Standardtabelle). FORCAM FORCE™ führt einen Kanal pro Arbeitsplatz. Bei einem FORCAM FORCE™ -(FFRuntime)-Systemstart werden per SOAP-Abfrage die aktuellen Kanalzählerstände aus SAP erfragt zur Initialisierung der APL-Serialisierung. Für diesen Abgleich wird ein Funktionsbaustein im SAP aufgerufen, der die aktuellen Zählerstände aus der Tabelle BDRGIN zurückliefert.

⚠ Diese Art der Serialisierung sollte, wenn möglich, nicht mehr angewendet werden. Sobald für diesen Arbeitsplatz in der Verbuchungskette ein Fehler auftritt, stauen sich alle zu verbuchenden Meldungen auch für unterschiedliche AVOs auf einem Arbeitsplatz auf.

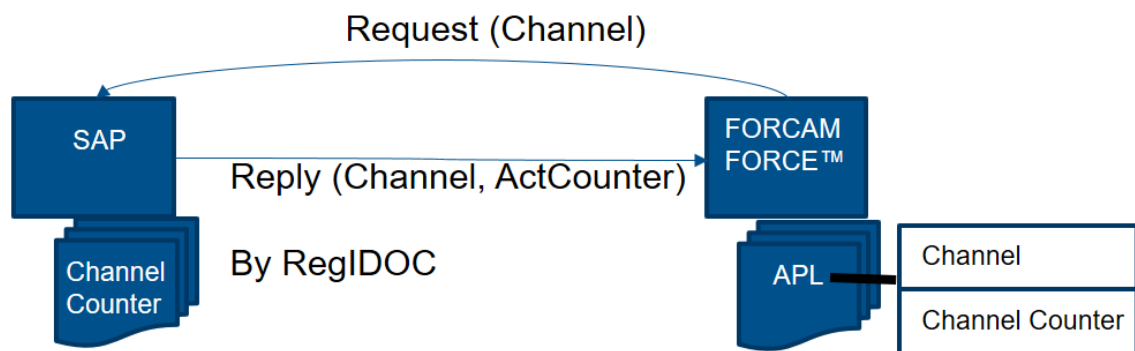


Bild 6: Prinzip Prozess der Arbeitsplatz-Serialisierung

3.6.2 AVO - Serialisierung

Keine Synchronisation mit SAP Kanalregistratur Tabelle (BDRGIN), sondern es wird ein Counter pro Rückmeldenummer in FORCAM FORCE™ in einer DB-Tabelle geführt. Im SAP ist in der Tabelle /FFMES/CONTROL der Parameter CHECK_IDOC_PRED aktiv.

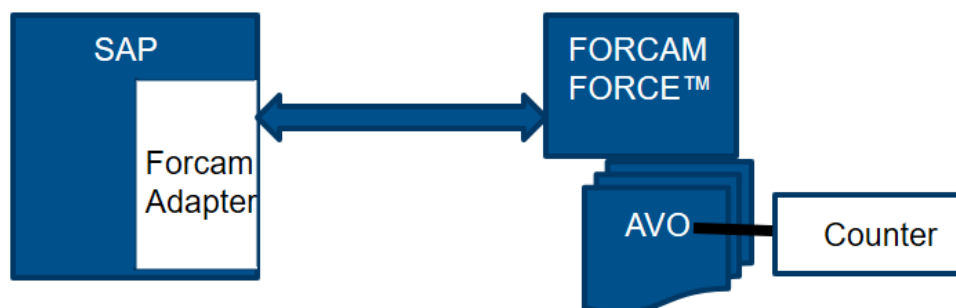


Bild 7: Prinzip Prozess der AVO - Serialisierung

4 Anhang

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 sel-
ect="EDI_DC40_MANDT"/></MANDT>

              </xsl:when>
              <xsl:otherwise>
                <MANDT></MANDT>
              </xsl:otherwise>
            </xsl:choose>

            <xsl:choose>
              <xsl:when test="DOCNUM">
                <DOCNUM><xsl:value-of sel-
ect="DOCNUM"/></DOCNUM>

                </xsl:when>
                <xsl:otherwise>
                  <DOCNUM></DOCNUM>
                </xsl:otherwise>
              </xsl:choose>

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

              <xsl:choose>
                <xsl:when test="IDOCTYP">
                  <IDOCTYP><xsl:value-of sel-
ect="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 select="SND-
POR"/></SNDPOR>

                                </xsl:when>
                                <xsl:otherwise>
                                    <SNDPOR></SNDPOR>
                                </xsl:otherwise>
                            </xsl:choose>

                            <xsl:choose>
                                <xsl:when test="SNDPRT">
                                    <SNDPRT><xsl:value-of select="SND-
PRT"/></SNDPRT>

