



FORCE Bridge API ERP Interface

Version 5.12

Manual



Document: Manual – FORCE Bridge API: ERP Interface



Release date: 2023-11-21



Document version: 2



Author: FORCAM GmbH

Contents

1	About this document	5
1.1	Target group	5
2	Introduction to FORCE Bridge API	6
2.1	Technical base knowledge	6
2.1.1	Explorable API	6
2.1.2	Collections	6
2.1.3	Revisioning	7
2.1.4	HTTP requests	7
2.1.5	HTTP status codes	7
2.1.6	Authentication	8
2.2	Using FORCE Bridge API	9
2.2.1	How to request OAuth tokens	9
2.2.2	How to send POST requests	10
3	Configuration of webservices.....	12
3.1	URLs.....	12
3.2	API clients	13
3.3	Job scheduling.....	15
3.4	Customer-specific settings.....	17
4	FORCE Bridge API as a digital image of a production site.....	19
4.1	Basic concepts and objects used in production	19
4.1.1	Entities in the domain of production.....	19
4.1.2	Resources of the FORCE Bridge API.....	22
4.2	Shifts and scheduled operating times of workplaces	23
4.2.1	Domain-specific knowledge.....	23
4.2.2	Workplace resources in the FORCE Bridge API.....	24
5	Callbacks and events.....	27
5.1	Callbacks	27
5.1.1	Callback request to the external webservice	27
5.1.2	Retries.....	27
5.1.3	Creating a callback.....	27
5.1.4	Get all registered callbacks.....	29
5.1.5	Delete a registered callback	29

5.1.6	Event types	30
5.1.7	Receiving events	30
5.1.8	Sending self-defined events to applications or third-party systems	30
6	ERP interface – data and events via XML	32
7	ERP download and upload services	33
8	ERP interface configuration	34
8.1	ERP download	34
8.2	ERP upload	35
8.2.1	ERP upload connection configuration	37
8.3	Filter – Unused closed operations	38
8.4	ERP control key	40
8.5	Connection via Infor ION	40
8.6	Assigning operations to workplace groups	41
8.6.1	Piece time factor	42
8.7	Workplace-specific configuration	43
8.7.1	Activating ERP upload	43
8.7.2	ERP cycle time	43
8.7.3	Send scrap quantity to ERP	44
9	Data structures	46
9.1	XSL basic data types	46
9.2	XSL complex data types	47
10	Download	49
10.1	Download basic data types	49
10.1.1	Time fields	49
10.1.2	Multilingual texts	50
10.1.3	Dataset action	51
10.1.4	ERP key	51
10.1.5	Material	52
10.1.6	User data fields	53
10.2	Order data	54
10.2.1	Order header	54
10.2.2	Material characteristics	56
10.2.3	Operations	57
10.2.4	Components	62

10.2.5 Production resources and tools	63
10.3 Shift data	64
10.4 Personnel data	66
10.5 Overhead cost data	67
10.6 Warehouse data from the warehouse management system	70
10.7 Response to quantity requests	71
11 Upload	72
11.1 Uploading basic data types	73
11.1.1 ERP key	73
11.1.2 Messages to ERP system	73
11.2 Messages on operations	74
11.2.1 Change of operation phase	75
11.2.2 Operation quantity bookings and reasons	75
11.2.3 Operation duration	76
11.2.4 Component quantities	76
11.2.5 Assembly of components	77
11.3 Operation scheduling	78
11.4 Overhead cost messages	78
11.5 Warehouse data to the warehouse management system	78
11.6 Quantity requests	80
11.7 Serialization and correction capability	80
12 Annex	81
12.1 Abbreviations and terms used	81

1 About this document

This document describes the use of the FORCE Bridge API and the ERP interface for data and events via XML.

- ❗ For better readability, we generally use the generic masculine in the text. The phrasing, however, is equally inclusive of all genders and intended to address all persons equally.

1.1 Target group

This manual requires knowledge in handling of FORCE MES FLEX (hereinafter simply referred to as MES FLEX), FORCE Bridge API and the ERP interface. If you do not have any knowledge in this area, take the time to familiarize yourself with the basics.

- ❗ We recommend that you use our Academy: <https://forcam.com/academie/>
The FORCAM Academy provides the knowledge to effectively use the methods for digital transformation and the technologies for the Smart Factory.
Based on lean manufacturing and TPM methods, our institute team will guide you to initiate changes in your company and to use the technologies appropriately.

2 Introduction to FORCE Bridge API

FORCE Bridge API is the Application Programming Interface between the objects of the production environment and the IT infrastructure of a production site.

FORCE Bridge API serves as the bridge between the real production site, IT systems and applications used for effective and efficient organization.

The industrial IoT platform fulfills two functions:

- FORCE Bridge API provides a complete digital image of a production plant with its relevant objects, including their statuses. Objects are real world entities such as persons, machines and tools as well as abstract entities such as production orders or operations.
- FORCE Bridge API ensures the organizational interoperability¹ of persons, production facilities and IT systems of a production plant with a view to maximizing resource effectiveness and process efficiency.

2.1 Technical base knowledge

FORCE Bridge API connects your application to an IoT platform. Your application can use the API to access data, send commands to FORCE Runtime or listen for events which occur in the system.

- All API access is performed via HTTP(S).
Request payloads are formatted as *application/json*.
Response payloads are formatted as *application/json* or *application/hal+json*.
- **OAuth2** is used for all authentication.
All API requests must be authenticated or there will be a “401 Unauthorized” error response (see **Authentication**).
- FORCE Bridge API is an **explorable API** and makes resources available.
- Every resource can be identified by its universal unique identifier (UUID).

 The complete Swagger specification of FORCE Bridge API is available [here](#).

2.1.1 Explorable API

Response payloads are enriched with hyperlinks using the **Hypermedia Application Language (HAL)**. HAL is a simple format that provides a set of conventions for expressing hyperlinks in JSON – it’s basically plain old JSON with hyperlinks. This encourages developers to build applications that use these hyperlinks to navigate between resources instead of using hard-coded URLs.

The following *fields* can be expected on every resource:

- *properties* – The regular properties of the resource
- *_links* – JSON object containing a set of URL paths relative to the API base URL
- *_embedded* – JSON object containing other resources embedded inside the current resource

2.1.2 Collections

A set of resources is considered a *collection*.

¹ Interoperability describes the ability to exchange and make use of information between two different systems.

2.1.3 Revisioning


The API is revised by identifiers in the URI.

The version of the API is revised (incremented) only when an essential change occurs in one of the webservises. Essential changes are:

- Removal of a JSON property
- Renaming of a JSON property
- General restructuring in the design of existing representations

Essential changes are not:

- Adding JSON properties
- Adding new webservises

 Revisioning is global. This means that a webservice can return the same response for different versions if no change has been made to that specific webservice.

It is possible that a new release adds new JSON properties to the response. This does not increment the webservice version. To ensure downward compatibility, every client communicating with the FORCE Bridge API should process new properties only if needed.

2.1.4 HTTP requests

The API strives to use appropriate **HTTP METHODS** for each action wherever possible.

Method	Description
GET	Used for retrieving resources.
POST	Used for creating resources or executing an action, among other things.
PUT	Used for replacing resources.
DELETE	Used for deleting resources.

Table 1: HTTP methods

Key	Value
authorization	Authentication token
accept header	<i>application/json</i> or <i>application/hal+json</i>
accept language	<i>en-GB</i> (standard), <i>en-US</i> , <i>de-DE</i> , <i>zh-CN</i> , etc. (all languages configured in the Workbench)

Table 2: HTTP methods headers


2.1.5 HTTP status codes


The API returns appropriate **HTTP status codes** for every request.

Code	Description
200 OK	Success
201 Created	The request succeeded and resulted in a new resource being created. The Location header of the response contains the URI of the new resource.
202 Accepted	The request has been accepted and is being processed, but processing has not been completed yet.
204 No content	The request succeeded but the server does not return any content. This is the response for most DELETE requests.
400 Bad request	The request was invalid or cannot be processed. An accompanying error message will explain why.
401 Unauthorized	Authentication credentials were missing or incorrect.
403 Forbidden	The request was received, but it has been refused or access is not allowed. An accompanying error message will explain why.
404 Not found	The URI requested is invalid or the resource requested (such as an event) does not exist.
415 Unsupported media type	The request entity has a content type that the server does not support.
500 Internal server error	An unexpected error occurred.
504 Gateway timeout	The server acting as a gateway or proxy did not receive a timely response from the upstream server it needed to access in attempting to complete the request.

2.1.6 Authentication

FORCE Bridge API uses the OAuth 2.0 protocol for authentication.

 OAuth is only secure if it is used via the TLS/SSL protocol, so every request and all API endpoints must use HTTPS.

 Tokens are passwords: Your application's client id, client secret and access tokens should be considered as sensitive as passwords and should not be shared.

All applications follow a basic pattern when accessing the FORCE Bridge API.

In general, follow three steps:

- Obtain an access token.
 - To make an authentication request to the API, your application must first obtain an OAuth access token using the client credentials flow.
- Use the access token to perform requests.
After your application obtains an access token and makes API requests, it sends the *access_token* property in a token response in a **Bearer authorization header**.
- Refresh the access token when it expires.
Access tokens have limited lifetimes. If your application needs to make a request beyond the lifetime of an access token, it has to obtain a new access token.

2.2 Using FORCE Bridge API

In contrast to all other HTTP methods, POST must be used for different purposes to avoid limitless calls in certain situations.

In a narrower sense, POST is used to create one or more new resources. To do this, the relevant collection is called with POST:

POST/{collection}

According to the hypermedia principle of the RESTful architecture, whether or not a resource can be created depends on whether the associated collection has a corresponding hyperlink. For example, FORCE Bridge API provides a hyperlink to the following method in the collection of tool assembly orders that applications can use to create new tools:

POST/tools/assemblyOrders

All properties of the tool assembly order are transferred with the method call to the IoT platform, which in turn responds with a representation of the newly created resource if the call was successful. In particular, the UUID of the newly created tool is communicated in the response.

Another use of POST is to change the state of resources. To do this, the single resource in question is called with POST and a specification of the new state:

POST/{collection}/{id}/{state}

FORCE Bridge API allows a tool management system to set the state of a tool assembly order, for example, by the following call:

POST/tools/assemblyOrders/{toolAssemblyOrderId}/{toolAssemblyOrderStateId}

Although the PUT or PATCH method could also be used for this purpose, the calls would be much more cumbersome since PUT requires passing also properties that are not changed. On the other hand, it is difficult to reconcile this with the hypermedia concept of a **REST** architecture. A REST architecture requires that an application representing a resource is presented with all possible state transitions as hyperlinks and that each state transition is triggered by following a hyperlink. It is possible to provide the following reference in HAL:

```
1. "updateToolAssemblyOrderState": {
2.   "method": "POST",
3.   "embeddable": false,
4.   "href": ".../tools/assemblyOrders/E446BB6B0C084CB09D0BC0319A8A1F1C/AVAILABLE"
5. }
```

It is also possible to use the POST method instead of the GET method. This is because the GET method is not suitable for setting a large number of filter parameters when it is called. This problem occurs if the collections of production orders or operations are to be filtered explicitly by hundreds of order numbers. The POST method is therefore also provided for retrieving both collections. The following notation is used to indicate that such a POST method is intended for retrieving a collection and not for creating a new resource:

POST{collection}/search

2.2.1 How to request OAuth tokens

Applications must use the **Client Credentials Flow of the OAuth2 specification** to get an access token that will provide access to FORCE Bridge API.

To create an authorization header:

1. URL encode your application's client ID and client secret according to **RFC 1738**.
2. Concatenate the encoded client ID, a colon character (:) and the encoded client secret into a single string.
3. Add "Basic" to the string.
4. Encode the string from the previous step in Base64.

Example

Header: { Authorisation : Basic xyz Content-Type: application/x-www-form-urlencoded }
 Body: 'grant_type=client_credentials&scope =read write'

Example code

```
{
    "// How to create Authorization String"
    "var encodedClientID = encodeURI(clientID); //encode with URLEncode"
    "var encodedClientSecret = encodeURI(clientSecret); //encode with URLEncode"
    "var pair = encodedClientID+':'+encodedClientSecret; //combine Strings with Semicolon"
    "var encodedPair = window.btoa(pair); //encode with Base64"
    "var authorizationValue = 'Basic' +encodedPair; //add Basic to the String"
}
```

2.2.2 How to send POST requests

The token URL is:

POST http(s)://\$HOST:\$PORT/ffwebservices/oauth/token

Name	Value
Authorization	"Basic" + Base64 encoded clientid:clientsecret from previous step
Content type	application/x-www-urlencoded

Table 3: POST request headers

Name	Type	Value
grant_type	string	Required. Value should be <i>client_credentials</i> .
scope	string	Space-delimited string of the scopes you would like.

Table 4: POST request parameters

Name	Description
read	Grants permission to call HTTP GET
write	Grants permission to call HTTP POST, PUT, DELETE

Table 5: POST request scopes

- ⚠ Token expiration: Applications should be written by anticipating the possibility that a granted token may no longer work. The application must authenticate again to get new access tokens. The time until a token expires is returned with the token response.

CURL

```
curl -X POST http(s)://$HOST:$PORT/ffwebservices/oauth/token \  
--header "accept:application/x-www-urlencode" \  
-d "client_id=$CLIENT_ID" \  
-d "client_secret=$CLIENT_SECRET" \  
-d "grant_type=client_credentials" \  
-d "scope=read%20write"
```

Example request

```
{  
  "//access token must be requested first"  
  "method": "POST",  
  "headers": "{",  
  "content-type": "application/json",  
  "Authorization": "'Bearer ' +access_token"  
}
```

Example response

```
{  
  "access_token": "537517ab-faa3-4ad2-8ae5-37ff91ffb7c0",  
  "token_type": "bearer",  
  "expires_in": 42523,  
  "scope": "read write"  
}
```

3 Configuration of webservices

If you intend to transfer data via the FORCE Bridge API, you need to configure the connection to the target or source system, as appropriate. This is done in the Workbench in the Configurations tile.

3.1 URLs

Path: Configurations > Modules > Webservices > URLs

Webservices Configuration is used for configuring webservices of external systems and their use. This requires configuring the connection data of the respective system in the URLs section.

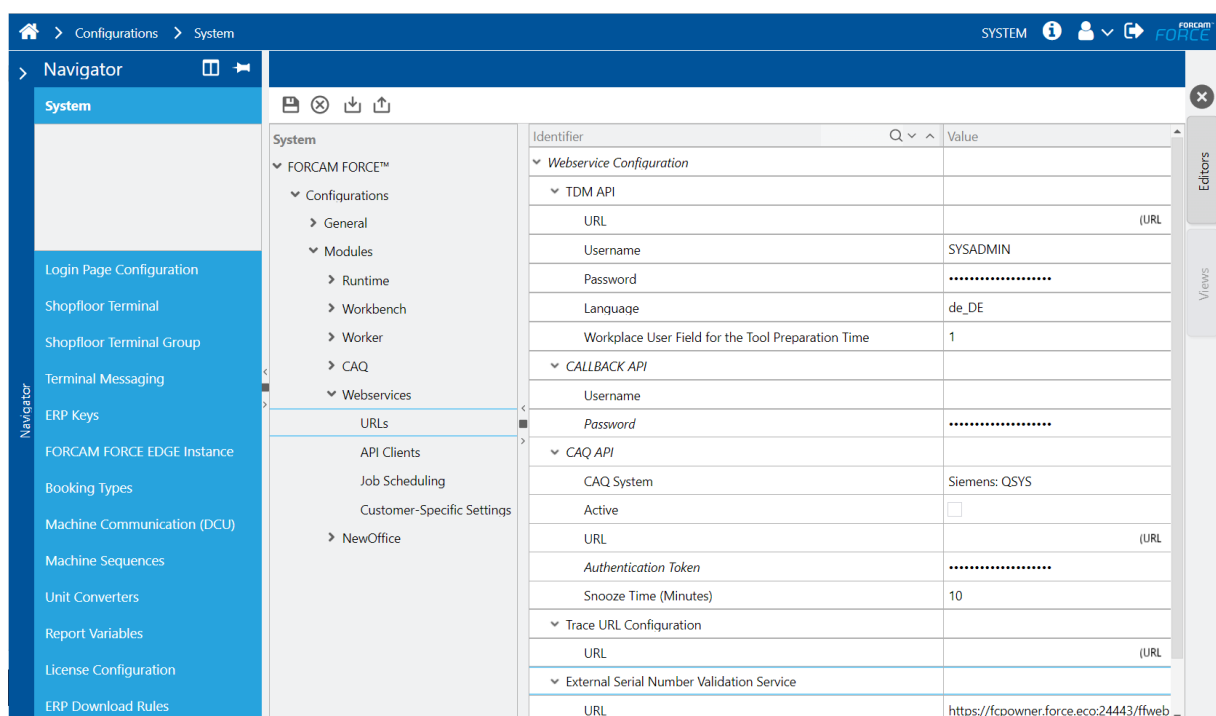


Fig. 1: Configuration: Webservices > URLs

Designation	Explanation
TDM API	
URL	URL of the TDM system
Username	Username for basic authentication
Password	Password for basic authentication
Language	Language used for messages to TDM
Workplace User Field for the Tool Preparation Time	Number of the field in which the time for tool preparation is specified

Designation	Explanation
CALLBACK API	
Username	Global username for basic authentication for callbacks
Password	Global password for basic authentication for callbacks
CAQ API	
CAQ System	Selection of the external CAQ system (currently QSYS and CAQ.net)
Active	Connection active
URL	URL of the CAQ system
Authentication Token	Token for authentication on the CAQ system
Snooze Time (Minutes)	Time interval for reminding the worker in case of negative acknowledgement
Trace URL Configuration	
URL	URL of the trace system
External Serial Number Validation Service	
URL	URL of the service for validating the track & trace serial numbers
Active	Service to be used
CAQ Module	
URL	URL of the service for integration of external CAQ systems via node RED
Active	Service to be used

Table 6: Configuration of Webservices > URLs

3.2 API clients

Path: Configurations > Modules > Webservices > API Clients > OpenAPI

This section is used for configuring API clients and the corresponding rights for using FORCE Bridge API.

