IFS Aviation Maintenance

External Supply Chain and Inventory Synchronization API Guide

ICAM 26.2



Document Revision: 26.2-1

Publication Date: November 02, 2026

Contents

1 Introduction.....5

2 Setting Up External Supply Chain API..... 9

Licensing and Subscription.....9

System Initialization.....9

Granting Security Permissions.....9

External Supply Chain API Authentication..... 10

Activate Alerts..... 10

Outbound Event Framework..... 11

Enabling the PubSub Framework..... 11

Subscribing for Outbound Events..... 11

Activating Outbound Events.....12

Mandatory Configuration Parameters Setup..... 20

Mandatory Job Setup.....21

3 Open API Specification..... 22

4 Part Demand.....23

Create/Update Part Request.....24

Publishing part requests raised in Maintenix.....24

Publishing part requests updated in Maintenix.....27

Update Part Request..... 30

Updating the notes on a Part Request in Maintenix..... 30

Estimated Arrival Date Update..... 31

Reserve Serialized Inventory on a Part Request..... 32

Reserve Batch on a Part Request.....33

Split Batch..... 35

Prevent Stealing Reserved Batch From Another Part Request.....47

Reduce Inventory Count on Unreserve..... 50

Cancel Part Request.....55

Inventory Item Issued..... 56

Notifying Maintenix that an inventory item was issued.....56

Inventory Turned In.....57

Inventory Turned In.....57

5 Inventory Synchronization & Availability 60

Create/Update Inventory 61

Create New Inventory 61

Update Inventory Location 62

Update Inventory Details 63

Update Inventory Condition 65

Update Inventory Ownership 67

Unarchive Existing Inventory 69

Update Serialized Inventory as KIT Content 70

Update Serialized Inventory as KIT Content: Error Case/Special Case 71

Inventory Condition Updated 71

Inventory Condition Change 71

Update Inventory Summary 73

Update Inventory Summary 73

6 Removals and Installs 77

Pre-conditions and Result 78

Component change - through labour pause 78

Component Removal - through labour row completion 80

Component Install - through 'Inventory Attach' function 83

Component Removal - through 'Inventory Detach' function 85

7 Part Synchronization 87

Create New Part Number 88

Approve Part Number 88

Obsolete Part Number 89

Part Definition Created 89

Part Definition Activated 89

Part Definition Obsolesced 90

Part Definition Updated 90

8 FAQs 92

9 Appendix 94

Pubsub-subscriptions.properties sample 94

Servers.properties sample..... 95
Example of Authentication (IAM/OIDC) within IFS Cloud.....96
Example of PubSub Configuration Within IFS Cloud..... 100

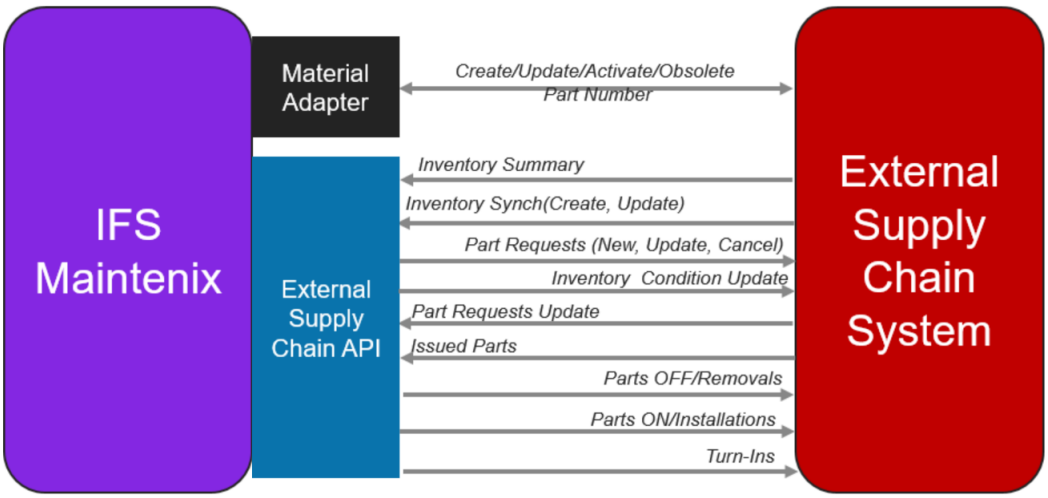
1

Introduction

The Integration of Maintenix with an External Supply Chain system is based on the following assumptions:

- Maintenix is the main system for part number control; however, our solution will still support the business scenario in which the external supply chain system initiates the part creation process.
- Maintenix remains the main system for configuration management, maintenance program management, line & base maintenance planning as well as line & base maintenance execution.
- The External Supply Chain system is the main system for procurement, warehousing, shipping and any other associated supply chain-related functionality.
- The initialization of the solution requires Maintenix to have a copy of each warehouse rotable component and inventory record. This means a one-time data migration activity must be performed to extract each warehouse rotable inventory record and upload them into IFS Maintenix.
- Key updates to each serial number must be transferred to IFS Maintenix: location change, condition change, received date, manufacture date, ICN, ownership change and technical references updates.
- New serial numbers created from procurement workflows (purchase/repair/exchange/ consignment orders) should be created in IFS Maintenix as soon as they are deemed serviceable after Quality Control in the external supply chain system

The different business integration points could be summarized as shown in the image below.