                                </xsl:when>
                                <xsl:otherwise>
                                    <SNDPRT></SNDPRT>
                                </xsl:otherwise>
                            </xsl:choose>

                            <SNDPFC></SNDPFC>

                            <xsl:choose>
                                <xsl:when test="SNDPRN">
                                    <SNDPRN><xsl:value-of select="SND-
PRN"/></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 sel-
ect="RCVPOR"/></RCVPOR>

                                    </xsl:when>
                                    <xsl:otherwise>
                                        <RCVPOR></RCVPOR>
                                    </xsl:otherwise>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="RCVPRT">
                                        <RCVPRT><xsl:value-of sel-
ect="RCVPRT"/></RCVPRT>

                                        </xsl:when>
                                        <xsl:otherwise>
                                            <RCVPRT></RCVPRT>
                                        </xsl:otherwise>
                                    </xsl:choose>

                                <RCVPFC></RCVPFC>

                                <xsl:choose>
                                    <xsl:when test="RCVPRN">
                                        <RCVPRN><xsl:value-of sel-
ect="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 sel-
ect="RUECK_MANDT"/></MANDT>

        </xsl:when>
        <xsl:otherwise>
            <MANDT></MANDT>
        </xsl:otherwise>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="BUKRS">
            <BUKRS><xsl:value-of sel-
ect="BUKRS"/></BUKRS>

        </xsl:when>
        <xsl:otherwise>
            <BUKRS></BUKRS>
        </xsl:otherwise>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="WERKS">
            <WERKS><xsl:value-of sel-
ect="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>

ect="RUECK"/></RUECK>

<xsl:choose>
  <xsl:when test="RUECK">
    <RUECK><xsl:value-of sel-
ect="RUECK"/></RUECK>

  </xsl:when>
  <xsl:otherwise>
    <RUECK></RUECK>
  </xsl:otherwise>
</xsl:choose>

<xsl:choose>
  <xsl:when test="AUFNR">
    <AUFNR><xsl:value-of sel-
ect="AUFNR"/></AUFNR>

  </xsl:when>
  <xsl:otherwise>
    <AUFNR></AUFNR>
  </xsl:otherwise>
</xsl:choose>

<xsl:choose>
  <xsl:when test="VORNR">
    <VORNR><xsl:value-of sel-
ect="VORNR"/></VORNR>

  </xsl:when>
  <xsl:otherwise>
    <VORNR></VORNR>
  </xsl:otherwise>
</xsl:choose>

<xsl:choose>
  <xsl:when test="ASPLT">
    <ASPLT><xsl:value-of select="ASPLT"/></AS-
PLT>

  </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 select="EZEIT"/></E-
ZEIT>
                            </xsl:when>
                            <xsl:otherwise>
                                <EZEIT></EZEIT>
                            </xsl:otherwise>
                        </xsl:choose>

                        <xsl:choose>
                            <xsl:when test="SCHIT">
                                <SCHIT><xsl:value-of sel-
ect="SCHIT"/></SCHIT>
                            </xsl:when>
                            <xsl:otherwise>
                                <SCHIT></SCHIT>
                            </xsl:otherwise>
                        </xsl:choose>

                        <xsl:choose>
                            <xsl:when test="SCHIK">
                                <SCHIK><xsl:value-of sel-
ect="SCHIK"/></SCHIK>
                            </xsl:when>
                            <xsl:otherwise>
                                <SCHIK></SCHIK>
                            </xsl:otherwise>
                        </xsl:choose>

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

```

```

ect="KOSMA"/></KOSMA>
                                <KOSMA><xsl:value-of sel-
                                </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                <xsl:when test="PZEIT">
                                <PZEIT><xsl:value-of sel-
ect="PZEIT"/></PZEIT>
                                </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                <xsl:when test="INTNR">
                                <INTNR><xsl:value-of sel-
ect="INTNR"/></INTNR>
                                </xsl:when>
                                <xsl:otherwise>
                                <INTNR></INTNR>
                                </xsl:otherwise>
                                </xsl:choose>

                                <xsl:choose>
                                <xsl:when test="AGRUN">
                                <AGRUN><xsl:value-of select="AG-
RUN"/></AGRUN>
                                </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                <xsl:when test="AVERU">
                                <AVERU><xsl:value-of sel-
ect="AVERU"/></AVERU>
                                </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                <xsl:when test="NGRUN">
                                <NGRUN><xsl:value-of sel-
ect="NGRUN"/></NGRUN>
                                </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                <xsl:when test="NVERU">
                                <NVERU><xsl:value-of sel-
ect="NVERU"/></NVERU>
                                </xsl:when>
                                </xsl:choose>

```

```

<xsl:choose>
  <xsl:when test="COLAR">
    <COLAR><xsl:value-of sel-
ect="COLAR"/></COLAR>
  </xsl:when>
</xsl:choose>