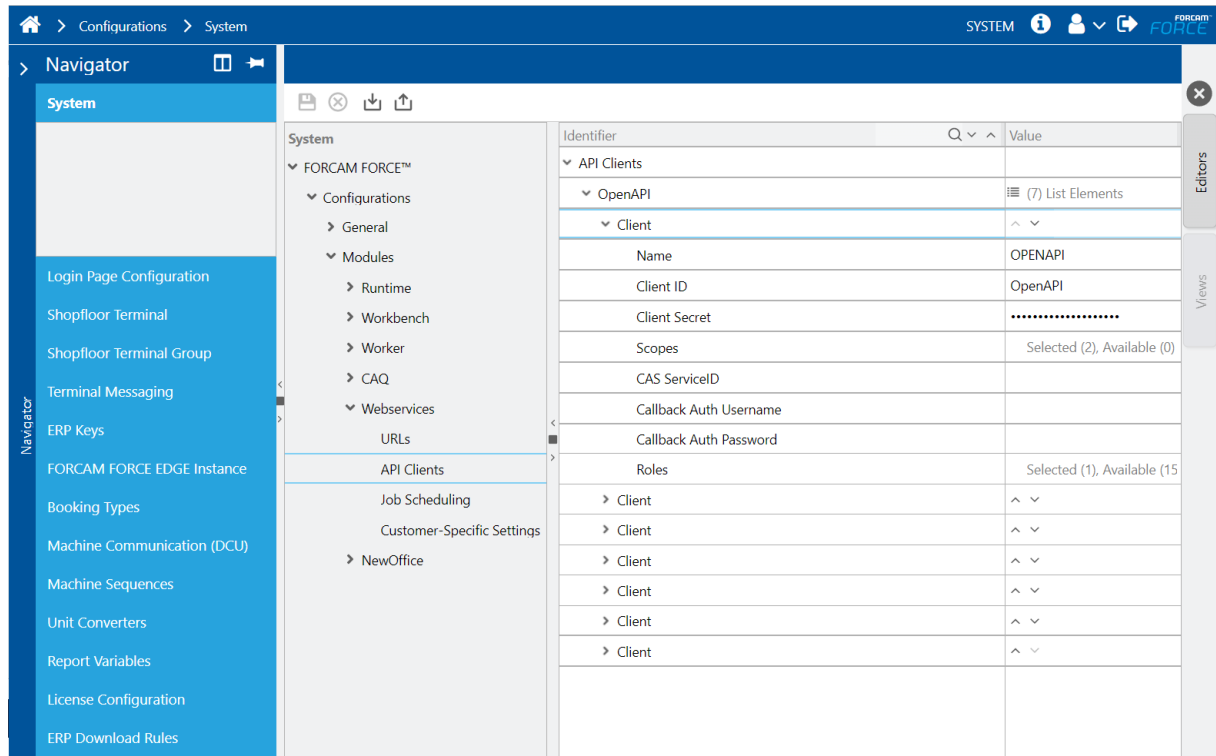


Fig. 2: Configuration of API clients

Designation	Explanation
Client	
Name	Descriptive name of the API client
Client ID	ID of the API client
Client Secret	Secret of the API client
Scopes	Scopes for which the API client has permissions (read / write)
CAS Service ID	CAS service ID when using OAuth 2 authorization grant
Callback Auth Username	Client-specific username for basic authentication for callbacks
Callback Auth Password	Client-specific password for basic authentication for callbacks
Roles	Roles of the API client

Table 7: Configuration of API clients

Role	Effect
ADMIN	Client access to everything except personnel data
CAQ	Client access to the CAQ interface
COMMAND_CALLBACK	Client access to the command and callback interface
CUSTOMIZED	Client access to the customized API. This is required for Shopfloor Connectivity (for more information, refer to the Manual - Shopfloor Connectivity).
ERP_DOWNLOAD	Client access to the ERP download interface (data flow from the ERP system to MES FLEX)
ERP_UPLOAD	Client access to the ERP upload interface (data flow from MES FLEX to the ERP system)
STAFF_MEMBER_MINIMAL	Client access to minimum personnel data information
STAFF_RESPONSIBILITY	Unrestricted client access to personnel data
TDM	Client access to TDM data
Entries with '_DOCS' suffix	Access to the interface description of the corresponding interface in the Swagger UI

Table 8: API client role descriptions

3.3 Job scheduling

Path: Configurations > Modules > Webservices > Job Scheduling

This section covers the configuration of jobs running in the background and communicating with external systems.

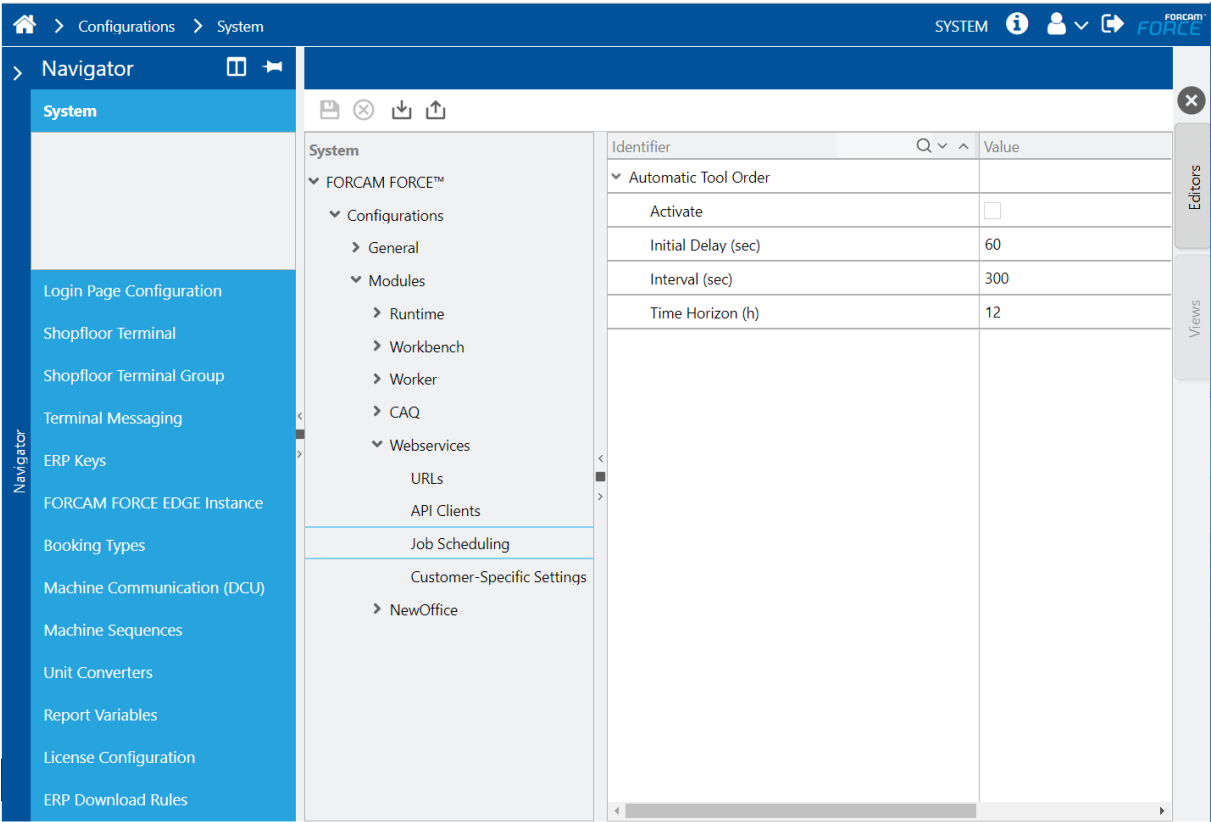


Fig. 3: Configuration of job scheduling

Designation	Explanation
Activate	Activate automatic tool commissioning with TDM
Initial Delay (sec)	Initial delay of automatic tool commissioning after starting ffwebservices
Interval (sec)	Interval in which automatic tool commissioning is executed
Time Horizon (h)	Time horizon for processes for automatic tool commissioning

Table 9: Configuration of job scheduling

3.4 Customer-specific settings

Path: Configurations > Modules > Webservices > Customer-Specific Settings

This section is provided to configure customer-specific settings made available via the corresponding API (*GET/masterData/customerSpecificSettings*).

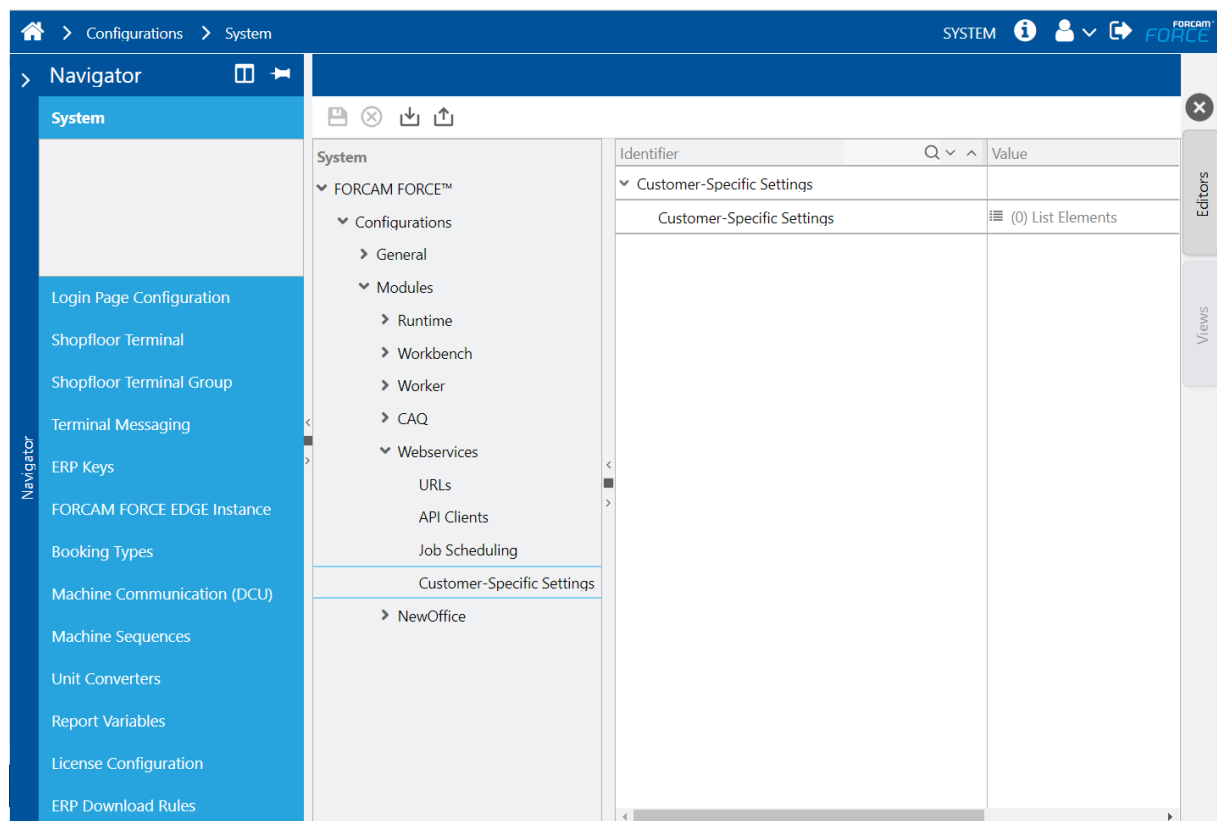


Fig. 4: Configuration of customer-specific settings

Designation	Explanation
UUID	Value for the 'id' of the customer-specific setting
Value	Value for the 'value' of the customer-specific setting

Topic	Value for the 'topic' of the customer-specific setting
Description	Value for the 'description' of the customer-specific setting

Table 10: Configuration of customer-specific settings

4 FORCE Bridge API as a digital image of a production site

This chapter describes the concepts and objects of a production site. It explains how they are represented in FORCE Bridge API and how they are accessed and used.

4.1 Basic concepts and objects used in production

The goal of the IoT is to create a digital image of the real world and its entities. These entities act as specific material objects on one hand and abstract concepts on the other. An IoT for the production domain generates a digital image of both the machines and tools used at the production plant and the concepts that govern everyday production such as, for example, production orders and operations.

4.1.1 Entities in the domain of production

Production orders

When a certain product is to be produced at a certain point in a specified quantity, the production planning system initiates a production order. The template for the production order is the routing, which contains each step (operation) required to produce the final product. The routing defines at which kind of workplace or machine the operation is to be carried out as well as how much time and which components are required.

The scenario for the creation of a production order is as follows:

- The company receives one or multiple customer orders to produce a certain product or wants to produce a certain quantity of it for the stock.
- A new production order with its own number is created in the ERP system which contains all the information from the routing.
- In addition to the information in the routing, the production order also contains the specification of the quantity of the product to be produced; this is called the target quantity.
- Finally, the delivery date is used to create a basic finish date for the production order and the result of material requirements planning is a basic start date. The start date corresponds to the time at which all of the components required for processing the production order are available.

Operations

An operation describes a single step that is required to process a production order. Each operation produces a specific material in a specified quantity (target quantity of the operation). This is either a material that is going to be processed further as a component in a subsequent operation or, in the case of a last operation, the end product of the production order.

The standard values stored in the operation include:

- The material to be produced.
- The target quantity of the material to be produced. This can differ from the target quantity of the production order (for example, 4 doors need to be produced in one operation in order to finally produce an automobile).
- The quantity unit of the target quantity (for example, piece, meter, kilogram, etc).

- The workplace or a group of equivalent workplaces at which the operation is to be performed.
- The target time per unit, i.e. the time required for the production of a unit of the material to be produced.
- The target processing time is the time specified for the entire processing time of the operation. It is the product of target quantity and target time per unit.
- The target setup time is the time scheduled for setting up the workplace.
- The target teardown time is the time scheduled for the teardown of the workplace.
- The target move time is the time scheduled for the transfer to the next workplace.
- The target wait time is the time scheduled for functions such as cooling, drying or curing of the material to be produced.
- The components needed to produce the material.
- The production resources and tools (e.g. tools, equipment or documents) that are needed at the workplace to execute the operation.

Operation phases

Operation phases describe the state of an operation from release by the ERP system to completion and reporting the output quantities of the operation back to the ERP system.

Workplaces

Each operation is executed at one workplace as a minimum. A workplace can be a simple manual workplace or a workplace with an automated production line (machine).

Workplace groups

Workplaces are often specified in ERP systems as a single workplace with a number of capacities. In FORCE Bridge API, each workplace is displayed individually and then assigned to a specific capacity group or production line, respectively.

- Capacity groups are workplaces at which the same production process is executed and which are therefore grouped together. For example, an injection molding workplace may belong to a capacity group named injection molding machines.
- Production lines are different workplaces that map a complete production process for a product. They use processes of the line production or mass production type. Here, the different production stations of a line are not interlinked by production cycle timing. Rather, buffers balance time differences occurring during processing between neighboring stations. Thus, for example, a faulty product can be removed from the production flow to be reworked without affecting the production process.

Each workplace can be assigned to exactly one capacity group and at most to one production line.

Staff members

The staff is the production personnel, which is allocated to shifts and can be assigned to the respective workplaces. Apart from their personnel data, employees are specified with additional information like e.g. qualifications, posted vacation, etc. that is important for scheduling and the production flow.

Tools

Production tools are needed at machine workplaces to properly complete an operation. A tool, for example, may be the die casting mold of an injection molding system. The tool has a storage location, a current place of use and a lifespan. The lifespan describes how much time is left until the tool must be replaced.