Supporting maintenance planning and execution teams as well as providing visibility to material availability to engineering departments are key objectives this new API will achieve.

The overall solution is split into four (4) business workflows groups:

1. **Part Demand:**The group includes creating and updating part requests from IFS Maintenix, canceling part requests from IFS Maintenix, reservation notification from the external supply chain system, issuing of inventory from the external supply chain system and inventory turn in from IFS Maintenix
2. **Inventory Synchronization & Availability:** This group includes the creating and updating single inventory as well as providing a different material availability view from IFS Maintenix tabs (Part availability Tab, Part group availability tab, Work package parts tab, Deferred fault part request and 'Add Part Requirement' page).The API also supports inventory condition update from IFS Maintenix to the external supply chain system.
3. **Removals & Installs:** This group includes removed and installed components from IFS Maintenix task cards. The solution also supports the functionalities called 'detach inventory' and 'attach inventory' in IFS Maintenix.
4. **Part Synchronization:** This group includes creating part numbers, updating part numbers, activating part numbers and obsoleting part numbers. As mentioned above, support is provided for 2 different workflows depending on which system is the main system for part creation.

The following table shows the personas that are involved in the External Supply Chain:

Persona/System	Description
AOG Desk	An Aircraft on Ground desk plays a crucial role in managing part and maintenance requirements in order to return an aircraft to service.
Receiving Inspector	A person inspecting inventory after receipt. Based on the results of the inspection, the QC Inspector passes the inventory on to be shelved for put-away or to the un-serviceable staging area to be routed to repair.
Material Controller	A person responsible for ensuring that all parts and tools are available and issued on time. This allows scheduled and unscheduled maintenance activities to take place.
Buyer/Procurement Agent	A person responsible for acquiring the materials, components and services required to support aircraft maintenance. The buyer ensures timely availability and regulatory compliance.
QC Inspector	A Quality Control Inspector is responsible for ensuring all maintenance activities are carried out in compliance with regulatory, safety and organizational standards.
Storeroom Clerk	A person responsible for managing the physical inventory within a storeroom or warehouse.

The following table shows the personas that are involved in IFS Maintenix:

Persona/System	Description
Line Planner	A person responsible for planning and coordinating short-term maintenance activities that occur during an aircraft's turnaround time or daily operations.
Technical Records Clerk	A person responsible for the entry and maintenance, i.e: resolution of discrepancies, of aircraft technical records.

Persona/System	Description
MP Engineer	A Maintenance Program Engineer is responsible for managing and maintaining the aircraft maintenance program.
Configuration Engineer	A person responsible for defining, managing and maintaining the aircraft or fleet configuration data.
Base Planner	A person responsible for long-term heavy maintenance activities that are conducted during scheduled maintenance checks.
Maintenance Technician	A technician responsible for performing and signing off on aircraft maintenance activities.

2

Setting Up External Supply Chain API

The following sections provide setup and configuration information for the Supply Chain APIs.

Licensing and Subscription

A new Maintenix license feature has been added to use the API called EXT_SUPPLY_CHAIN_INV_SYNC_API.

To use the Supply Chain API, your organization must hold a valid license for it. Therefore, the Supply Chain API feature must be added to your Maintenix license.xml. If your current license does not include this feature, contact your IFS representative.

> **Note** license.xml resides in the /domain/config/maintenix folder, on the Maintenix application server host.

System Initialization

New customers initializing Maintenix for the first time need to migrate the allowable and current configuration data into the system.

For customers with an existing Maintenix installation, no special data migration is required to enable External Supply Chain.

Granting Security Permissions

In order to invoke the Supply Chain API, the Maintenix user must be granted the relevant permissions.

About this task

There are configuration parameters that need to be enabled for each of the integration points of the Supply Chain API to be successfully invoked.

Parameter Name	Description
SC_API_CREATE_UPDATE_INVENTORY	Permission to access the SC Create Update Inventory API.
SC_API_ISSUE_INVENTORY	Permission to access the SC Issue Inventory API.
SC_API_UPDATE_INVENTORY_SUMMARY	Permission to access the SC Update Inventory Summary API.
SC_API_UPDATE_PART_REQUEST	Permission to access the SC Update Part Request API.

Steps in Maintenix

1. Log in as a user with the **ADMIN** role.
2. Go to **Administrator**, click **Role Search**.
3. Search for the role that you want to grant permission to, then select it from the list.
4. On the **Role Details** page, click the **Permissions** tab.
5. Search and select the parameters listed above.
6. Click **Enable**.