<xsl:choose>
  <xsl:when test="PEOBJ">
    <PEOBJ><xsl:value-of select="PEOBJ"/></PE-
OBJ>
  </xsl:when>
</xsl:choose>

<xsl:choose>
  <xsl:when test="AUSTA">
    <AUSTA><xsl:value-of sel-
ect="AUSTA"/></AUSTA>
  </xsl:when>
  <xsl:otherwise>
    <AUSTA></AUSTA>
  </xsl:otherwise>
</xsl:choose>

<xsl:choose>
  <xsl:when test="MASTA">
    <MASTA><xsl:value-of sel-
ect="MASTA"/></MASTA>
  </xsl:when>
</xsl:choose>

<xsl:choose>
  <xsl:when test="PERNR">
    <PERNR><xsl:value-of sel-
ect="PERNR"/></PERNR>
  </xsl:when>
  <xsl:otherwise>
    <PERNR></PERNR>
  </xsl:otherwise>
</xsl:choose>

<xsl:choose>
  <xsl:when test="ZAUSW">
    <ZAUSW><xsl:value-of sel-
ect="ZAUSW"/></ZAUSW>
  </xsl:when>
</xsl:choose>

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

```

```

ect="PROPZ"/></PROPZ>
                                <PROPZ><xsl:value-of sel-
                                </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                <xsl:when test="PAUSE">
                                <PAUSE><xsl:value-of sel-
ect="PAUSE"/></PAUSE>
                                </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                <xsl:when test="LSTAR">
                                <LSTAR><xsl:value-of sel-
ect="LSTAR"/></LSTAR>
                                </xsl:when>
                                <xsl:otherwise>
                                <LSTAR></LSTAR>
                                </xsl:otherwise>
                                </xsl:choose>

                                <xsl:choose>
                                <xsl:when test="SOLLS">
                                <SOLLS><xsl:value-of sel-
ect="SOLLS"/></SOLLS>
                                </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                <xsl:when test="SOLLE">
                                <SOLLE><xsl:value-of sel-
ect="SOLLE"/></SOLLE>
                                </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                <xsl:when test="LOARR">
                                <LOARR><xsl:value-of sel-
ect="LOARR"/></LOARR>
                                </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                <xsl:when test="DAUER">
                                <DAUER><xsl:value-of sel-
ect="DAUER"/></DAUER>
                                </xsl:when>
                                <xsl:otherwise>
                                <DAUER></DAUER>

```

```

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

                                <xsl:choose>
                                    <xsl:when test="ZFSL1">
                                        <ZFSL1><xsl:value-of sel-
ect="ZFSL1"/></ZFSL1>

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

                                <xsl:choose>
                                    <xsl:when test="LMNGA">
                                        <LMNGA><xsl:value-of sel-
ect="LMNGA"/></LMNGA>

                                    </xsl:when>
                                    <xsl:otherwise>
                                        <LMNGA></LMNGA>
                                    </xsl:otherwise>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="XMNGA">
                                        <XMNGA><xsl:value-of sel-
ect="XMNGA"/></XMNGA>

                                    </xsl:when>
                                    <xsl:otherwise>
                                        <XMNGA></XMNGA>
                                    </xsl:otherwise>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="RMNGA">
                                        <RMNGA><xsl:value-of sel-
ect="RMNGA"/></RMNGA>

                                    </xsl:when>
                                    <xsl:otherwise>
                                        <RMNGA></RMNGA>
                                    </xsl:otherwise>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="MEINH">
                                        <MEINH><xsl:value-of sel-
ect="MEINH"/></MEINH>

                                    </xsl:when>
                                    <xsl:otherwise>
                                        <MEINH></MEINH>

```



```

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

    <xsl:choose>
        <xsl:when test="MATNR">
            <MATNR><xsl:value-of sel-
ect="MATNR"/></MATNR>

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

    <xsl:choose>
        <xsl:when test="KOSTL">
            <KOSTL><xsl:value-of sel-
ect="KOSTL"/></KOSTL>

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

    <xsl:choose>
        <xsl:when test="ZTEAM">
            <ZTEAM><xsl:value-of sel-
ect="ZTEAM"/></ZTEAM>

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

    <xsl:choose>
        <xsl:when test="ZFLAG">
            <ZFLAG><xsl:value-of sel-
ect="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 sel-
ect="KOOBJ"/></KOOBJ>

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

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

```

```

                                <APTYP><xsl:value-of select="APTYP"/></AP-
TYP>

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

                                <xsl:choose>
                                    <xsl:when test="VGWMZ">
                                        <VGWMZ><xsl:value-of sel-
ect="VGWMZ"/></VGWMZ>