Documents

The documents that are required for the execution of an operation are stored in production folders. These documents usually originate from the product life cycle system. The documents may be NC programs for controlling machine tools or other documents such as test plans, assembly instructions or assembly sketches. If documents are changed, the IoT platform automatically creates a new version.

All documents belonging to an operation are displayed in the Shop Floor Terminal. From there, the NC programs from the production folder can also be loaded into the machine control.

Devices

Device refers to all devices that are connected to the IoT platform. These include in particular the programmable logic controllers (PLC) of the connected machines and production plants. These are permanently assigned to a specific workplace.

You can use *GET/workplaces/{workplaceId}/devices* to call a list of all devices that are available at a workplace or machine. All process and sensor data recorded can be queried for the devices.

4.1.2 Resources of the FORCE Bridge API

FORCE Bridge API maps the complete digital image of a production plant. The RESTful² webservice can be called directly via a web browser to interact with the resources provided in the API.

Operation	>
Production Order	>
Workplace	>
Workplace Group	>
Staff Member	>
Tool	>
Device	>
Document	∨
<div> <div>GET</div> <div>/api/v1/documents/folder/ Get all folders</div> <div>🔒</div> </div>	
<div> <div>POST</div> <div>/api/v1/documents/folder/ Create a folder</div> <div>🔒</div> </div>	
<div> <div>GET</div> <div>/api/v1/documents/folder/{folderId} Get a folder</div> <div>🔒</div> </div>	

Fig. 5: Accessing API resources through the Swagger UI

For a technical description of the interfaces offered in FORCE Bridge API, refer to [SwaggerHub](#).

Entry point and top-level resources

RESTful usually has only one *entry point* (API base) from which all other outgoing resources can be retrieved. In order to be able to find the API resources, it is therefore important that the uniform resource indicator (URI) is correctly communicated with the REST clients.

This chapter covers the following top-level resources in further detail:

- *productionOrders/* Production orders
- *operations/* Operations of the production orders
- *workplaces/* Workplaces (machines or manual workplaces) where operations are performed
- *workplaceGroups/* Capacity groups (functionally similar machines) or production lines
- *staffMembers/* The personnel available for performing operations
- *tools/* The tools available for performing operations
- *documents/* Documents that are important for the entire production plant
- *devices/* Devices such as programmable logic controllers of machines

² RESTful refers to applications or web services offering states and functions as a set of resources and allow interoperability between computer systems on the Internet.

Creating / editing / deleting data via the FORCE Bridge API

The following resources can be created via the FORCE Bridge API:

- *productionOrders*/ Production orders
- *operations*/ Operations of the production orders
- *staffMembers*/ The personnel available for performing operations
- *workplaces/shifts*/ Workplace shifts
- *materials*/ Materials of orders and operations

In this context, operations are sub-resources of production orders.

However, it is also possible to create, edit or delete orders and operations separately in individual transactions. These transactions are performed synchronously in such a way that the response reveals whether the transaction was successful.

References to other resources such as ERP context and material must be specified by UUID in the request payload. Such resources need to exist already or be created in advance.

4.2 Shifts and scheduled operating times of workplaces

4.2.1 Domain-specific knowledge

Workplace shifts

The term 'shift work' describes a work structure where employees work on the same job in succession and according to a specific schedule, so that they are scheduled to do their job within a specific timeframe at different times. A plant will use shift work when it is necessary to work longer than the usual daily work hours or when operations or on-call duties are needed outside the usual daily work hours for other reasons.

Shifts are defined for workplaces: For instance, in three-shift operation, these are usually morning shift, late shift and night shift. This concept describes a so-called shift model.

If one of the shifts should be omitted on a certain day, the shift is marked as an idle period or nonworking shift. The other shifts are working shifts. Shift breaks, during which production will be halted, can be defined within a working shift.

Scheduled maintenance times

These are times during which machine workplaces will be serviced. This does not apply to manual workplaces. The goal of scheduled maintenance times is to reduce disruptions, so that machines do not unexpectedly stop working, which would consequently trigger an unscheduled maintenance action. These times usually take place at regular maintenance intervals.

Scheduled operating times

The scheduled operating time is the period in which a workplace should be in production according to shift planning and maintenance planning. This includes all working shifts minus shift breaks and scheduled maintenance.

4.2.2 Workplace resources in the FORCE Bridge API

The above mentioned objects form part of the sub-resources of workplaces or a specific workplace.

Workplace			^
GET	/workplaces	Retrieve the collection of workplaces.	✓ 🔒
GET	/workplaces/shifts	Retrieve the collection of workplace shifts.	✓ 🔒
GET	/workplaces/shifts/{shiftId}	Retrieve a workplace shift.	✓ 🔒
GET	/workplaces/{workplaceId}	Retrieve a workplace.	✓ 🔒
GET	/workplaces/{workplaceId}/customFields	Request workplace custom fields	✓ 🔒
GET	/workplaces/{workplaceId}/customFields/{customFieldId}	Request workplace specific custom field	✓ 🔒
GET	/workplaces/{workplaceId}/devices	Retrieve the devices that belong to a workplace.	✓ 🔒
GET	/workplaces/{workplaceId}/operatingStateSummary	Retrieve the operating state summary of a workplace.	✓ 🔒
GET	/workplaces/{workplaceId}/quantitySummary	Retrieve the quantity summary of a workplace.	✓ 🔒
GET	/workplaces/{workplaceId}/recordedOperatingStates	Retrieve the recorded operating states of a workplace.	✓ 🔒
GET	/workplaces/{workplaceId}/recordedOperationPhases	Retrieve the recorded operation phases of a workplace and their durations.	✓ 🔒
GET	/workplaces/{workplaceId}/recordedOutputQuantities	Retrieve the recorded output quantities of a workplace.	✓ 🔒
GET	/workplaces/{workplaceId}/scheduledMaintenanceTimes	Retrieve the scheduled maintenance times of a workplace.	✓ 🔒
GET	/workplaces/{workplaceId}/scheduledOperatingTimes	Retrieve the originally scheduled operating times of a workplace.	✓ 🔒

Fig. 6: Workplace resources as shown in the Swagger UI

GET workplaces/shifts

Retrieve shifts of a workplace.

GET workplaces/shifts/{shiftId}

Retrieve details for a certain shift.

GET workplace/{workplaceId}/scheduledMaintenanceTimes

Retrieve the scheduled maintenance times of a workplace.

GET workplaces/{workplaceId}/scheduledOperatingTimes

Retrieve workplace scheduled operating times. The digital image in FORCE Bridge API.


Operating states and workplace status

Operating states are defined in order to determine the current state of a workplace. Possible operating states of a workplace are, for example, production, machine malfunction, tool damage, etc.

Each operating state has a workplace status to determine if a workplace is currently working or stopped. They are defined as follows.

Workplace status	Description
DOWNTIME	The workplace stands still.
PRODUCTION	The workplace is producing.

Table 11: Workplace status

-  Operating states can be configured freely in MES FLEX. In this way, each workplace can have individually customized operating states.

Operation phases

Between the release of an operation and its completion including final confirmation to the ERP system, the following phases occur in production. These are either reported on the shop floor terminal or updated automatically by MES FLEX.

Phase	Description
RELEASED	The production order an operation belongs to has been released by the ERP system, but the operation has not yet been started or scheduled in detail.
DISPATCHED	The operation has been dispatched to a specific workplace.
SETUP	The operation has been logged on to a workplace and this workplace is set up for processing the operation.
PROCESSING	The operation is being processed.
INTERRUPTED	The operation was interrupted.
COMPLETED	The operation has been completed.
CLOSED	The operation was completed and a final confirmation was sent to the ERP system.

Table 12: Operation phases

Plant data collection

Recorded time series

GET operations/{operationId}/recordedOperatingStates

Contains all operating states which occurred during the execution of the operation.

GET operations/{operationId}/recordedOperationPhases

Contains all operation phases which occurred during the execution of the operation.

GET operations/{operationId}/recordedOutputQuantities

Contains all output quantities (yield, rework, scrap) that were produced during the execution of the operation.

Condensed values

GET operations/{operationId}/recordedOperatingStates

GET operations/{operationId}/recordedOperationPhases

Machine data collection

Actual values

GET workplaces/{workplaceId}/recordedOperatingStates

Contains all operating states which occurred at the workplace.

GET workplaces/{workplaceId}/recordedOperationPhases

Contains all operation phases which occurred at the workplace.


GET workplaces/{workplaceId}/recordedOutputQuantities

Contains all output quantities (yield, rework, scrap) that were produced at the workplace.

Condensed values

GET workplaces/{workplaceId}/recordedOperatingStates

GET workplaces/{workplaceId}/recordedOperationPhases

 Each resource contains the actual values or – for the summaries – the condensed actual values that are used to calculate the KPIs.

5 Callbacks and events

The FORCE Bridge API allows applications and third-party systems to be notified about events via HTTP or MQTT. This requires registration. To do this, a callback (also referred to as a webhook) must be created via POST, specifying the type of event about which the application or third-party system is to be notified and the URL at which it can be reached.

For notification via MQTT, a topic has to be defined that must always begin with *external* in order to be distinguished from the topics used internally in MES FLEX:

```
"url": "mqtt://{host}:1883/external/{any topic}"
```

Third-party systems that are to be informed about events not via MQTT but via HTTP require their own HTTP server. In this case, the callback itself is an HTTP POST request with JSON data that is sent to the URL of the HTTP server to be specified as follows:

```
"url": "http://{host}:{port}/{path}"
```

5.1 Callbacks

5.1.1 Callback request to the external webservice

Whenever an event which matches an existing registration occurs, the event data are sent via a HTTP POST request to the URL specified.

5.1.2 Retries

If a callback fails (for example, if the system which provides the registered endpoint is down), retries will be triggered every 10 seconds until the maximum number of retries is reached. When the maximum number of retries is reached, the callback will be discarded and no further retries for this callback will be sent. Retries implicate that callbacks might not be sent in the correct order. Therefore the timestamp of the event which triggered the callback has to be considered by the receiving system. When the maximum number of failed callbacks is reached, the oldest one will be discarded even if the maximum number of retries is not reached yet.

5.1.3 Creating a callback

A callback must be created to be notified about events. The callback is registered for a specific EventType (e.g. ERP) to receive all messages of this type. Additional filtering by the EventName is possible, if a notification about a certain event of an EventType should be made.

A callback is created by executing POST callbacks with the following content in the request body:

```


6. {
7.   "name": "string",
8.   "objectFilter": [
9.     {
10.      "name": "string",
11.      "value": "string"
12.    }
13.  ],
14.  "eventType": "COMMAND",
15.  "eventName": "string",
16.  "url": "string",
17.  "maxRedeliveryAttempts": 100,
18.  "maxUnconfirmedMessages": 5000
19. }

```

The *maxRedeliveryAttempts* parameter defines the number of redelivery attempts if callback notifications fail.

The *maxUnconfirmedMessages* parameter is used for setting the maximum number of failed callback messages. When the maximum number of failed messages is reached, the oldest one will be discarded even if the maximum number of retries for the message is not reached yet.


A callback can be registered with *maxUnconfirmedMessages = 0* to ensure that no messages are lost. Internally, messages are sent to a dynamic MQTT queue which saves all messages before they are sent to the final destination.

 For callbacks with *maxUnconfirmedMessages = 0*, it is necessary to ensure that they will finally be delivered some time, otherwise messages will accumulate in the persistent queue and sooner or later cause the message broker to stop.

 Deleting a callback will also delete all associated queues.

For the *maxUnconfirmedMessages = 0* setting, the following rules apply for retries:

- By default, the redelivery function will start 10 seconds after a failed delivery or if delivery takes longer than 10 seconds.
- Retries are initiated with an exponential backoff (beginning at 200 ms up to 10 seconds by default).
- Messages will still be removed from the queue when *maxRedeliveryAttempts* has been reached.
- Aborting the redelivery function is possible in one of two ways:
 - Deleting the callback
 - Restarting the ffwebservice application

 If a callback is registered with *maxUnconfirmedMessages = 0* and *maxRedeliveryAttempts = 0*, message delivery will be retried without limit. If this setting is used, the order of messages is kept also in the case of transmission errors.

The request must include at least two specifications:

- Event type about which to inform
- URL to which the information about an event is to be sent

Events that refer to a particular type of primary resource can be restricted to a subset or a single primary resource when creating the callback. For this purpose, object filters are used that are passed as an array consisting of name-value pairs.

Name specifies a property of the reference object.

Value is a value that the property must have in order for the filter criterion to apply. In the current version, only the *ID* property is supported, i.e. the reference objects of an event type can only be filtered by UUIDs.

Examples

CURL example using HTTP

```
1. curl -X POST http://$HOST:$PORT/ffwebservice/api/v2/callbacks \
2.   --header "Content-Type: application/json" \
3.   --header "Accept: application/json" \
4.   --header "Authorization: Bearer $TOKEN" \
5.   -d '{
6.     "objectFilter": [
7.       {
8.         "name": "id",
9.         "value": "DA892519507444768080F0E81879513"
10.      }
11.    ],
12.    "eventName": "Operation 100 has changed phase",
13.    "eventType": "OPERATION_PHASE_CHANGED",
14.    "url": "http://third-party-system:24080/webserver"
15.  }
```

CURL example using MQTT

```
1. curl -X POST http://$HOST:$PORT/ffwebservice/api/v2/callbacks \
2.   --header "Content-Type: application/json" \
3.   --header "Accept: application/json" \
4.   --header "Authorization: Bearer $TOKEN" \
5.   -d '{
6.     "objectFilter": [
7.       {
8.         "name": "id",
9.         "value": "DA892519507444768080F0E81879513"
10.      }
11.    ],
12.    "eventName": "Operation 100 has changed phase",
13.    "eventType": "OPERATION_PHASE_CHANGED",
14.    "url": "mqtt://third-party-system:1883/external/OperationPhaseChanged"
15.  }
```

5.1.4 Get all registered callbacks

To get all registered callbacks, a request must be send to *callbacks/* via *GET*.

Example CURL

```
1. curl -X GET "http://$HOST:$PORT/ffwebservice/api/v1/callbacks/" \
2.   --header "Content-Type: application/json" \
3.   --header "Accept: application/json" \
```

5.1.5 Delete a registered callback

To delete a registered callback, a request must be sent via *callbacks/{callbackId}* via *DELETE* , where *callbackId* is the *ID* of the callback.

Example CURL

```
1. curl -X DELETE "http://$HOST:$PORT/ffwebservice/api/v1/callbacks/$CALLBACK_UUID" \
2.   --header "Content-Type: application/json" \
3.   --header "Accept: application/json" \
```

5.1.6 Event types

A list of all available callbacks can be found in Swagger at </ffwebservices/docs/index.html#callback-api> and commands can be found at </ffwebservices/docs/index.html#command-api>.

Swagger lists all the event types for which a callback registration is possible. They are provided as condensed information by MEX FLEX as a result of certain other individual events. In addition, numerous events that inform about details are generated.

The event types *COMMAND*, *SHOP_FLOOR_TERMINAL* and *INTERNAL* are reserved for internal events of MES FLEX.

5.1.7 Receiving events

If an event occurs (*OPERATION_PHASE_CHANGED* in the following example) for which a third-party system has registered via HTTP callback, the HTTP server of the third-party system receives, for example, the following POST request:

```
<curl -X POST
--header 'Content-Type: application/json'
-d '{
  "properties" : {
    "callbackId" : "B1B1BFB60ECF4DB7B095F096992FB9FA",
    "timestamp" : "2019-03-06T11:04:16.786Z",
    "data" : {
      "currentOperationPhase" : "PROCESSING",
      "previousOperationPhase" : "INTERRUPTED",
      "operationId" : "2A05AAEF15FD41A8A4A36C45516871AA"
    },
    "objectId" : "DA892519507444768080F0E81879513",
    "objectType" : "OPERATION",
    "eventType" : "OPERATION_PHASE_CHANGED",
    "eventName" : "Operation 100 has changed phase"
  },
  "_links" : {
    "callback" : {
      "method" : "GET",
      "embeddable" : true,
      "href" : "http://localhost:24080/ffwebservices/api/v2/callbacks/B1B1BFB60ECF4DB7B095F096992FB9FA"
    }
  }
}' 'CALLBACK_URL'>
```

It is possible to configure Basic Auth authentication for callbacks. If an event occurs for which an application has registered via MQTT, the application receives an MQTT message.

5.1.8 Sending self-defined events to applications or third-party systems

An application can generate events that can be received by other applications or third-party systems via callback. To do this, POST events must be called with the following content in the request body:

```
{
  "data": {},
  "eventName": "string",
  "eventType": "EXTERNAL",
  "objectId": "string",
  "objectType": "OPERATION",
  "timestamp": "2019-02-23T17:42:59.429Z"}
```

The event type must always be EXTERNAL in the request body.

The object type may be blank or have one of the following values:

- *OPERATION*
- *PRODUCTION_ORDER*
- *WORKPLACE, STAFF_MEMBER*
- *TOOL*

Any amount of information can be transferred with the event in the data field. The event name and the object ID are optional.