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

                                    <xsl:choose>
                                        <xsl:when test="VGWRZ">
                                            <VGWRZ><xsl:value-of sel-
ect="VGWRZ"/></VGWRZ>

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

                                        <xsl:choose>
                                            <xsl:when test="KORID">
                                                <KORID><xsl:value-of select="KORID"/></KO-
RID>

                                                </xsl:when>
                                                <xsl:otherwise>
                                                    <KORID></KORID>
                                                </xsl:otherwise>
                                            </xsl:choose>

                                        <xsl:choose>
                                            <xsl:when test="KORTY">
                                                <KORTY><xsl:value-of sel-
ect="KORTY"/></KORTY>

                                                </xsl:when>
                                                <xsl:otherwise>
                                                    <KORTY></KORTY>
                                                </xsl:otherwise>
                                            </xsl:choose>

                                        <xsl:choose>
                                            <xsl:when test="KORDT">
                                                <KORDT><xsl:value-of sel-
ect="KORDT"/></KORDT>

                                                </xsl:when>
                                                <xsl:otherwise>
                                                    <KORDT></KORDT>
                                                </xsl:otherwise>
                                            </xsl:choose>

                                        <xsl:choose>

```

```

                                <xsl:when test="KORZT">
                                    <KORZT><xsl:value-of sel-
ect="KORZT"/></KORZT>

                                </xsl:when>
                                <xsl:otherwise>
                                    <KORZT></KORZT>
                                </xsl:otherwise>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="KORSA">
                                        <KORSA><xsl:value-of sel-
ect="KORSA"/></KORSA>

                                    </xsl:when>
                                    <xsl:otherwise>
                                        <KORSA></KORSA>
                                    </xsl:otherwise>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="TSTMS">
                                        <TSTMS><xsl:value-of sel-
ect="TSTMS"/></TSTMS>

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

                                <xsl:choose>
                                    <xsl:when test="FAK01">
                                        <FAK01><xsl:value-of sel-
ect="FAK01"/></FAK01>

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

                                <xsl:choose>
                                    <xsl:when test="FAK02">
                                        <FAK02><xsl:value-of sel-
ect="FAK02"/></FAK02>

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

                                <xsl:choose>
                                    <xsl:when test="FAK03">
                                        <FAK03><xsl:value-of sel-
ect="FAK03"/></FAK03>

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

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

```

```

MIN"/></ZUMIN>
                                <ZUMIN><xsl:value-of select="ZU-
                                </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="ZZFSL">
                                        <ZZFSL><xsl:value-of sel-
ect="ZZFSL"/></ZZFSL>
                                    </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="ZZBWK">
                                        <ZZBWK><xsl:value-of sel-
ect="ZZBWK"/></ZZBWK>
                                    </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="ZZBKS">
                                        <ZZBKS><xsl:value-of sel-
ect="ZZBKS"/></ZZBKS>
                                    </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="ZZBAP">
                                        <ZZBAP><xsl:value-of sel-
ect="ZZBAP"/></ZZBAP>
                                    </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="KOMPO">
                                        <KOMPO><xsl:value-of sel-
ect="KOMPO"/></KOMPO>
                                    </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="TERID">
                                        <TERID><xsl:value-of select="TERID"/></TE-
RID>
                                    </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="TEXT1">

```

```

                                <TEXT1><xsl:value-of sel-
ect="TEXT1"/></TEXT1>

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

                                <xsl:choose>
                                    <xsl:when test="VERID">
                                        <VERID><xsl:value-of select="VERID"/></VE-
RID>

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

                                <xsl:choose>
                                    <xsl:when test="RSNUM">
                                        <RSNUM><xsl:value-of sel-
ect="RSNUM"/></RSNUM>

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

                                <xsl:choose>
                                    <xsl:when test="RSPOS">
                                        <RSPOS><xsl:value-of sel-
ect="RSPOS"/></RSPOS>

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

                                <xsl:choose>
                                    <xsl:when test="CHARG">
                                        <CHARG><xsl:value-of sel-
ect="CHARG"/></CHARG>

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

                                <xsl:choose>
                                    <xsl:when test="PALET">
                                        <PALET><xsl:value-of select="PALET"/></PA-
LET>

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

                                <xsl:choose>
                                    <xsl:when test="LSDAT">
                                        <LSDAT><xsl:value-of sel-
ect="LSDAT"/></LSDAT>

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

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

```

```

                                <GWCHT><xsl:value-of sel-
ect="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 Aufträge

```

<?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="''&#10;''" />
    </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>

    <!-- Template for mapping the units SEK and SEC to S. -->
    <xsl:template name="editTimeUnits">
        <xsl:param name="originalTimeUnit" />
        <xsl:choose>
            <xsl:when test="$originalTimeUnit='SEK' or $original-
TimeUnit='SEC'">
                <xsl:text>S</xsl:text>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="$originalTimeUnit" />
            </xsl:otherwise>
        </xsl:choose>
    </xsl:template>

    <!-- Template for removing the leading zeros in a string. -->
    <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>