Example CURL

```
1. curl -X POST https://$HOST:$PORT/ffwebservice/api/v2/events"
2.
3.   -H "accept: */*"
4.   -H "authorization: Bearer $TOKEN
5.   -H "Content-Type: application/json"
6.   -d "{
7.       \"data\": {\"myData1\" : \"123\", \"myData2\" : \"456\"},
8.       \"eventName\": \"MY_EVENT\",
9.       \"eventType\": \"EXTERNAL\",
10.      \"objectId\": \"4DDE054C036C4F8CAB424C7FB70E86B9\",
11.      \"objectType\": \"TOOL\",
12.      \"timestamp\": \"2019-03-01T07:24:37.756Z\"
13.    }
```

6 ERP interface – data and events via XML

There are several options, or alternatives, for the type of communication between MES FLEX and an ERP system.

One method is communication directly via the Runtime of MES FLEX using the open and standardized XSL transformation (XSLT) method. XSLT forms part of XSL (Extensible Stylesheet Language) and performs an application-specific transformation of the data structures transmitted. This method is used, for example, when connecting ERP systems exchanging data in the SAP IDoc format and is described in more detail in the manual on **ERP Interface – Data and Events via SAP IDoc**.

A more advanced, safer and much more reliable method is the alternative connection via the FORCE Bridge API programming interface provided by MES FLEX as a RESTful interface. This variant has been described in the previous chapters and provides the basis for using the ERP interface.

Both methods involve data records being exchanged in XML format via the HTTP or HTTPS protocol. This involves using a http POST request with the XML datasets as payload.

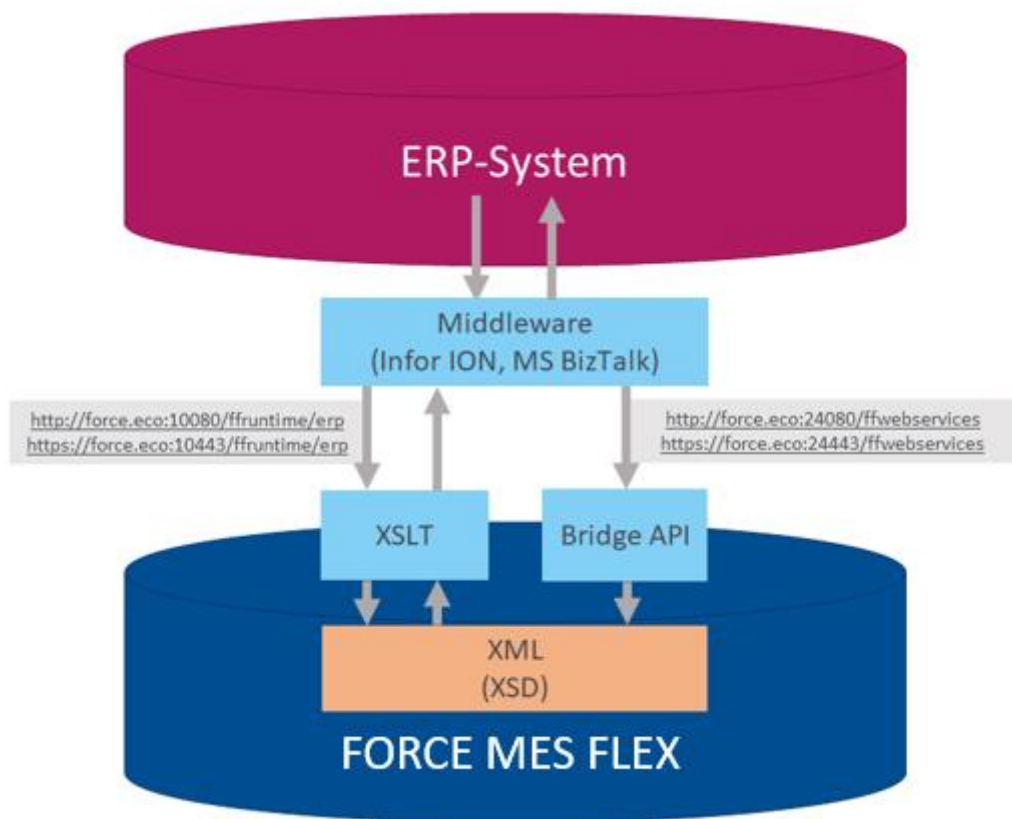


Fig. 7: Alternative connections between the ERP system and FORCE MES FLEX

In addition, middleware such as Infor ION or Microsoft BIZ Talk can be used to connect to the ERP. Their functionality can be used to transform the required datasets into the internal XML structure of MES FLEX.

7 ERP download and upload services

A download and an upload data service are available for communication between MES FLEX and the connected ERP system.

The download service is used to supply the MES FLEX Rule Engine with master data and order data.

The incoming data (data message stream) are processed asynchronously.

The upload data service is used for the confirmation of quantities, times and states (status messages) of orders (event message stream). For durations and quantities, the absolute values are transmitted here in addition to the relative changes.

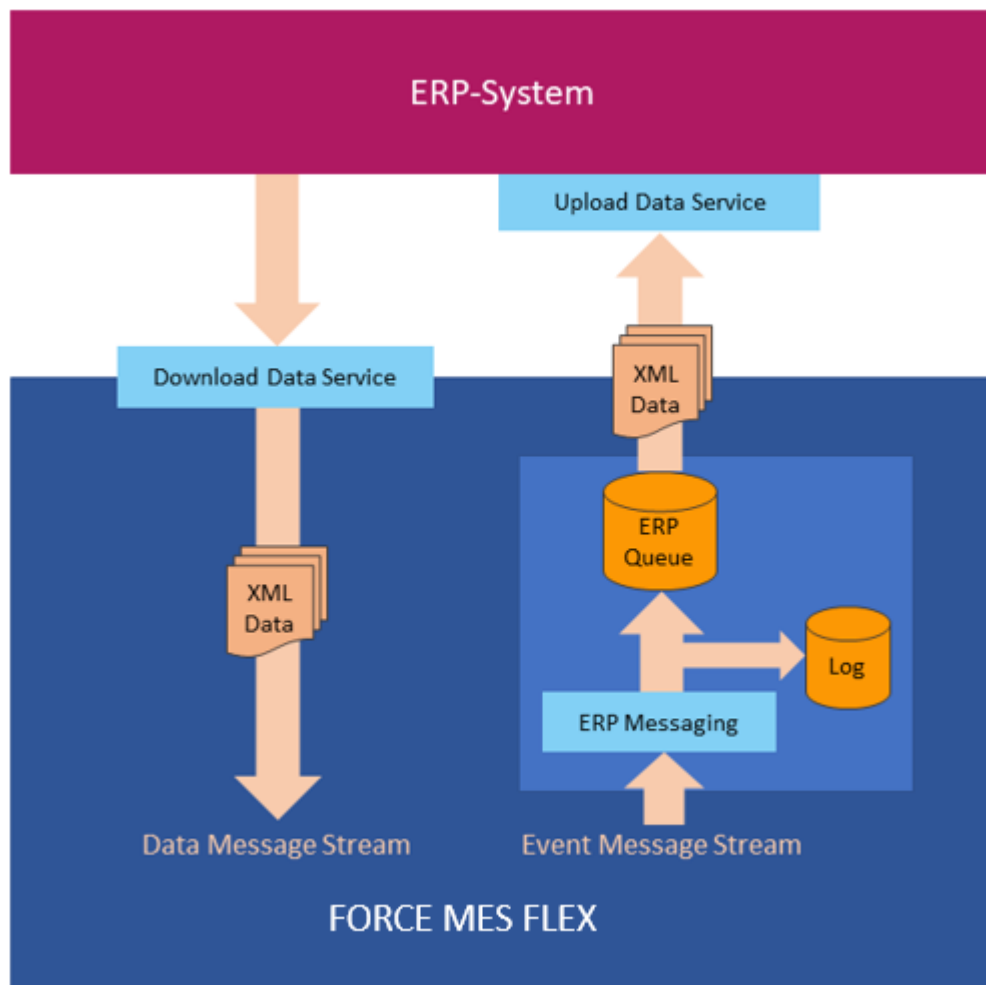


Fig. 8: ERP download and upload services

In the upload data service, persistent queues are used to prevent any data loss in case of a system failure. Only messages transferred successfully to the ERP are acknowledged towards the queue. This ensures fail-safe operation. Any failed transmission attempts remain in the queue.

Traceability of messages communicated is ensured by a log file which is continuously updated.

8 ERP interface configuration

Whenever data are to be transferred via the FORCE Bridge API, a user with password and the required ERP user roles must be configured for the API client connection in the Workbench.

8.1 ERP download

Path: Configurations > Modules > Runtime > ERP > ERP Download

This configuration controls the mapping of messages of an ERP system. In this process, the specific SAP IDOC format is converted to the generic ERP XML format. The conversion is done via XSLT.

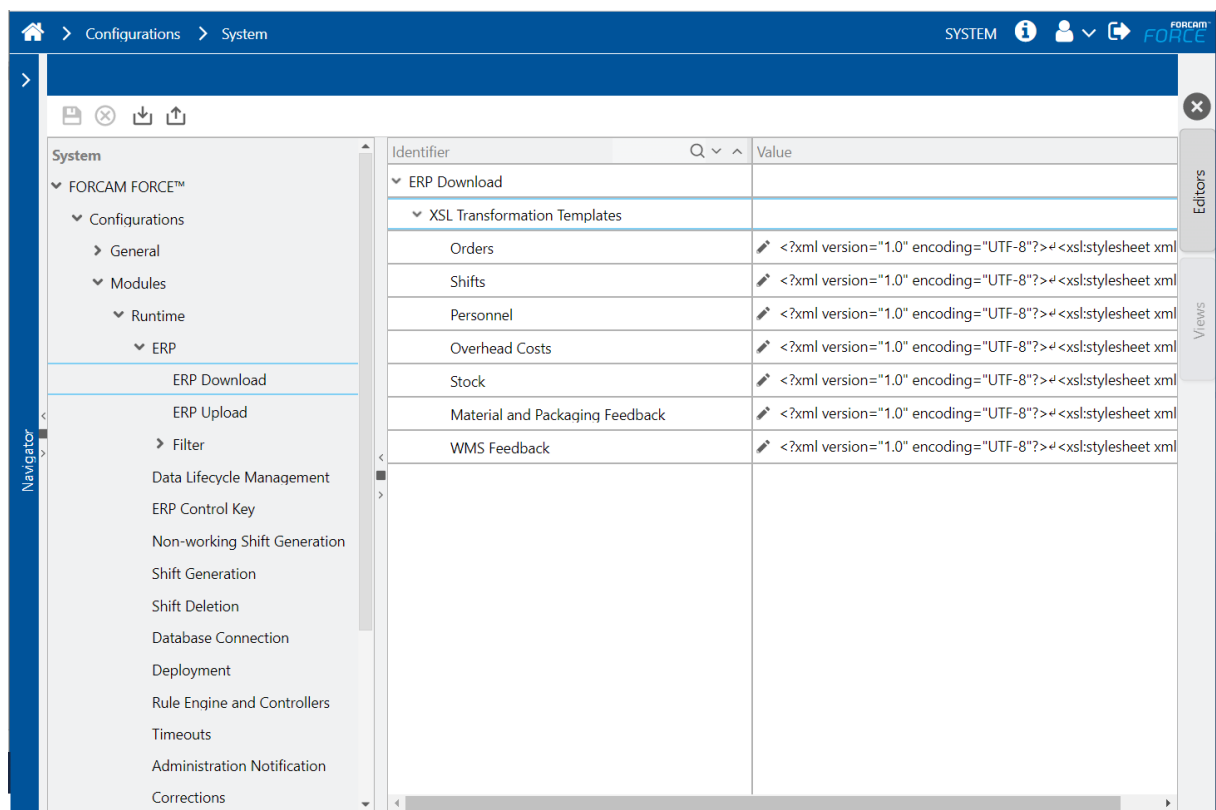


Fig. 9: ERP download configuration

Designation	Explanation
validation	Defines whether to validate incoming data according to the XSD pattern and the associated action (ADD, DELETE). If activated, invalid data will be rejected and not processed.
XSL Transformation Templates	
Orders	XSLT defining the mapping of IDOC to ERP XML for orders
Shifts	XSLT defining the mapping of IDOC to ERP XML for shifts
Personnel	XSLT defining the mapping of IDOC to ERP XML for personnel

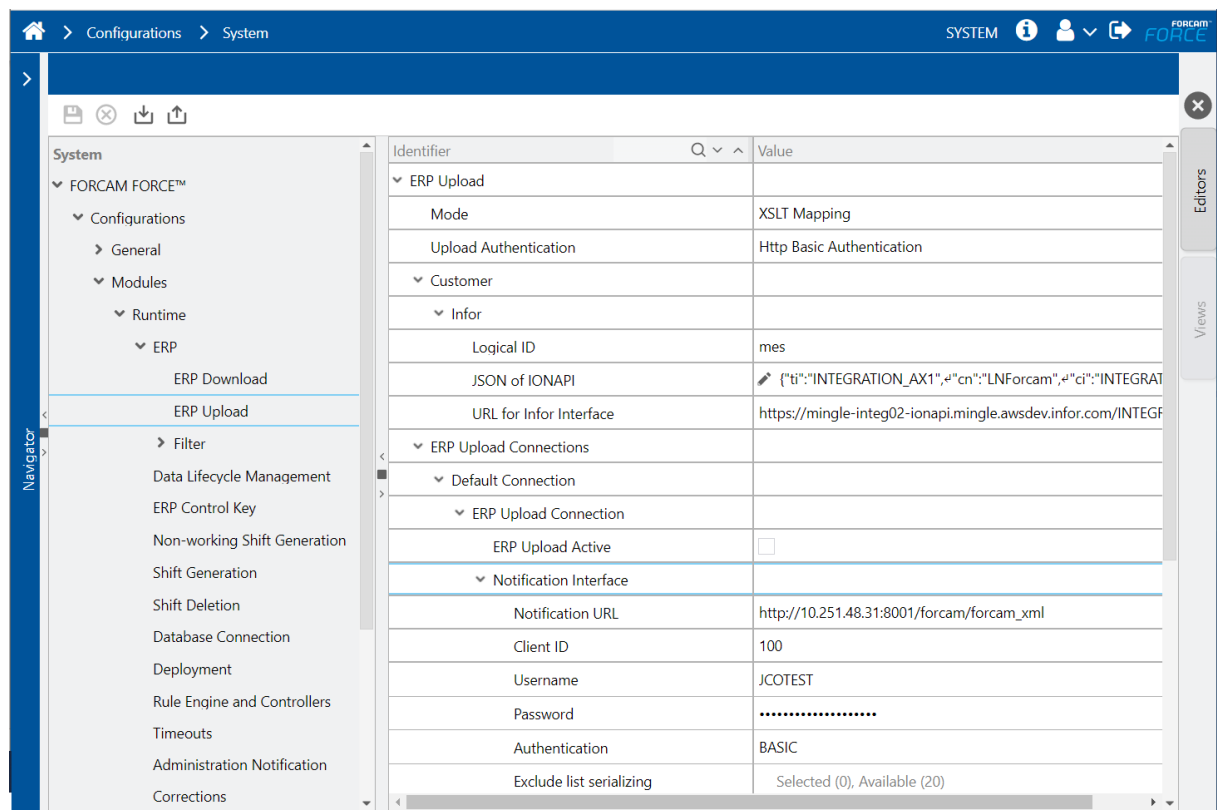
Overhead Costs	XSLT defining the mapping of IDOC to ERP XML for overhead costs
Stock	XSLT defining the mapping of IDOC to ERP XML for stock
WMS Feedback	XSLT defining the mapping of IDOC to ERP XML for WMS Feedback
Material and Packaging Feedback	XSLT defining the mapping of IDOC to ERP XML for material and packaging units

Table 13: ERP download configuration

8.2 ERP upload

Path: Configurations > Modules > Runtime > ERP > ERP Upload

This configuration sets the connection parameters to ERP systems for ERP upload.



Identifier	Value
ERP Upload	
Mode	XSLT Mapping
Upload Authentication	Http Basic Authentication
Customer	
Infor	
Logical ID	mes
JSON of IONAPI	{ "ti": "INTEGRATION_AX1", "cn": "LNForcam", "ci": "INTEGRAT
URL for Infor Interface	https://mingle-integ02-ionapi.mingle.awsdev.infor.com/INTEGF
ERP Upload Connections	
Default Connection	
ERP Upload Connection	
ERP Upload Active	<input type="checkbox"/>
Notification Interface	
Notification URL	http://10.251.48.31:8001/forcam/forcam_xml
Client ID	100
Username	JCOTEST
Password
Authentication	BASIC
Exclude list serializing	Selected (0), Available (20)

Fig. 10: ERP upload configuration

Designation	Explanation
Mode	XSLT mapping for SAP IDOC upload. Generic for XML upload.
Upload Authentication	Http Basic Authentication Infor-specific authentication according to IONAPI definition
Customer	
Infor: Logical ID	Logical ID for Infor integration

Infor: JSON of IONAPI	IONAPI JSON for Infor integration
Infor: URL for Infor Interface	URL of the upload endpoint for Infor integration

Table 14: ERP upload configuration

8.2.1 ERP upload connection configuration

The *standard connection* initially applies to all ERP keys.

Designation	Explanation
ERP Upload Active	Flag: Send ERP messages
Notification Interface	
Notification URL	URL of the upload endpoint for standard ERP integration
Client ID	Client ID transmitted for SAP integration in the IDOC header as a value for MANDT.
Username	Username for basic authentication
Password	Password for basic authentication
Exclusion List	IDOCs listed are not serialized
IDOC Type	Client ID transmitted for SAP integration in the IDOC header as a value for IDOCTYP.
MES Type	Client ID transmitted for SAP integration in the IDOC header as a value for MESTYP.
Sender Port	Client ID transmitted for SAP integration in the IDOC header as a value for SNDPOR.
Sender Type	Client ID transmitted for SAP integration in the IDOC header as a value for SNDPRT.
Sender Number	Client ID transmitted for SAP integration in the IDOC header as a value for SNDPRN.
Receiver Port	Client ID transmitted for SAP integration in the IDOC header as a value for RCVPOR.
Receiver Type	Client ID transmitted for SAP integration in the IDOC header as a value for RCVPRT.
Receiver Number	Client ID transmitted for SAP integration in the IDOC header as a value for RCVPNR.
IDOC Template	XSLT used for generating the IDOC from ERP XML
Serialize Interface	
Serialization Active	Flag: Serialization active
Serialize Mode	Operation-based serialization or workplace-based serialization
URL to synchronise the serialization	URL for requesting the channel/counter numbers from SAP via SOAP request
Username	Username for basic authentication for requesting the channel/counter numbers from SAP via SOAP request
Password	Password for basic authentication for requesting the channel/counter numbers from SAP via SOAP request
Authentication	Authentication mode used (BASIC, NONE) for requesting the channel/counter numbers from SAP via SOAP request
Synchronise SOAP-Request Template	Template for requesting the channel/counter numbers from SAP via SOAP request

Designation	Explanation
Least Channel Number (Inclusive)	Lowest channel number for requesting the channel/counter numbers from SAP via SOAP request
Greatest Channel Number (Inclusive)	Highest channel number for requesting the channel/counter numbers from SAP via SOAP request
Greatest Counter Number	Highest counter number used for serialization
Prefix	Prefix used for serial numbers

Table 15: ERP upload connection configuration

Other connections may be defined for specific ERP keys.

Designation	Explanation
Client	Client for whom the connection is defined.
Company Code	Company code for which the connection is defined.
Plant	Plant for which the connection is defined.
System ID	System ID for which the connection is defined.

Table 16: Configuration of other connections

8.3 Filter – Unused closed operations

Path: Configurations > Modules > Runtime > ERP > Filter > Unused Closed Operations

This setting controls a filter which permits changing the technically closed operations if their last status change occurred within the number of days specified.

Example: If an operation was closed two days ago and the value specified in the filter is 2 days, the operation cannot be changed in the ERP. At least, processing is rejected at the interface between MES FLEX and ERP.

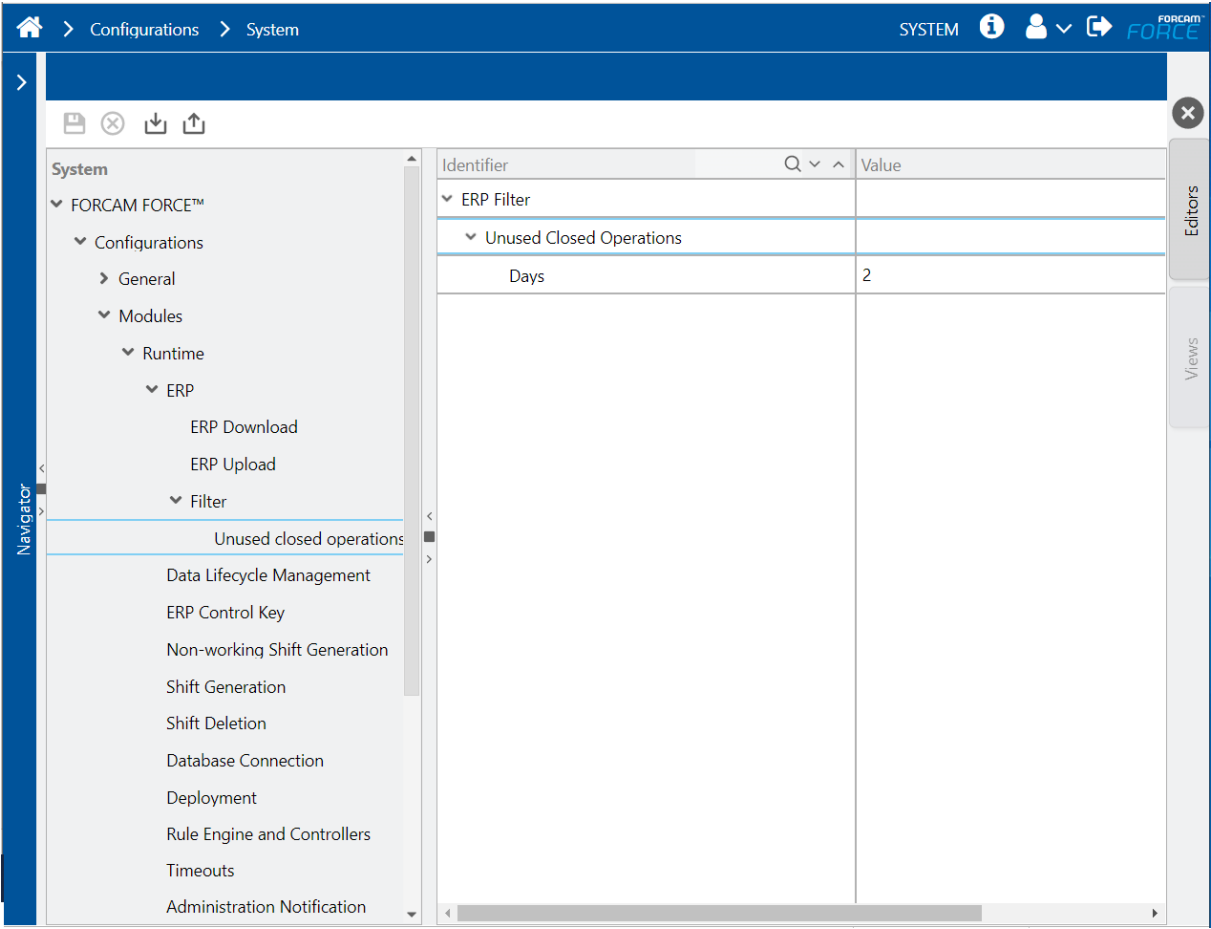


Fig. 11: Filter configuration for unused closed operations

Designation	Explanation
Days	Number of days in the past for which technically closed operations can be updated from the ERP.

Table 17: Configuration of ERP filter

8.4 ERP control key

Path: Configurations > Modules > Runtime > ERP Control Key

Every operation has an associated control key. If a change is made to an operation, a DomainAttributeChangeCommand is generated for the control key. This command is used for sending an OperationRescheduleERPDTO to the ERP system. The configuration controls whether this function is active.

8.5 Connection via Infor ION

If Infor ION is used for a connection, the following configuration must be set in the Workbench of MES FLEX:

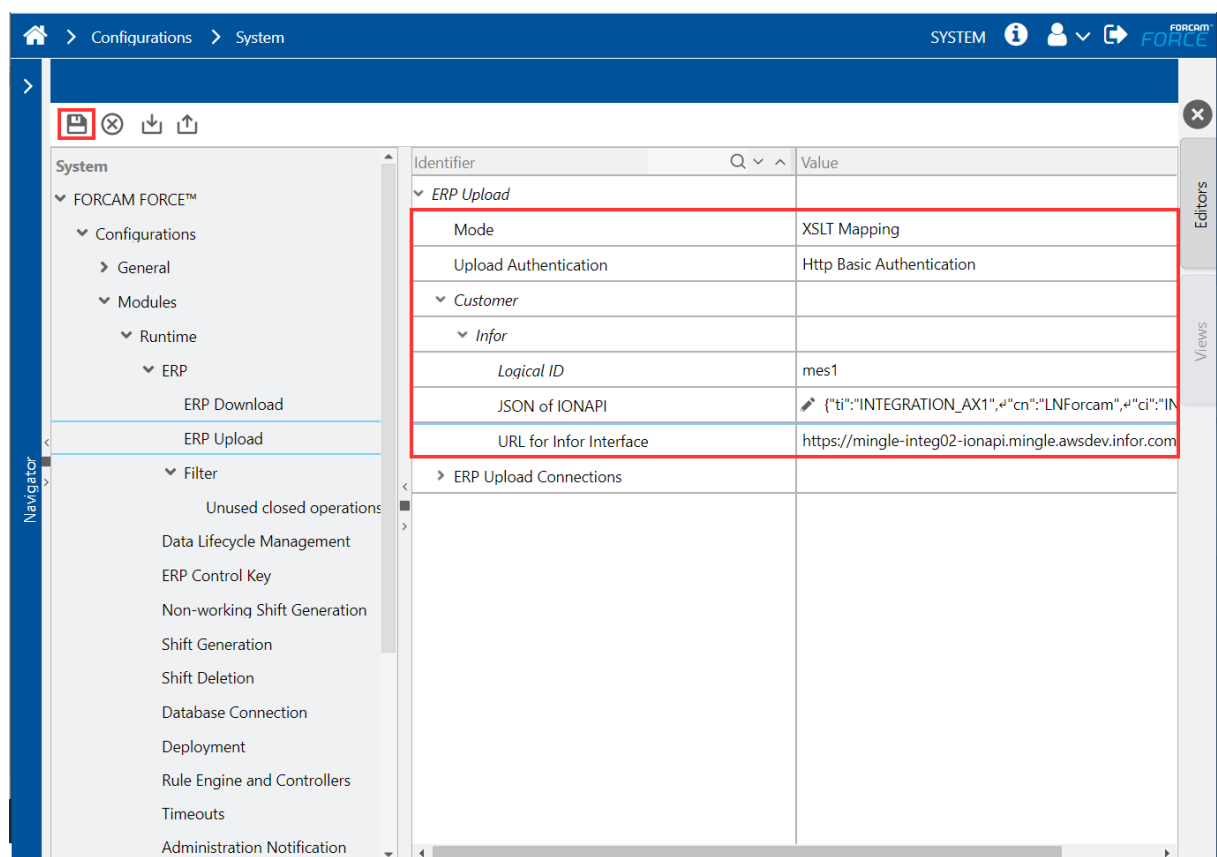


Fig. 12: Configuration of the upload data service for Infor ION

To configure the ERP upload data service for Infor ION:

1. Set **Mode** to **Generic**.
2. Set **Upload Authentication** to **Infor**.
3. Enter the **Logical ID** predefined by the Infor ERP.

4. Insert the **JSON of IONAPI** document predefined by the Infor ERP.
5. Enter the **Upload URL for Infor interface**.
6. Save the data.

8.6 Assigning operations to workplace groups

Usually individual operations are assigned to a specific workplace within a production order using the **TargetWorkplace** attribute. However, MES FLEX offers an additional extended function for dynamic assignment of operations to workplaces within a group.

Workplaces can be arranged hierarchically in the Workbench using the **Workplace Hierarchy** function (for more information, refer to the manual on **Master Data and System Configuration**).

The ERP hierarchy can be utilized for dynamic assignment of the operations. At the lowest level, workplaces can be arranged in a workplace group. If workplaces have the same function or capacity within such a group, the group is also referred to as a capacity group. You can define for such a capacity group how dynamic assignment of operations of orders released should be done.

Procedure	Description
Auto-Split	Operations are distributed to the workplaces within the capacity group based on the target quantity to be produced (operation split). Every workplace should get the same share of the target quantity to be produced as far as possible. If the target quantity cannot be divided by the number of workplaces, the remaining amount is distributed to individual workplaces. For example, a target quantity of 100 pieces distributed to 3 workplaces results in a split of 33 + 33 + 34 pieces.
Pull	The operations are initially available to all workplaces within the capacity group. An operation can then be assigned manually (e.g. by a foreman) to one workplace within the capacity group or the first workplace that wants to begin processing may assign the operation to itself. The operation assigned disappears from the shopfloor display of all other workplaces. If an operation assigned is aborted, it becomes available again to all other workplaces and the process begins again.

Table 18: Procedure for assigning operations in a capacity group

The assignment procedure can be selected accordingly in the Workbench.

Path: Master Data > Workplace > Workplace Hierarchy > Hierarchy Tree > ERP – ERP

To assign operations to a workplace group with the Auto-Split or Pull procedure:

1. Select the corresponding node at the lowest level of the hierarchy tree.
2. Open the drop-down menu in the **Attributes** column.
- A dialog window opens.
3. For the **Behaviour of Split** attribute, select **Auto-Split**.
or
For the **Behaviour of Split** attribute, select **Pull**.
4. Click **Ok** to confirm your selection.

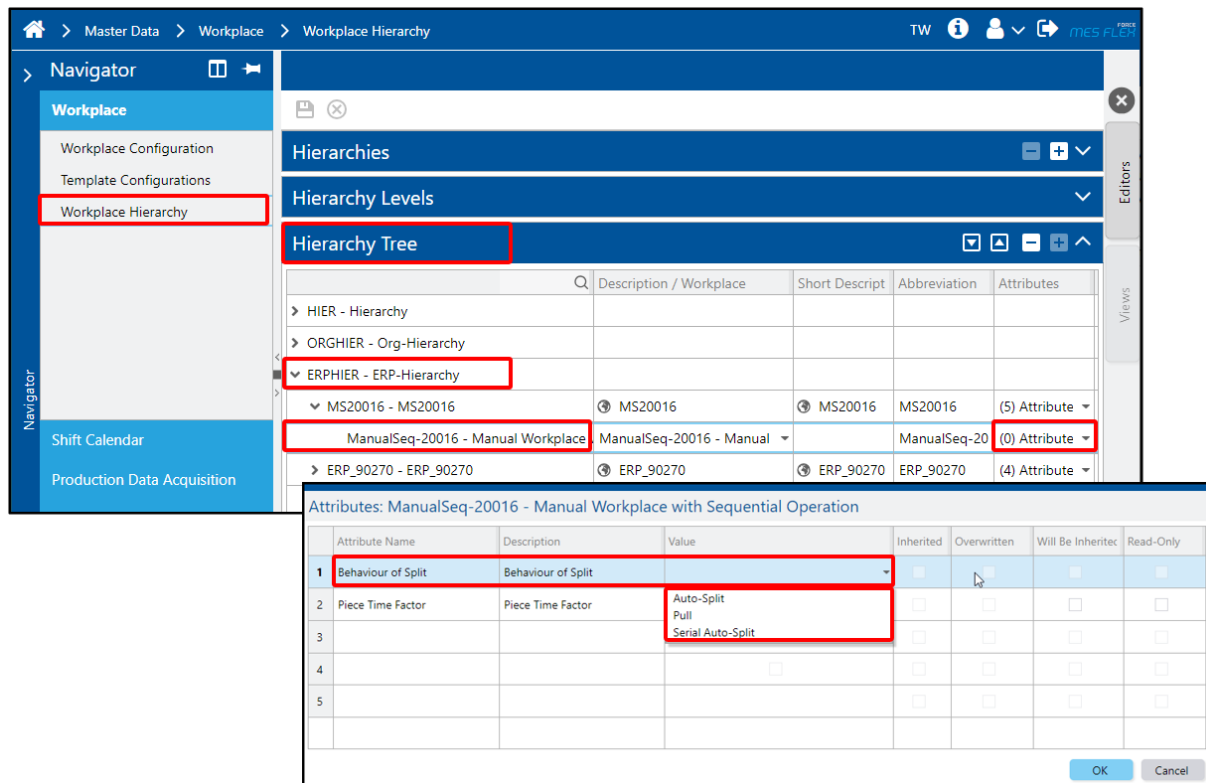


Fig. 13: Choosing the procedure for assigning operations

8.6.1 Piece time factor

i The **Piece Time Factor** can only be used in a combination with the **Auto-Split** procedure.

It is always used when the piece time at the workplace is not specified as time per piece but as time per s piece. You can then use the piece time factor s to compensate and calculate the time per unit produced (TIME_PER_UNIT) correctly:

Time per unit = Piece time * Unit / Piece time factor

In a workplace group composed of machines of different technologies it may happen that these have different processing speeds. The piece time of a slower machine can be readjusted in this case using a piece time factor s < 1 and the one of a faster machine using a piece time factor s > 1 (for example, you may use 2.0 for a machine producing at double speed or 0.75 for a machine which is 25% slower).

i The default value of the piece time factor is 1 and – just as a value of 0 – does not make a change.