```

```

<!--
    Transform a date depending on the language to the format YYYY-MM-DDT:00:00:00.
-->
<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', $timezone)"/>
        </xsl:when>
        <!-- hu-HU => yyyy.mm.dd, jp-JP => yyyy/mm/dd -->
        <xsl:when test="$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" sel-
ect="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:ERPSta-
tusCodesTranslation>
            </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" sel-
ect="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:ERPSta-
tusCodesTranslation>

        <xsl:if test="APRIO">
            <ord:OrderPriority><xsl:value-of sel-
ect="APRIO"/></ord:OrderPriority>
        </xsl:if>

        <!-- Create release timestamp in format dd.MM.yyyyT00:00:00 -->
        <ord:ReleaseDateTime>
            <xsl:call-template name="transformDateToTi-
mestamp">
                <xsl:with-param name="originalDate" sel-
ect="FRDAT"/>
            </xsl:call-template>

```

```

        </ord:ReleaseDateTime>

        <!-- Create target start timestamp in format dd.MM.yyyyT00:00:00 -->
        <ord:TargetStartDateTime>
            <xsl:call-template name="transformDateToTi-
mestamp">
                <xsl:with-param name="originalDate" sel-
ect="SSTRT"/>
            </xsl:call-template>
        </ord:TargetStartDateTime>

        <!-- Create target end timestamp in format dd.MM.yyyyT00:00:00 -->
        <ord:TargetEndDateTime>
            <xsl:call-template name="transformDateToTi-
mestamp">
                <xsl:with-param name="originalDate" sel-
ect="SLIEF"/>
            </xsl:call-template>
        </ord:TargetEndDateTime>

        <ord:TargetQuantity>
            <xsl:call-template name="transformNumberDelime-
ters">
                <xsl:with-param name="number" select="AF-
ANZ"/>
            </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 sel-
ect="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:PlannerGrou-
pInCharge>
        </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>
  </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 sel-
ect="___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:characteristi-
cName>
          <ord:characteristicValue1><xsl:value-of select="ATWRT"/></ord:characteristi-
cValue1>
          <ord:characteristicValue2><xsl:value-of select="ATW02"/></ord:characteristi-
cValue2>
          <ord:characteristicValue3><xsl:value-of select="ATW03"/></ord:characteristi-
cValue3>
          <ord:characteristicValue4><xsl:value-of select="ATW04"/></ord:characteristi-
cValue4>
          <ord:characteristicValue5><xsl:value-of select="ATW05"/></ord:characteristi-
cValue5>
          <ord:characteristicValue6><xsl:value-of select="ATW06"/></ord:characteristi-

```

```

cValue7>      <ord:characteristicValue7><xsl:value-of select="ATW07"/></ord:characteristi-
cValue8>      <ord:characteristicValue8><xsl:value-of select="ATW08"/></ord:characteristi-
cValue9>      <ord:characteristicValue9><xsl:value-of select="ATW09"/></ord:characteristi-
cValue10>     <ord:characteristicValue10><xsl:value-of select="ATW10"/></ord:characteristi-
cValue11>     <ord:characteristicValue11><xsl:value-of select="ATW11"/></ord:characteristi-
cValue12>     <ord:characteristicValue12><xsl:value-of select="ATW12"/></ord:characteristi-
cValue13>     <ord:characteristicValue13><xsl:value-of select="ATW13"/></ord:characteristi-

```

```

      </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 below).

```
-->
```

```
<!--
```

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>
      <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>
```

```

</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:Confirma-
tionNumber>

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

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

```

```

        <ord:ERPPlannedScrapQuantity><xsl:value-of sel-
ect="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 sel-
ect="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___SA-
FOTX"/>
            </xsl:variable>

            <ord:OperationText>
            <Translation>
            <Language><xsl:value-of select="$language-
Code"/></Language>
            <Text><xsl:value-of select="$concatenated-
Text"/></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="transformDateToTi-
mestamp">
                        <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="transformDateToTi-
mestamp">
                        <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: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="transformDateToTi-
mestamp">
                        <xsl:with-param name="originalDate"
select="__FFMES__SOPPLD/SSEDD"/>
                        <xsl:with-param name="originalTime"
select="__FFMES__SOPPLD/SSEDZ"/>
                        <xsl:with-param name="timezone"
select="UTCOFF"/>
                    </xsl:call-template>
                </ord:TargetEndDateTime>
            </xsl:when>
            <xsl:otherwise>
                <ord:TargetEndDateTime>
                    <xsl:call-template name="transformDateToTi-
mestamp">