Path: Master Data > Workplace > Workplace Hierarchy > Hierarchy Tree > ERP – ERP

To set the piece time factor:

1. Select the corresponding node at the lowest level of the hierarchy tree.
2. Open the drop-down menu in the **Attributes** column.
- ➔ A dialog window opens.
3. For the **Piece Time Factor** attribute, enter the desired **Value**.
4. Click **Ok** to confirm your selection.

Attributes: ManualSeq-20016 - Manual Workplace with Sequential Operation

	Attribute Name	Description	Value	Inherited	Overwritten	Will Be Inherited	Read-Only
1	Behaviour of Split	Behaviour of Split		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Piece Time Factor	Piece Time Factor	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

OK Cancel

Fig. 14: Setting the piece time factor

8.7 Workplace-specific configuration

8.7.1 Activating ERP upload

Path: Master Data > Workplace Configuration > Workplace Configuration

The ERP upload data service can also be activated or deactivated for each individual workplace using the **ERP Upload Active** setting.

- Temporary deactivation is usually used when creating a new workplace. Its ERP upload is not activated until the new workplace is effectively enabled. This helps to avoid double bookings in the transition phase.

Master Data

Number of Workplaces: 46/200

Workplace Name	Booking Active	ERP Upload Active	Assigned Clampings
100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
110	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
120	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
130	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
200	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
210	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
220	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Fig. 15: ERP upload activation for individual workplaces

8.7.2 ERP cycle time

Path: Master Data > Workplace Configuration > Template Configurations > Template Selection Tree > Workplace > Booking Logic

You can set the time interval for ERP uploads relating to quantities in the Workbench. Increasing the time interval may be useful, for example, if the ongoing data transmissions cause a high system load.

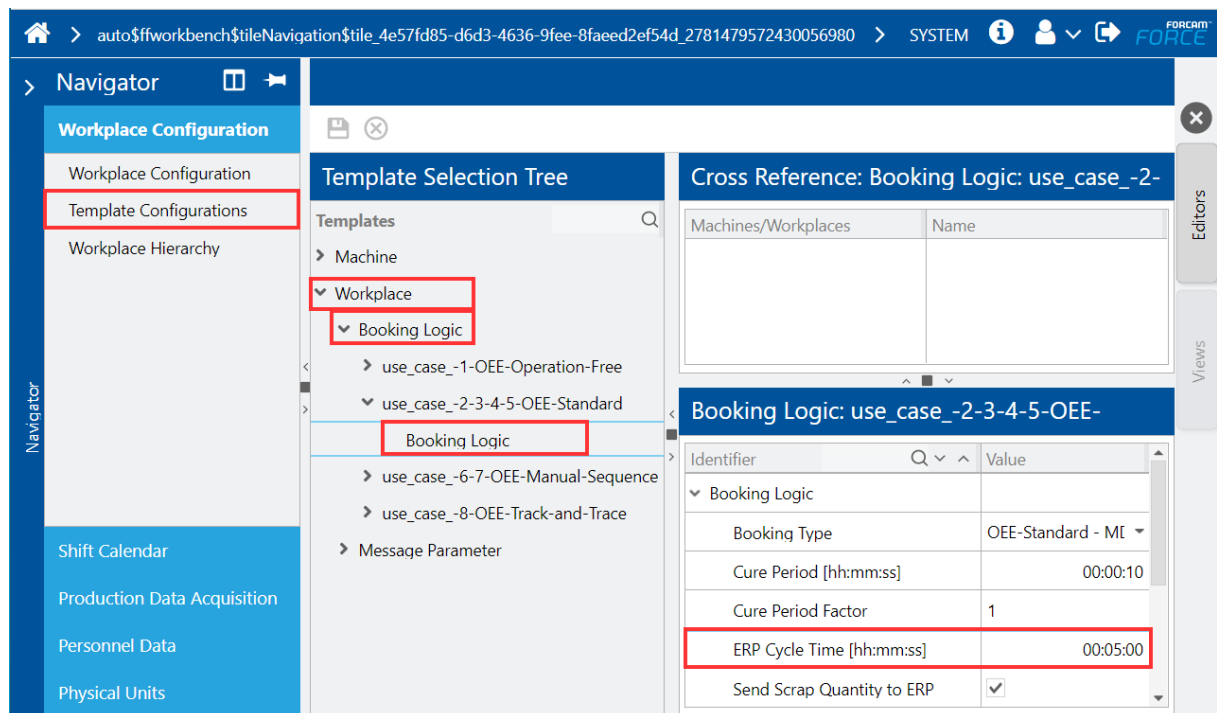


Fig. 16: Setting the ERP cycle time

8.7.3 Send scrap quantity to ERP

Path: Master Data > Workplace Configuration > Template Configurations > Template Selection Tree > Workplace > Booking Logic

In the case of a scrap quantity booking, the quantity can be reported to the ERP system together with the scrap reason. This can also be activated or deactivated in the Workbench.

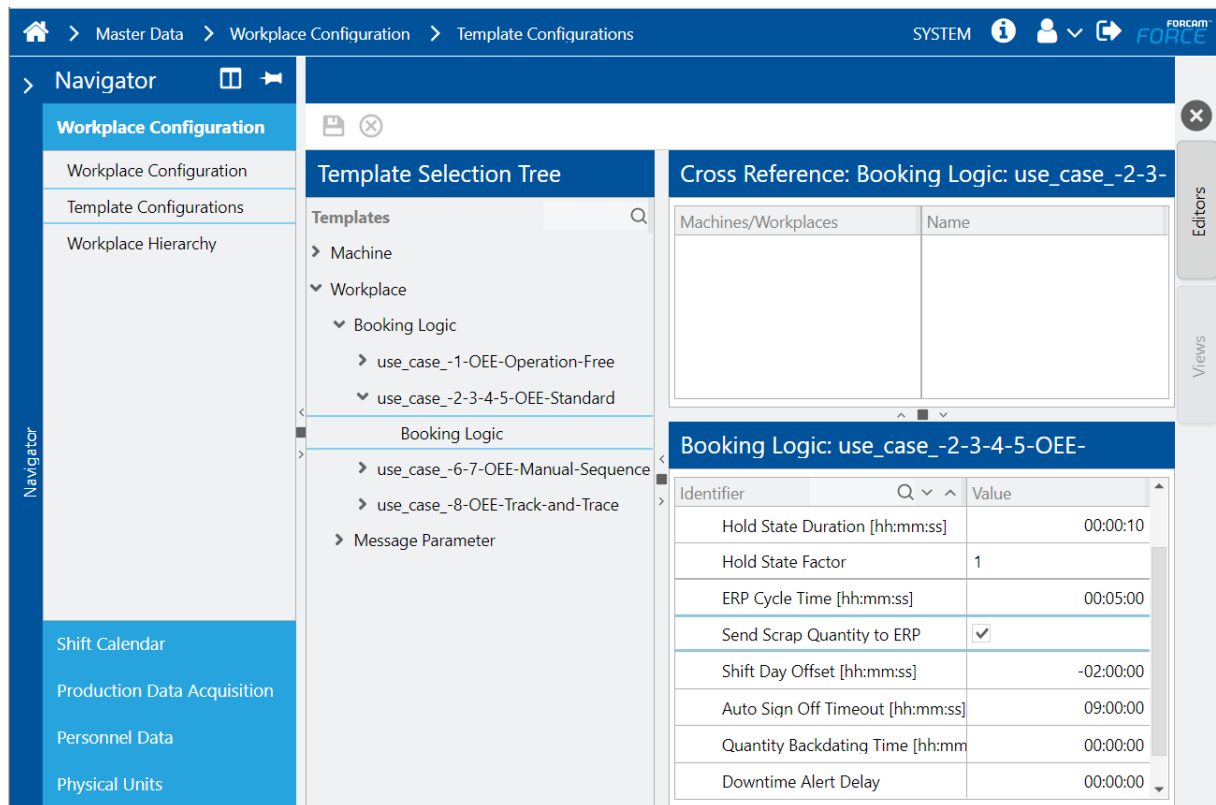


Fig. 17: Send scrap quantity to ERP

The scrap reason must match the scrap reasons defined in the ERP system. ERP scrap reasons are defined in the Workbench master data.

Path: Master Data > Production Data Acquisition > Operation Quality Type

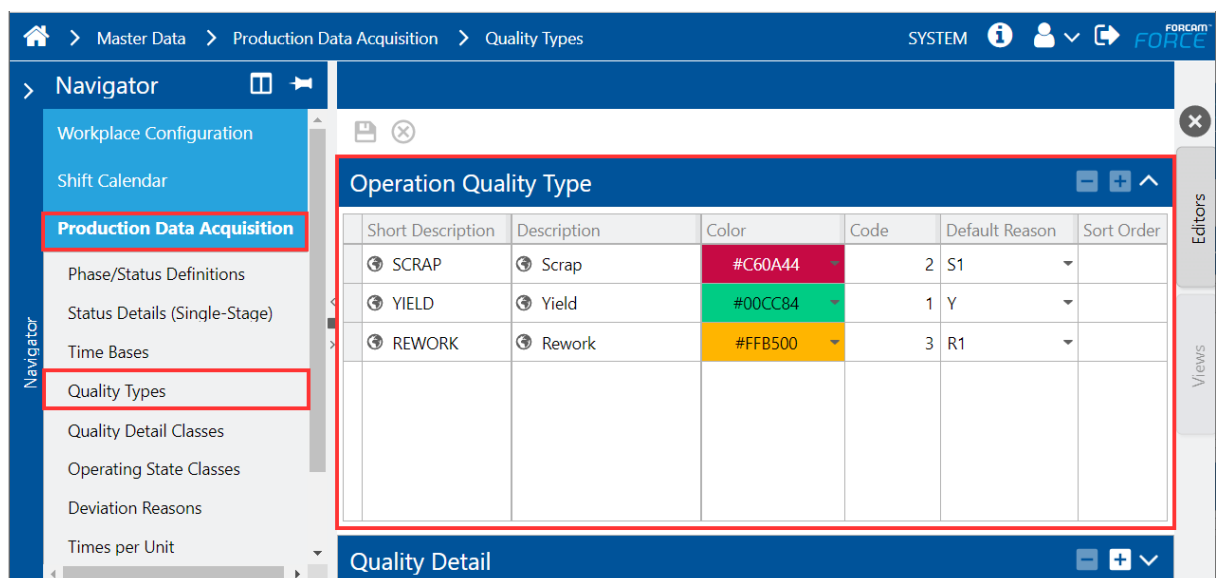


Fig. 18: Managing scrap reasons

9 Data structures

9.1 XSL basic data types

The transformation of the XML data transmitted is controlled by XSDs (XML Schema Definitions) permanently included in MES FLEX. XSDs contain information about the data structures used. Data elements and attributes are specified on the basis of standard data types defined in XSL. This includes the following data types:

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

Table 19: XSL basic data types used

9.2 XSL complex data types

XSL basic data types can be used to form XSL complex data types. This is done by joining several data elements and/or attributes in a hierarchical structure. These complex data types formed can in turn be utilized for defining application-specific data structures. Examples of application-specific data structures in MES FLEX are production orders, personnel master records or shifts.

An XSD for ERP keys (*ERPKeyType*) and a data type for orders built thereon (*OrderType*) may look as follows:

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

Fig. 19: Sample XSD for the *ERPKeyType* and *OrderType* data types

XML datasets of an order (*OrderType* data type) generated in this way may look as follows:

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

Fig. 20: Example of a XML dataset for a production order

Hence, an XSD provides the information for the structure and arrangement of the elements and/or attributes of a dataset in XML format.

This comprises the following information:

- Number and order of elements and child elements
- Data types of elements
- Preset and constant values of elements


The following chapters describe the data structures and XSDs used in MES FLEX more specifically. A distinction is made between the download and upload data services.


10 Download

The download data service supplies MES FLEX with the following datasets relevant for production:

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

These datasets are transmitted as a sequence of attributes, the structure of which is precisely defined by related XSDs.

 Mandatory datasets are the order header data and the associated operations. All other datasets are only needed for display on the MES FLEX user interface or for functional extensions such as tool data management (TDM), document management and container tracing.

 If a data processing error occurs, the data are rolled back, i.e. message processing is discarded.

Some attributes are mandatory. This depends on whether the dataset action is about a new entry or a change or a deletion. This is evident from the following tables (where relevant) in the right-hand columns (mandatory field for action).

10.1 Download basic data types

In addition to the data types existing by default in XSL, MES FLEX also uses special data types that are also used for the specification of the download datasets.

10.1.1 Time fields

In addition to the *xs:dateTime* data type already existing in XLS (see Table 19: XSL basic data types used Table 19), time data are also transmitted in the form of a sequence of time fields. For this purpose, MES FLEX provides an XSD for the complex *ct:timeUnitType* data type.

The definition of a time field always comprises the definition of a time value and an associated time unit.

Based on this unit, the value of the time field transmitted is converted to an internal MES FLEX standardized time base value. Correct conversion and interpretation of the time value is only possible with a valid unit.

The following time units are supported for the ERP download:

Time unit	Description
MS	Millisecond
S	Second
MIN	Minute
STD	Hour
H	Hour
M90	Minute / 90
TMU	Technical minute

Table 20: Supported time units for time fields

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

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

Fig. 21: Example of a time specification in the form of time fields

10.1.2 Multilingual texts

Texts can generally be transmitted as multilingual texts. MES FLEX currently supports the following languages on the graphic user interface:

- German
- American English
- British English
- French
- Chinese

The following complex data types are provided for this purpose in MES FLEX XSDs:

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

Table 21: Data types for multilingual texts

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

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

Fig. 22: Example of a multilingual text

If a text translation should be missing, the text of the default language set in the system will be used. If this text should also be missing, the language codes are sorted in descending order and the translation of the first language code available will be used.

Languages configured	Default	Translations transmitted	Translations adopted
de-DE	x	n/a	British English
en-GB		British English	British English
en-US		American English	American English
fr-FR		n/a	British English

Table 22: Example of text substitution if a translation in the default language is not available

10.1.3 Dataset action

The purpose of the dataset transmitted, i.e. whether it should be used for a new entry, a change or deletion of the corresponding dataset in MES FLEX, is set by an attribute of the complex *ct:actionType* data type.

Action	Description
ADD	Add/change dataset
DELETE	Delete dataset
NOP	Do not adopt dataset. This does not apply to child elements. In this way, subordinate attributes can specifically be transmitted for processing. Superordinate attributes are not adopted.

Table 23: Available actions in the data transmission process

10.1.4 ERP key

An ERP key is an individual key for assigning orders, machines, personnel or shifts to a client, company code or building in the ERP system. ERP keys can be configured in the Workbench so that

they match with the ERP system. ERP keys are transmitted with the complex *ct:ERPKeyType* data type with the following attributes:

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
Client	Client	xs:string(32)	x	x
CompanyCode	Company code	xs:string(32)	x	x
Plant	Plant	xs:string(32)	x	x
SystemId	System ID	xs:string(32)	x	

Table 24: Attributes in an ERP key (see also ERPDownloadOrder.XSD)

The company code can be used as an accounting unit of the company in order to distinguish business processing.

The system ID is used for unique identification of the ERP key in the system. This is important if the client, company code and plant attributes are not unique. Hence, the system ID can be used, for example, to distinguish connections to the test or productive system.

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

10.1.5 Material

For the definition and transmission of a material, the complex *ct:MaterialType* data type is available with the following attributes:

Attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
Description	Description	ct:MultilanguageTextType	x	
Number	Material number	xs:string(255)	x	
Type	Material type	xs:string(255)		

Table 25: Attributes in the material dataset

The material number, also referred to as part number, is primarily used for identifying the material. The material type specification is optional.

‘Material’ is a general term in MES FLEX referring to parts processed in production. A material is processed at a workplace (e.g. machined or installed) and the process may (not necessarily) produce a new material. The materials input into this process at the workplace are referred to as components

if a different or new material is produced in the process. If a material is only machined at a workplace, e.g. milled or formed, it is still referred to as a material.

Hence, there are materials composed of several components. The material to be produced requires a minimum number of such components to be installed correctly and result in the desired product. The bill of material is produced therefrom and can be transmitted with operations.

For example: The desired output material is a cylinder engine. Simplified it consists of 4 cylinders, 4 screws and 1 crankshaft. These parts then form the components of the cylinder engine material. They also appear in the form of a bill of materials.

When a material is transmitted to MES FLEX for the first time, it is saved in a MES FLEX-internal list of materials and from then on available for a search (e.g. in the performance analysis of the office client). For this reason, it is useful to transmit not only the material number but also a material type and a meaningful description. It is also possible to use the product characteristics as described in chapter 0 for a more detailed specification.

10.1.6 User data fields

The complex *ct:UserData* data type is available for the transmission of user-specific data and their display on the user interface of MES FLEX along with the following freely usable attributes:

Attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
UF1	User field 1	xs:string(255)		
UF2	User field 2	xs:string(255)		
...				
UF50	User field 50	xs:string(255)		

Table 26: Attributes in the user data fields dataset


10.2 Order data

An XSD for orders is used for order supply from the ERP system.

This XSD controls the transmission of the order header data and the following subordinate objects:

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

Separate XML datasets are created for the order header data and the subordinate objects.

-  Components are only required if the components to be installed are displayed in MES FLEX or need to be confirmed to the ERP system.

Production resources and tools are only required if they are needed for display in MES FLEX or if tool data management (TDM) is used.

10.2.1 Order header

The essential elements in the order header dataset are:

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

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
action	Dataset action	ct:actionType	x	x
ERPKey	ERP key	ct:ERPKeyType	x	x
OrderNumber	Order number	xs:string(32)	x	x
OrderSplit	Order split	xs:string(8)	x	x
ERPStatusIds	Status IDs from ERP system (with blanks as delimiters)	xs:string(255)		x
ERPStatusCodes	Associated status codes from ERP system (with blanks as delimiters)	xs:string(255)		
ERPStatusCodesTranslation	Translation for status codes (with blanks as delimiters)	xs:string(255)		
OrderPriority	Priority	xs:string(255)		

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
ReleaseDateTime	Release date & time	xs:dateTime		
TargetStartDateTime	Target start date & time	xs:dateTime	x	
TargetEndDateTime	Target end date & time	xs:dateTime	x	
TargetQuantity	Target quantity	xs:double	x	
DisplayQuantityUnit	Quantity unit	xs:string(255)	x	
ProductionVersion	Production version	xs:string(255)		
PlannerGroupInCharge	Planner group/department in charge	xs:string(255)		
Description	Description	ct:MultilanguageTextType		
ScheduledStartDate	Scheduled start date	xs:dateTime		
ScheduledEndDate	Scheduled end date	xs:dateTime		
MaterialCharacteristics	Sequence of product characteristics of the material produced	Sequence of ord:MaterialCharacteristicType		
UserDataFields	User data fields	ct:UserDataTypes		
Operations	Sequence of operations	Sequence of ord:OperationType	x	
Material	Material produced	ct:MaterialType	x	

Table 27: Attributes in the dataset order header

10.2.2 Material characteristics

For the definition and transmission of the 'material characteristics' dataset, the complex data type *ord:MaterialCharacteristicType* is available. It offers the following attributes:

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

Table 28: Attributes in the material characteristics dataset

10.2.3 Operations

The operations pertaining to the order are transmitted as a sequence of individual operations.

The essential elements of an operation are:

- *ERP key, operation number and long text* for the identification of the operation
- *Workplace or workplace group* for production
- *Operation split* for defining the assignment to a workplace of the workplace group (capacity group in the ERP hierarchy) in the Auto-Split procedure
- *Confirmation number* as an identifier for the ERP system to be able to assign a later confirmation to the order or operation again
- *Target quantity* along with the quantity unit as a quantity specification for the material produced
- *Planned start date & time and end date & time* as time specifications for the production of the material
- *Setup time and piece time* with time unit as another time specification for the production of the material
- *Piece time factor* for the calculation of a *piece time per quantity unit*
- *Bill of material* for defining the required components
- *Production resources and tools* for defining the required auxiliary equipment (e.g. tools)

The piece time factor can be used for adjusting the piece time per quantity unit to the machine technology:

Piece time per quantity unit = Piece time / piece time factor

For the definition and transmission of an operation, the complex *ord:OperationType* data type is available. It offers the following attributes:

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
action	Dataset action	ct:actionType	x	x
OrderERPKey	ERP key	ct:ERPKeyType	x	x
OperationNumber	Operation number	xs:string(32)	x	x
OperationSplit	Operation split <ul style="list-style-type: none"> — 0: Without operation split (pull procedure) — 1: Assignment to the 1st workplace of the workplace group / capacity group — 2: Assignment to the 2nd workplace of the workplace group — 3: etc. 	xs:string(8)	x	x

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
ConfirmationNumber	Confirmation number	xs:string(255)	x	x
ERPYieldQuantity	Reported yield quantity	xs:double		
ERPReworkQuantity	Reported rework quantity	xs:double		
ERPScrapQuantity	Reported scrap quantity	xs:double		
ERPPlannedScrapQuantity	Planned scrap quantity	xs:double		
ERPStatusIds	Status IDs in the ERP system (with blanks as delimiters)	xs:string(255)		x
ERPStatusCodes	Status codes in the ERP system (with blanks as delimiters)	xs:string(255)		
ERPStatusCodesTranslation	Translation for status codes	xs:string(255)		
Description	Description	ct:MultilanguageTextType		
OperationText	Long text	ct:MultilanguageTextType	x	
ControlKey	Control key	xs:string(255)		
OrderType	Order type	xs:string(255)		
FunctionType	Function of the operation <ul style="list-style-type: none"> 1: Normal 2: Training 3: Maintenance 	xs:long		
LeadingOperation	Leading operation within the operation block (yes/no) <ul style="list-style-type: none"> true false 	xs:boolean		
BlockNumber	Operation block	xs:string(255)		
BlockDescription	Description of operation block	ct:MultilanguageTextType		
OperationSequence	Order sequence	xs:string(255)		
AutoStartId	ID coming from the machine to start the operation automatically	xs:string		
TargetWorkplace	Workplace for production	xs:string(255)	x	

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
TargetWorkGroup	Workplace group ID of the workplace	xs:string		
WorkplaceGroup	Workplace group of the workplace	xs:string(255)		
TargetStartDateTime	Planned start date & time	xs:dateTime	x	
TargetEndDateTime	Planned end date & time	xs:dateTime	x	
DefaultStrokeFactor	Stroke factor (quantity produced per stroke or cycle)	xs:double		
DefaultPieceTimeFactor	Piece time factor (used for calculating the piece time per quantity unit)	xs:double	x	
DisplayQuantityUnit	Quantity unit	xs:string(255)	x	
TargetQuantity	Target quantity	xs:double	x	
StandardValue1	Target setup time	xs:string(255)	x	
StandardUnit1	Target setup time unit	ct:timeUnitType	x	
StandardValue2	Piece time	xs:string(255)	x	
StandardUnit2	Piece time unit	ct:timeUnitType	x	
StandardValue3	Default value 3	xs:string(255)		
StandardUnit3	Default value 3 unit	ct:timeUnitType		
StandardValue4	Default value 4	xs:string(255)		
StandardUnit4	Default value 4 unit	ct:timeUnitType		
StandardValue5	Default value 5	xs:string(255)		
StandardUnit5	Default value 5 unit	ct:timeUnitType		
StandardValue6	Default value 6	xs:string(255)		
StandardUnit6	Default value 6 unit	ct:timeUnitType		
AlternateOperation Number	Alternative operation number	xs:string(255)		

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
ProductionVersion	Production version	xs:string(255)		
CounterNumber	Counter number (press station counter)	xs:long		
UserStatus	User status in the ERP system	xs:string(255)		
OverdeliveryCheck	Checks for observance of the overdelivery tolerance	xs:boolean		
OverdeliveryQuantity	Overdelivery tolerance (percentage limiting the quantity booked above the target quantity)	xs:double		
UnderdeliveryCheck	Checks for observance of the underdelivery tolerance	xs:boolean		
UnderdeliveryQuantity	Underdelivery tolerance (percentage limiting the quantity booked below the target quantity)	xs:double		
DefaultTransportQuantity	Default quantity for material movements in Track & Trace	xs:double		
QualityCheckCycle	Quantity produced after which a quality check should be performed	xs:double		
StandardPackSize	Maximum quantity that can be produced into the output container (Track & Trace)	xs:double		
DefaultComponentConsumptionFactor	Predefined quantity of a component consumed per material unit produced (if this component is not included in the bill of material)	xs:double		
EarliestStartDate	Earliest start date & time	xs:dateTime		
LatestStartDate	Latest start date & time	xs:dateTime		
EarliestEndDate	Earliest end date & time	xs:dateTime		
LatestEndDate	Latest end date & time	xs:dateTime		
ScheduledStartDate	Scheduled start date & time	xs:dateTime		
ScheduledEndDate	Scheduled end date & time	xs:dateTime		
TargetQueueTime	Target Queue Time (default queue time in ms)	xs:long		

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
TargetQueueTimeUnit	Target queue time unit	ct:timeUnitType		
TargetProcessingTime	Target processing time (default processing time in ms if the processing time does not depend on the piece count) Example: Baking 1 bread roll takes 15 minutes. If you bake 3 rolls simultaneously, it does not take longer.	xs:long		
TargetProcessingTimeUnit	Target processing time unit	ct:timeUnitType		
TargetTeardownTime	Target teardown time (default teardown time in ms)	xs:long		
TargetTeardownTimeUnit	Target teardown time unit	ct:timeUnitType		
TargetWaitTime	Target wait time (default wait time)	xs:long		
TargetWaitTimeUnit	Target wait time unit	ct:timeUnitType		
TargetMoveTime	Target move time	xs:long		
TargetMoveTimeUnit	Target move time unit	ct:timeUnitType		
MinimumSendAheadQuantity	Minimum send ahead quantity (after producing this amount, the subsequent operation may start already)	xs:double		
MinimumOverlapTime	Minimum overlap time (for intended overlapping) The subsequent operation should start before the end of the preceding one. The overlap must have at least this predefined duration.	xs:long		
MinimumOverlapTimeUnit	Minimum overlap time unit	ct:timeUnitType		
UserFields	User data fields	ct:UserDataTypes		
OperationComponents	Components used, or bill of material (see chapter 10.2.4)	Sequence of ord:OperationComponentType		

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
ProductionResourceTools	Production resources and tools used (see chapter Fehler! Verweisquelle konnte nicht gefunden werden.)	Sequence of ord:ProductionResourceToolType		
Material	Material produced	ct:MaterialType	x	
CustomerEnhancements	User fields	Sequence of ord:PartialOrderQuantityDataSection (*)		

Table 29: Attributes in the operations dataset

(*) Attributes in the *ord:PartialOrderQuantityDataSection* data type:

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
PartialOrderQuantityDataSection	User fields	Sequence of ord:PartialOrderQuantityDataType (**)		

Table 30: Attributes in the *ord:PartialOrderQuantityDataSection* data type(**) Attributes in the *ord:PartialOrderQuantityDataType* data type:

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
UF1	User field	xs:string(255)		
...				
UTF50	User field	xs:string(255)		

Table 31: Attributes in the *ord:PartialOrderQuantityDataType* data type

10.2.4 Components

The components required for an operation are transmitted as a sequence of individual components and thereby form the bill of material.

A component can be identified by its *component number* and *position number*.
The position number is used for reserving or specifying dependent requirements.

For the definition and transmission of a component, the complex *ord:OperationComponentType* data type is available. It offers the following attributes:

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
action	Dataset action	ct:actionType	x	x
ComponentNumber	Component number	xs:string(255)	x	x
PositionNumber	Position number of the reservation / dependent requirements	xs:string(255)	x	x
BaseUnitOfMeasure	Base unit of measure	xs:string(255)		
BatchNumber	Batch number	xs:string(255)		
ConfirmationNumber	Confirmation number of the operation	xs:string(255)		
Description	Description	ct:MultilanguageTextType		
MaterialGroup	Material group	xs:string(255)		
MaterialType	Material type	xs:string(255)		
RequirementDate	Requirement date	xs:dateTime		
RequirementQuantity	Requirement quantity	xs:double		
ReservationNumber	Number of the reservation or dependent requirements	xs:string(255)		
StorageBin	Storage bin	xs:string(255)		
StorageLocation	Storage location	xs:string(255)		
TraceNeeded	Specifies whether the SFT user needs to enter the batch number to pick this component	xs:boolean		
UserFields	User data fields	ct:UserDataTypes		

Table 32: Attributes in the component dataset

10.2.5 Production resources and tools

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

An individual production resource or tool is identified by a number, type identifier and optionally a group.

For the definition and transmission of an individual production resource or tool, the complex *ord:ProductionResourceToolType* data type is available. It offers the following attributes:

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
Number	Number	xs:string(255)	x	x
Type	Type identifier	xs:string(255)	x	x
Sequence	Sequential number	xs:string(255)		
Quantity	Quantity	xs:double		
Unit	Quantity unit	xs:string(255)		
Group	Group	xs:string(255)		
UserFields	User data fields	ct:UserDataTypes		
Description	Description	ct:MultilanguageTextType		

Table 33: Attributes in the production resources and tools dataset

10.3 Shift data

Shift data are always transmitted as a complete update. This means that the data of all shifts are transmitted together in a dataset. All existing shifts except the current one are deleted. The current one is not changed.

Shifts can also be edited completely in MES FLEX.

Every attribute that can be transmitted has a corresponding entry in the database used in MES FLEX. Some attributes are mandatory, depending on whether the dataset action initiates a new entry, a change or a deletion. This is evident in the following tables in the right-hand columns (mandatory field for action).

The shifts are combined and transmitted as a sequence of individual shift definitions.

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
action	Dataset action	ct:actionType	x	x
ShiftsType	Shifts	Sequence of shift:ShiftType	x	x

Table 34: Attributes in the shifts dataset

The essential elements of a shift definition are:

- the *ERP key* for identification
- the *workplace* to which the shift definition applies
- the *shift type* to define free, late, night, or manual shift
- the *shift day* to set the weekday
- the *start date & time* and *end date & time* to define the time interval (hours) of the shift

In contrast with the other shift types defined, manual shifts can be created using a special dialog in the Shopfloor Terminal (SFT).

This shift type can be used, for example, by workers to assign short-term unplanned activities or operations to a shift on the spot. This eliminates the need to correct a shift later.

For the definition of an individual shift, the complex *shift:ShiftType* data type is used with the following attributes:

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

Table 35: Attributes in the complex *shift:ShiftType* data type


The complex *shift:ShiftBreakType* can be used to define one or more additional time intervals for breaks within a shift.

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
StartDateTime	Start date & time	xs:dateTime	x	x
EndDateTime	End date & time	xs:dateTime	x	x

Table 36: Attributes in the complex *shift:ShiftBreakType* data type

10.4 Personnel data

Personnel data are always transmitted as a complete update. This means that the data of the complete personnel involved in production are transmitted. Hence, employees who are no longer included in the complete update are deleted.

-  The personnel master data can also be updated completely in MES FLEX. It is also possible to define different or abstract person names or roles there (e.g. "User01"). You can prevent it from being overwritten in a download by setting an associated attribute '**Do not delete in update**' in MES FLEX.

The personnel data are combined and transmitted as a sequence of all employees included in the personnel master data:

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
action	Dataset action	ct:actionType	x	x
PersonsType	Personnel type	Sequence of pers:PersonType	x	x

Table 37: Attributes in the personnel data dataset

The essential elements of a personnel dataset are:

- ERP key, personnel number, identity card number and name for the identification of the employee
- Workplace to which the shift definition applies
- Shift type to define free, late, night, or manual shift
- Shift day to set the weekday

For the definition of an employee, the complex *pers:PersonType* data type is used with the following attributes:

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
PersonERPKey	ERP key	pers:PersonERPKeyType		
PersonnelNumber	Personnel number	xs:string(255)	x	x
IdentificationNumber	Identity card number	xs:string(255)	x	x
FirstName	First name	xs:string(255)	x	x
LastName	Last name	xs:string(255)	x	x
CostCenter	Cost center	xs:string(255)		

Table 38: Attributes in the complex *pers:PersonType* data type

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
Client	Client	xs:string(32)	x	x
CompanyCode	Company code	xs:string(32)	x	x
PersonnelArea	Personnel area	xs:string(255)	x	x
SystemId	System ID	xs:string(32)		

Table 39: Attributes in the complex *pers:PersonERPKeyType* data type

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

10.5 Overhead cost data

MES FLEX includes a function to accumulate overhead costs incurred by order processing by means of overhead cost collectors and report this information back to the ERP system via the upload data service. However, reporting back is only possible so far for SAP ERP systems and in IDoc format. The definitions of the overhead cost collectors, or the overhead cost collector master data, can be transmitted as a complete update from the ERP system to MES FLEX.

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

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
Client	Client	xs:string	x	x
CompanyCode	Company code	xs:string	x	x
ControllingArea	Controlling area	xs:string	x	x
SystemId	System ID	xs:string	x	x

Table 40: Attributes in the complex *oc:OverheadCostERPKeyType* data type

The dataset for the definition of the overhead cost collectors used is combined as a sequence of individual overhead cost collector definitions and transmitted from the ERP system to MES FLEX.