```

```

select="SSEDD"/>
select="SSEDZ"/>
select="UTCOFF"/>

        <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="transformNumberDelimiters">
                <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 sel-
ect="MEINH"/></ord:DisplayQuantityUnit>
        </xsl:when>
        <xsl:otherwise>
                <ord:DisplayQuantityUnit>stk</ord:DisplayQuanti-
tyUnit>
        </xsl:otherwise>
</xsl:choose>

<ord:TargetQuantity>
        <xsl:call-template name="transformNumberDelimiters">
                <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="transformNumber-
Delimiters">
                                <xsl:with-param name="number" sel-
ect="___FFMES___SOPPLD/RUEST"/>
                        </xsl:call-template>
                </ord:StandardValue1>
        </xsl:when>
        <xsl:otherwise>
                <ord:StandardValue1>
                        <xsl:call-template name="transformNumber-
Delimiters">

```

```

ect="VGW01"/>
                                <xsl:with-param name="number" sel-
                                </xsl:call-template>
                                </ord:StandardValue1>
                                </xsl:otherwise>
                                </xsl:choose>

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

                                <ord:StandardValue2>
                                <xsl:call-template name="transformNumberDelimiters">
                                <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 sel-
                                ect="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:AlternateOperati-
                                onNumber>
                                </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:OverdeliveryCh-
eck>

          </xsl:when>
          <xsl:otherwise>
            <ord:OverdeliveryCheck>false</ord:OverdeliveryCh-
eck>

            </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:Underdelivery-
Check>

              </xsl:when>
              <xsl:otherwise>
                <ord:UnderdeliveryCheck>false</ord:Underdelivery-
Check>

                </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">

```

30.09.19

```

                                <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="transformDateToTi-
mestamp">
                                <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="transformDateToTi-
mestamp">
                                <xsl:with-param name="originalDate"
select="FSEDD"/>
                                <xsl:with-param name="timezone"
select="UTCOFF"/>
                                </xsl:call-template>
                                </ord:EarliestEndDate>
                                </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="transformDateToTi-
mestamp">
                                <xsl:with-param name="originalDate"
select="___FFMES___SOPPLD/SSEDD"/>
                                <xsl:with-param name="timezone"
select="UTCOFF"/>
                                </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="transformDateToTi-
mestamp">
                                            <xsl:with-param name="originalDate"
select="SSEDD"/>
                                            <xsl:with-param name="timezone"
select="UTCOFF"/>
                                        </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="transformDateToTi-
mestamp">
                                            <xsl:with-param name="originalDate"
select="___FFMES___SOPPLD/EPANF"/>
                                            <xsl:with-param name="timezone"
select="UTCOFF"/>
                                        </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="transformDateToTi-
mestamp">
                                            <xsl:with-param name="originalDate"
select="EPANF"/>
                                            <xsl:with-param name="timezone"
select="UTCOFF"/>
                                        </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="transformDateToTi-
mestamp">
                                <xsl:with-param name="originalDate"
select="___FFMES___SOPPLD/EPEND"/>
                                <xsl:with-param name="timezone"
select="UTCOFF"/>
                                </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="transformDateToTi-
mestamp">
                                <xsl:with-param name="originalDate"
select="EPEND"/>
                                <xsl:with-param name="timezone"
select="UTCOFF"/>
                                </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="transformNumber-
Delimiters">
                <xsl:with-param name="number" sel-
ect="___FFMES___SOPPLD/ZWNOR"/>
            </xsl:call-template>
        </ord:TargetQueueTime>
    </xsl:when>
    <xsl:when test="ZWNOR">
        <ord:TargetQueueTime>
            <xsl:call-template name="transformNumber-
Delimiters">
                <xsl:with-param name="number" sel-
ect="ZWNOR"/>
            </xsl:call-template>
        </ord:TargetQueueTime>
    </xsl:when>
</xsl:choose>

<xsl:choose>
    <xsl:when test="___FFMES___SOPPLD/ZEIWN">
        <ord:TargetQueueTimeUnit><xsl:value-of sel-
ect="___FFMES___SOPPLD/ZEIWN"/></ord:TargetQueueTimeUnit>