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
action	Dataset action	ct:actionType	x	x
OverheadCost	Overhead costs	Sequence of <i>oc:OverheadCostsCollectorType</i>	x	x

Table 41: Attributes in the overhead cost dataset

For the definition of an individual overhead cost collector, the complex *oc:OverheadCostsCollectorType* data type is used with the following attributes:

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
OverheadCostsERPKey	ERP key	<i>oc:OverheadCostERPKeyType</i>	x	x
Plant	Plant	xs:string		
OverheadCostsCollectorType	Cost collector type: OVERHEAD_COST_CENTER (for overhead cost collector) INTERNAL_ORDER (for internal orders)	xs:string	x	x

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
CostUnit	Cost center (for overhead cost collector)	xs:string		
OrderNumber	Order number	xs:string		
OrderType	Order type	xs:string		
Abbreviation	Abbreviation	ct:MultilanguageTextType		
Description	Description	ct:MultilanguageTextType		
ResponsiblePerson	Responsible person	xs:string		
Color	Color code for visualization (RGB value)	xs:string		
ValidFrom	Start date & time of validity	xs:dateTime		
ValidTo	End date & time of validity	xs:dateTime		
ExecutionCostUnitChangeable	Refers to the sender cost center (cost center to be credited). Setting for entry/change. Defines whether the sender cost center can be entered and changed on the SFT.	xs:boolean		
ReceiverCostUnitChangeable	Refers to the originator cost center (cost center to be debited). Used directly for internal performance accounting. Defines whether the originator cost center can be entered and changed on the SFT.	xs:boolean		
ERPAutomaticBooking	Booking type of the overhead costs reported in the ERP system: AUTO: Automatic assignment or booking of the costs in the ERP system. MANUAL: Costs are first written into a table. Assignment or booking of the costs in the ERP	xs:string		

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
	system is done manually later.			
ERPReportingEnabled	Activate reporting of overhead costs incurred	xs:boolean		
UserStatus	User status	xs:string		

Table 42: Attributes in the complex *oc:OverheadCostsCollectorType* data type

10.6 Warehouse data from the warehouse management system

ERP systems with an existing warehouse management system can transmit data on the stock levels of materials in addition.

The complete warehouse data are then combined and transmitted as a sequence of stock levels of individual materials:

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
StocksType	Warehouse data	Sequence of sto:StockType	x	x

Table 43: Attributes in the warehouse data dataset

For the definition of a stock level of a specific material, the complex *sto:StockType* data type is used with the following attributes:

XML attribute			Mandatory field for action	
Name	Description	Data type (field length)	Add/change	Delete
Material	Material	ct:MaterialType	x	x
Quantity	Quantity	xs:double	x	x
QuantityUnit	Quantity unit	xs:string(3)	x	x
StockLocation	Storage location	xs:string(10)	x	x

Table 44: Attributes in the complex *sto:StockType* data type

10.7 Response to quantity requests

In addition to the datasets mentioned, there are also datasets with quantity specifications that serve as responses to requests from MES FLEX to the ERP system regarding the following information:

- Requests relating to material units
- Requests relating to packaging units
- Requests relating to remaining quantities in the warehouse management system (after completion of operations)

11 Upload

The upload data service is used for transmitting events or sequences of events and the associated datasets from MES FLEX to the ERP system in the form of upload messages.

Within MES FLEX, trigger events such as, for example, a change of the operation phase (e.g. *Setup*, *Processing* or *Interrupted*) or quantity bookings generate a message in the event message stream, complemented with the related event data. Adjustments can be made in the Rule Engine to complement events with custom calculation data (parameters).

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

- Change of operation phase
- Messages on operation duration or work time
 - Only generated in MES FLEX when an operation is completed or at shift end (inter-shift operation)
 - Per capacity type, i.e. machine production time, machine setup time
 - For personnel time, per person and activity type
- Quantity bookings
 - Asynchronous or individual & | | controlled by intervals
 - One separate message per quantity type
 - Yield quantity
 - Scrap quantity
 - Rework quantity
- Corrections
 - Delete / insert
 - Restrictions must be specified in detail and be supported by the ERP system.
 - Only possible with a unique message number
- Key figures
 - Must be defined according to the ERP interface characteristics
 - E.g. actual piece time calculation → Rule Engine application
- Ad-hoc maintenance via the Shopfloor Terminal (SFT)
 - Request ad-hoc maintenance
 - Start ad-hoc maintenance
 - Finish ad-hoc maintenance

The data structures of these events are created as DTOs (Data Transfer Objects) and built according to the object-based inheritance principle. Each event receives a generic standard content of the basic data structure and appropriate further inheritance items as well as its own content specific to the event.

The DTOs are then converted by MES FLEX to transferable XML documents.

11.1 Uploading basic data types

Two basic data types are available for the messages to be transmitted.

11.1.1 ERP key

For the identification of ERP resources such as workplaces, operations or personnel, the *ERPContext* dataset of *ERPContextType* data type is available.

It offers the following attributes:

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

Table 45: Attributes in the ERP key dataset

11.1.2 Messages to ERP system

For messages to the ERP system, generally the complex *ERPUploadMessage* data type is used. It offers the following attributes:

ERPUploadMessage			
Attribute	Description	Data type (field length)	Mandatory field
ConfirmationNumber	Confirmation number	DefaultStringType	x
MessageIdentifier	Unique message ID	xs:string(36)	x
MessageType	Message type: — REGULAR — CORRECTION	xs:string	x
MessageTimestamp	Timestamp	xs:dateTime	x
ActionTimestamp	Timestamp (for corrections, the time of correction)	xs:dateTime	x
CustomMessageData	Sequence of custom data (concatenated pairs from key with value)	Sequence of xs:string	

ERPUploadMessage			
Attribute	Description	Data type (field length)	Mandatory field
OperationMessage	Message regarding operations	OperationMessageType (see chapter 11.2)	
OperationSchedule	Message regarding operation scheduling	OperationScheduleType (see chapter 11.3)	
WarehouseMessage	Message regarding warehouse management system	WarehouseMessageType (see chapter 11.4)	

Table 46: Attributes in the *Message to ERP system* dataset

Depending on the message type, one of the data structures *OperationMessage*, *OperationSchedule* or *WarehouseMessage* is used.

The following sections describe the use of these data structures in more detail.

11.2 Messages on operations

This section describes the messages relating to operations. They can alternatively also be sent as callbacks via the FORCE Bridge API (see chapter 5 “Callbacks and events”). For more information on ERP callbacks, refer to the on-line documentation in the Swagger UI. For messages on operation events, the complex *OperationMessageType* data type is available. It offers the following attributes:

OperationMessageType				
Attribute	Sub-attributes	Description	Data type (field length)	Mandatory field
WorkplaceIdentification	UUID	Globally unique ID	xs:string(36)	x
	ERPContext	ERP key	ERPContextType	x
	ERPWorkplaceNumber	Workplace number in the ERP system	xs:string(255)	x
	WorkplaceNumber	Workplace	xs:string(255)	x
OperationIdentification	UUID	Globally unique ID	xs:string(36)	x
	ERPContext	ERP key	ERPContextType	x
	OrderNumber	Order number	xs:string(255)	x
	OperationNumber	Operation number	xs:string(255)	x
	OrderSplit	Order split	xs:string(255)	x
	OperationSplit	Operation split	xs:string(255)	x
	OrderProductionVersion	Production version	xs:string(255)	
PersonnelIdentification	OperationSequence	Operation sequence in the order	xs:string(255)	
	UUID	Globally unique ID	xs:string(36)	x
	ERPContext	ERP key	ERPContextType	x
	PersonnelNumber	Personnel number	xs:string(255)	x ³
	UUID	Globally unique ID	xs:string(36)	x
	ShiftDay	Date of shift event (in UTC)	xs:date	x

³ Not a mandatory field in phase change to *Setup*, *Processing* or *Interrupted*

OperationMessageType				
Attribute	Sub-attributes	Description	Data type (field length)	Mandatory field
ShiftIdentification	ShiftCode	Shift code	xs:string(255)	x

Table 47: Attributes in the *Operation event* dataset

Other attributes of the *OperationMessageType* data type are available depending on the type of operation message.

11.2.1 Change of operation phase

For messages on operation events occurring as a change of the operation phase, the following additional attributes are available in the *OperationMessageType* data type:

OperationMessageType				
Attribute	Sub-attributes	Description	Data type (field length)	Mandatory field
OperationPhase		Change of operation phase to: <ul style="list-style-type: none"> — SETUP — PROCESSING — INTERRUPTED — COMPLETED — CLOSED 	xs:string	x

Table 48: Attributes in the *Change of operation phase* dataset

11.2.2 Operation quantity bookings and reasons

For messages on operation information in the form of quantity bookings and reasons, the *OperationMessageType* data type offers the following additional attributes:

OperationMessageType				
Attribute	Sub-attributes	Description	Data type (field length)	Mandatory field
OperationQuantity	MaterialNumber	Number of the material produced in the operation according to the ERP system	xs:string(255)	x
	RelativeQuantity	Applied reasons as a sequence of: <ul style="list-style-type: none"> — Quality detail — Quality detail according to ERP (optional) 	Sequence of <ul style="list-style-type: none"> — xs:string(255) — xs:string(255) 	x

OperationMessageType				
Attribute	Sub-attributes	Description	Data type (field length)	Mandatory field
	AbsoluteQuantity	<p>Applied quantity booking as a sequence of:</p> <ul style="list-style-type: none"> — Quality <ul style="list-style-type: none"> ○ YIELD ○ SCRAP (optional) ○ REWORK (optional) — Quantity — Quantity unit <p>Each quality type is allowed only once in this message.</p>	<p>Sequence of</p> <ul style="list-style-type: none"> — xs:string(255) — xs:double — xs:string(255) 	x

Table 49: Attributes in the *Quantity bookings and reasons* dataset

11.2.3 Operation duration

For messages on operation duration, the *OperationMessageType* data type offers the following additional attributes:

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

Table 50: Attributes in the *Operation duration* dataset

11.2.4 Component quantities

For messages on operation information in the form of quantity bookings for components, the *OperationMessageType* data type offers the following additional attributes:

OperationMessageType				
Attribute	Sub-attributes	Description	Data type (field length)	Mandatory field
	ComponentNumber	Component number	xs:string(255)	x
	RelativeQuantity	Applied reasons as a sequence of:	Sequence of	x

OperationMessageType				
Attribute	Sub-attributes	Description	Data type (field length)	Mandatory field
OperationComponent		<ul style="list-style-type: none"> Quality detail Quality detail according to ERP (optional) 	<ul style="list-style-type: none"> xs:string(255) xs:string(255) 	
	AbsoluteQuantity	<p>Applied quantity booking as a sequence of:</p> <ul style="list-style-type: none"> Quality <ul style="list-style-type: none"> YIELD SCRAP (optional) REWORK (optional) Quantity Quantity unit <p>Each quality type is allowed only once in this message.</p>	<p>Sequence of</p> <ul style="list-style-type: none"> xs:string(255) xs:double xs:string(255) 	x

Table 51: Attributes in the *Component quantities* dataset

11.2.5 Assembly of components

For messages on operation events relating to assembly of components, the *OperationMessageType* data type offers the following additional attributes:

OperationMessageType				
Attribute	Sub-attributes	Description	Data type (field length)	Mandatory field
OperationAsBuilt	ComponentNumber	Component number	xs:string(255)	x
	ComponentPosition	Position number	xs:string(255)	x
	SerialNumberOutput	Serial number of the material produced	xs:string(255)	x
	ContainerNumberOutput	Output container	xs:string(255)	
	AssemblyType	<p>Assembly type:</p> <ul style="list-style-type: none"> SERIAL BATCH 	xs:string	x
	SerializationType	<p>Serialization type</p> <ul style="list-style-type: none"> INPUT (assemble) DELETE (disassemble) UPDATE (exchange) 	xs:string	x

Table 52: Attributes in the *Assembly of components* dataset

11.3 Operation scheduling

For messages regarding a modified scheduling (rescheduling) of an operation, the following additional attributes are available:

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

Table 53: Attributes in the *Operation scheduling* dataset

11.4 Overhead cost messages

MES FLEX includes a function to accumulate overhead costs incurred by order processing by means of overhead cost collectors and report this information back to the ERP system via the upload data service.

However, reporting back is only possible so far for SAP ERP systems via XSLT in IDoc format. For more information, refer to the manual on **ERP Interface – Data & Events via SAP IDoc**.

11.5 Warehouse data to the warehouse management system

The upload data service can be used for reporting stock levels changing by the ongoing production to the warehouse management system of the ERP system. Stock levels are used for Track & Trace applications (e.g. for container quantities).

For this type of messages on changed stock levels of specific materials, the complex *WarehouseMessageType* data type is available. It offers the following attributes:

WarehouseMessageType				
Attribute	Sub-attributes	Description	Data type (field length)	Mandatory field
OrderIdentification	UUID	Globally unique ID	xs:string(36)	x
	ERPContext	ERP key	ERPContextType	x
	OrderNumber	Order number	xs:string(255)	x
	OrderSplit	Order split	xs:string(255)	x
	OrderProduction Version	Production version	xs:string(255)	
OperationIdentification	UUID	Globally unique ID	xs:string(36)	x
	ERPContext	ERP key	ERPContextType	x
	OrderNumber	Order number	xs:string(255)	x
	OrderSplit	Order split	xs:string(255)	x
	OperationSplit	Operation split	xs:string(255)	x
	OrderProduction Version	Production version	xs:string(255)	
	OperationSequence	Operation sequence in the order	xs:string(255)	
WarehouseMessageType		Warehouse storage type to which the message refers: <ul style="list-style-type: none"> — SERIAL (reports the stock level for a serialized single piece) — PRODUCTION_CONTAINER (reports the stock level of a production container unknown to the warehouse management system) — INVENTORY_CONTAINER (reports the stock level of a known storage container) 	xs:string	
WarehouseEntity	ReferenceValue	Reference value (tracking number of a container or leading serial number of a single piece)	xs:string(255)	
	MaterialNumber	Material number	xs:string(255)	
	AbsoluteQuantity	Sequence of <ul style="list-style-type: none"> — Quality <ul style="list-style-type: none"> ○ YIELD ○ SCRAP (optional) ○ REWORK (optional) — Quantity (stock level) — Quantity unit Each quality type is allowed only once in this message.	Sequence of <ul style="list-style-type: none"> — xs:string(255) — xs:double — xs:string(255) 	

Table 54: Attributes in the *Warehouse data* dataset

It depends on the *WarehouseMessageType* whether *OrderIdentification* (for *PRODUCTION_CONTAINER*) or *OperationIdentification* (for *SERIAL*) is used.

11.6 Quantity requests

In addition to the datasets mentioned, there are also datasets for quantity requests to the ERP system regarding the following contents:

- Requests relating to material units
- Requests relating to packaging units
- Requests relating to remaining quantities in the warehouse management system (after completion of operations)

11.7 Serialization and correction capability

Unambiguous serialization of the data transmitted is necessary to ensure sequence orders and unique assignment of messages.

Correction capability can only be ensured if every single message is provided with a universally unique message number and the superordinate ERP system allows canceling as well (delete / insert messages). A correction process must be specified comprehensively for each ERP system.

12 Annex

12.1 Abbreviations and terms used

Abbreviation/Term	Description
OAuth	Open Authorization Authorization protocol for accessing external resources
API	Application programming interface
AVO (German only)	Operation
FORCE Bridge API	RESTful API for MES FLEX, which allows data to be both retrieved and written in a REST-compliant manner. The FORCE Bridge API MDC (Machine Data Collection) of MES FLEX can collect quantity-based piece count information from the machine to calculate the actually produced quantity.
CAQ	Computer-Aided Quality Assurance
DTO	Data Transfer Object
ERP	Enterprise Resource Planning Describes a software solution for resource planning in a company. An ERP system can be a data source.
HAL	Hypermedia Application Language
Infor ION	Infor Intelligent Open Network Middleware cloud platform that facilitates standardized data exchange in XML format between ERP systems and MES FLEX.
IoT / IIoT	Internet of Things / Industrial Internet of Things
JSON	Text-based data interchange format for transmitting structured data.
KPI	Key Performance Indicator
MDE	Machine Data Collection
RESTful	Representational State Transfer – a programming paradigm for distributed systems. The idea behind this is to offer server-side database entries as resources that can be created, retrieved, modified or deleted through a standardized set of commands.
RFC	Remote Function Call – standardized interface or procedure for communication with SAP systems.
SFT	Shopfloor Terminal
Swagger UI	A tool to visualize and interact with Open API resources
TDM	Tool Data Management
URI	Uniform Resource Identifier

	An identifier consists of a string used for identifying an abstract or physical resource. RFC 3986
UUID	Universally Unique Identifier. A standard for identifiers used in software engineering.
XML	Extensible Markup Language – a markup language to represent hierarchically structured data
XSL	Extensible Stylesheet Language – transformation languages for the definition of layouts for XML documents
XSD	XML Schema Definition – A standard for defining data structures for XML documents based on a large number of data types supported. The definition is made via an XSD file created in XML document format.
XSLT	XSL Transformation – a component of XSL. Used for converting XML documents among different systems or standards. MES FLEX uses XSLT for conversion between SAP IDoc documents and XML documents that can be processed by MES FLEX.