```



```

        </xsl:when>
        <xsl:when test="ZEIWN">
            <ord:TargetQueueTimeUnit><xsl:value-of sel-
ect="ZEIWN"/></ord:TargetQueueTimeUnit>
        </xsl:when>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="___FFMES___SOPPLD/BEARZ">
            <ord:TargetProcessingTime>
                <xsl:call-template name="transformNumber-
Delimiters">
                    <xsl:with-param name="number" sel-
ect="___FFMES___SOPPLD/BEARZ"/>
                </xsl:call-template>
            </ord:TargetProcessingTime>
        </xsl:when>
        <xsl:when test="BEARZ">
            <ord:TargetProcessingTime>
                <xsl:call-template name="transformNumber-
Delimiters">
                    <xsl:with-param name="number" sel-
ect="BEARZ"/>
                </xsl:call-template>
            </ord:TargetProcessingTime>
        </xsl:when>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="___FFMES___SOPPLD/BEAZE">
            <ord:TargetProcessingTimeUnit><xsl:value-of sel-
ect="___FFMES___SOPPLD/BEAZE"/></ord:TargetProcessingTimeUnit>

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

    <xsl:choose>
        <xsl:when test="___FFMES___SOPPLD/ABRUE">
            <ord:TargetTeardownTime>

```

```

                                <xsl:call-template name="transformNumber-
Delimiters">
                                <xsl:with-param name="number" sel-
ect="___FFMES___SOPPLD/ABRUE"/>
                                </xsl:call-template>
                                </ord:TargetTeardownTime>
                                </xsl:when>
                                <xsl:when test="ABRUE">
                                    <ord:TargetTeardownTime>
                                        <xsl:call-template name="transformNumber-
Delimiters">
                                            <xsl:with-param name="number" sel-
ect="ABRUE"/>
                                            </xsl:call-template>
                                            </ord:TargetTeardownTime>
                                        </xsl:when>
                                    </xsl:choose>
                                <xsl:choose>
                                    <xsl:when test="___FFMES___SOPPLD/ARUZE">
                                        <ord:TargetTeardownTimeUnit><xsl:value-of sel-
ect="___FFMES___SOPPLD/ARUZE"/></ord:TargetTeardownTimeUnit>
                                    </xsl:when>
                                    <xsl:when test="ARUZE">
                                        <ord:TargetTeardownTimeUnit><xsl:value-of sel-
ect="ARUZE"/></ord:TargetTeardownTimeUnit>
                                    </xsl:when>
                                </xsl:choose>
                                <xsl:choose>
                                    <xsl:when test="___FFMES___SOPPLD/ZLMAX">
                                        <ord:TargetWaitTime>
                                            <xsl:call-template name="transformNumber-
Delimiters">
                                                <xsl:with-param name="number" sel-
ect="___FFMES___SOPPLD/ZLMAX"/>
                                                </xsl:call-template>
                                            </ord:TargetWaitTime>
                                        </xsl:when>
                                    <xsl:when test="ZLMAX">
                                        <ord:TargetWaitTime>
                                            <xsl:call-template name="transformNumber-
Delimiters">
                                                <xsl:with-param name="number" sel-
ect="ZLMAX"/>
                                                </xsl:call-template>
                                            </ord:TargetWaitTime>
                                        </xsl:when>
                                    </xsl:choose>
                                <xsl:choose>
                                    <xsl:when test="___FFMES___SOPPLD/ZEILM">

```

```

                                <ord:TargetWaitTimeUnit><xsl:value-of sel-
ect="___FFMES___SOPPLD/ZEILM"/></ord:TargetWaitTimeUnit>
                                </xsl:when>
                                <xsl:when test="ZEILM">
                                    <ord:TargetWaitTimeUnit><xsl:value-of sel-
ect="ZEILM"/></ord:TargetWaitTimeUnit>
                                </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="___FFMES___SOPPLD/ZTNOR">
                                        <ord:TargetMoveTime>
                                            <xsl:call-template name="transformNumber-
Delimiters">
                                                <xsl:with-param name="number" sel-
ect="___FFMES___SOPPLD/ZTNOR"/>
                                            </xsl:call-template>
                                        </ord:TargetMoveTime>
                                    </xsl:when>
                                    <xsl:when test="ZTNOR">
                                        <ord:TargetMoveTime>
                                            <xsl:call-template name="transformNumber-
Delimiters">
                                                <xsl:with-param name="number" sel-
ect="ZTNOR"/>
                                            </xsl:call-template>
                                        </ord:TargetMoveTime>
                                    </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="___FFMES___SOPPLD/ZEITN">
                                        <ord:TargetMoveTimeUnit><xsl:value-of sel-
ect="___FFMES___SOPPLD/ZEITN"/></ord:TargetMoveTimeUnit>
                                    </xsl:when>
                                    <xsl:when test="ZEITN">
                                        <ord:TargetMoveTimeUnit><xsl:value-of sel-
ect="ZEITN"/></ord:TargetMoveTimeUnit>
                                    </xsl:when>
                                </xsl:choose>

                                <xsl:choose>
                                    <xsl:when test="___FFMES___SOPPLD/MINWE">
                                        <ord:MinimumSendAheadQuantity>
                                            <xsl:call-template name="transformNumber-
Delimiters">
                                                <xsl:with-param name="number" sel-
ect="___FFMES___SOPPLD/MINWE"/>
                                            </xsl:call-template>
                                        </ord:MinimumSendAheadQuantity>
                                    </xsl:when>
                                </xsl:choose>

```

```

        <xsl:when test="MINWE">
            <ord:MinimumSendAheadQuantity>
                <xsl:call-template name="transformNumber-
Delimiters">
                    <xsl:with-param name="number" sel-
ect="MINWE"/>
                </xsl:call-template>
            </ord:MinimumSendAheadQuantity>
        </xsl:when>
    </xsl:choose>

    <xsl:choose>
        <xsl:when test="___FFMES___SOPPLD/ZMINU">
            <ord:MinimumOverlapTime>
                <xsl:call-template name="transformNumber-
Delimiters">
                    <xsl:with-param name="number" sel-
ect="___FFMES___SOPPLD/ZMINU"/>
                </xsl:call-template>
            </ord:MinimumOverlapTime>
        </xsl:when>
        <xsl:when test="ZMINU">
            <ord:MinimumOverlapTime>
                <xsl:call-template name="transformNumber-
Delimiters">
                    <xsl:with-param name="number" sel-
ect="ZMINU"/>
                </xsl:call-template>
            </ord:MinimumOverlapTime>
        </xsl:when>
    </xsl:choose>
    <xsl:choose>
        <xsl:when test="___FFMES___SOPPLD/ZEIMU">
            <ord:MinimumOverlapTimeUnit><xsl:value-of sel-
ect="___FFMES___SOPPLD/ZEIMU"/></ord:MinimumOverlapTimeUnit>
        </xsl:when>
        <xsl:when test="ZEIMU">
            <ord:MinimumOverlapTimeUnit><xsl:value-of sel-
ect="ZEIMU"/></ord:MinimumOverlapTimeUnit>
        </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="deleteOperation-
Components"/>
    </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="addProduction-
ResourceTools"/>
    </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 va-
lue.

```

50 userfields from UF01 - UF50 are available (only the first then userfields are in the sample below).

-->

<!--

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>
```

```
    </ord:UserFields>
```

```
  </xsl:for-each>
```

```
</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:Confirma-
tionNumber>

            <ord:ERPStatusIds><xsl:value-of select="STATU_ID"/></ord:ERPStatusIds>
            <ord:ERPStatusCodes><xsl:value-of select="STATU"/></ord:ERPStatusCodes>
            <ord:ERPStatusCodesTranslation><xsl:value-of sel-
ect="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 dele-
ted. -->
<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="{ $PRODUCTION_RESOURCE_TOOL_ACTION }">
            <ord:Number><xsl:value-of select="MATNR"/></ord:Number>
            <ord:Type><xsl:value-of select="FHART"/></ord:Type>
            <ord:Sequence><xsl:value-of select="LFDNR"/></ord:Sequence>
            <ord:Quantity>
                <xsl:call-template name="transformNumberDelimiters">
                    <xsl:with-param name="number" sel-
ect="MGVGW"/>
                </xsl:call-template>
            </ord:Quantity>
            <ord:Unit><xsl:value-of select="MGEINH"/></ord:Unit>
            <ord:Group><xsl:value-of select="STATU"/></ord:Group>
            <ord:Description>
                <Translation>
                    <Language><xsl:value-of select="$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 sel-
ect="MATNR"/></ord:ComponentNumber>
            <ord:PositionNumber><xsl:value-of select="RSPOS"/></ord:Position-
Number>
            <ord:BaseUnitOfMeasure><xsl:value-of select="MEINS"/></ord:Base-
UnitOfMeasure>

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

```



```

        <ord:ConfirmationNumber><xsl:value-of sel-
ect="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" sel-
ect="BDTER"/>
                </xsl:call-template>
            </ord:RequirementDate>

            <ord:RequirementQuantity>
                <xsl:call-template name="transformNumberDelimeters">
                    <xsl:with-param name="number" select="BDMNG"/>
                </xsl:call-template>
            </ord:RequirementQuantity>
            <ord:ReservationNumber><xsl:value-of select="RSNUM"/></ord:Re-
servationNumber>

            <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 deleted. -->
<xsl:template name="deleteOperationComponents">
  <xsl:for-each select="___FFMES___SAFOKO[VAKZ = 'D']">
    <ord:OperationComponent action="DELETE">
      <ord:ComponentNumber><xsl:value-of select="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 Schichten

```
<?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>
            </ERPKey>
        </Shift>
    </xsl:template>
</xsl:stylesheet>
```

```
<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>

    <!-- 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 Personal

```
<?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.for-
cam.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>
            <PersonnelNumber><xsl:value-of select="PERNR"/></PersonnelNumber>
            <IdentificationNumber><xsl:value-of select="ZAUSW"/></IdentificationNum-
ber>

            <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>
```