External Supply Chain API Authentication

Authentication for integration messages to Maintenix is handled in two different ways depending on whether the Supply Chain API is deployed in IFS Maintenix context (basic authentication) or within IFS Cloud (IAM/OIDC). For set up Authentication with IFS Cloud, please refer to Appendices 9.3 Example of Authentication (IAM/OIDC) within IFS Cloud.

About this task

The invoking of the Supply Chain API authentication is restricted to the organization that have a valid Maintenix license, and for user roles that the security permissions are granted to.

Activate Alerts

When the Supply Chain API encounters an error, the business user is notified through IFS Maintenix Alert feature.

About this task

You can activate, deactivate and set alert priority of the Supply Chain API alerts:

Alert ID	Alert Name
109180	Supply Chain outbound message generation failure.
109179	Supply Chain outbound message trigger failure.
109139	Supply Chain No Allowable part.

Steps in IFS Maintenix

1. Go to the **Alerts Setup** page.
2. Click the Category drop-down list and select "**SCAPI**".
3. Select the check box beside the alerts: **Supply Chain outbound message generation failure, Supply Chain outbound message trigger failure** and **Supply Chain No Allowable part**.
4. Click **Activate Alert**.

> **Note**
>
> You can edit the roles assigned to alerts by selecting the check box beside the alert and clicking Edit Role Assignments. From this page you can select, or clear roles as required.

Outbound Event Framework

Enabling the PubSub Framework

PubSub is the Maintenix publish-subscribe message delivery framework. The PubSub framework operates as a message broker and facilitates communication between the Supply Chain API and the External Supply Chain system. For steps on enabling the PubSub framework, see the Maintenix PubSub User Guide.

Subscribing for Outbound Events

In order to receive notifications from the Supply Chain API, the External Supply Chain system's outbound message notification must be configured as a subscriber to the message channels for outbound events by updating the pubsub-subscriptions.properties file.

For the standalone Maintenix environment, please refer the samples of pubsub-subscription.properties and servers.properits files are found in the Appendices 9.1 and 9.2.

For the ICAM (IFS Cloud for Aviation Maintenance), environment, please refer Appendices 8.4 Example of PubSub Configuration Within IFS Cloud.

> **Note** A subscription can be added for each event through the HTTP notification model. For steps on how to set up the subscription, see the Maintenix PubSub User Guide.

Supply Chain API outbound events with corresponding PubSub channels:

Outbound Event	PubSub Channel
PartRequestCreatedUpdated	MX_PART_REQUEST_CREATED_UPDATED
PartRequestCancelled	MX_PART_REQUEST_CANCELLED
InventoryTurnedIn	MX_INVENTORY_TURNED_IN
InventoryConditionUpdated	MX_INVENTORY_CONDITION_UPDATED
InventoryRemovedInstalled	MX_REMOVAL_INSTALL

Activating Outbound Events

You must activate the outbound events for the Supply Chain API by enabling the applicable database triggers in the UTL_TRIGGER table:

Outbound Event – PartRequestCreatedUpdated.

Trigger_cd	Trigger_name	Class_name
MX_TS_SCHEDULE_INTERNAL	- Publish part request message by work order. - Trigger Point: When scheduling a work package internally. - Not applicable to adhoc part requests.	com.mxi.mx.scapi.trigger.partrequestcreatedupdated.PartRequestCreatedUpdatedTrigger

Trigger_cd	Trigger_name	Class_name
MX_TS_SCHEDULE_EXTERNAL	- Publish part request message by work order. - Trigger Point: When scheduling a work package externally. - Not applicable to adhoc. part request.	com.mxi.mx.scapi.trigger.partrequestcreatedupdated.PartRequestCreatedUpdatedTrigger
MX_TS_COMMIT	- Publish part request message by work order. - Trigger Point: When a work package is committed. - Not applicable to adhoc part request.	com.mxi.mx.scapi.trigger.partrequestcreatedupdated.PartRequestCreatedUpdatedTrigger

Trigger_cd	Trigger_name	Class_name
MX_TS_UNCOMMIT	- Publish part request message by work order. - Trigger Point: When a work package is uncommited. - Not applicable to adhoc part request.	com.mxi.mx.scapi.trigger.partrequestcreatedupdated.PartRequestCreatedUpdatedTrigger
MX_TS_REQUEST_PARTS	- Publish part request message by work order. - Trigger Point: When parts are requested. - Not applicable to adhoc part requests.	com.mxi.mx.scapi.trigger.partrequestcreatedupdated.PartRequestCreatedUpdatedTrigger

Trigger_cd	Trigger_name	Class_name
MX_TS_TASKASSIGN:	- Publish part request message by task. - Trigger Point: When you assign task to or unassign task from work package. - Not applicable to adhoc part requests.	com.mxi.mx.scapi.trigger.partrequestcreatedupdated.PartRequestCreatedUpdatedTrigger
MX_TS_ADD_PART_REQUIREMENT:	- Publish part request message by part requirement. - Trigger Point: When a part requirement is added to a task. - Not applicable to adhoc part requests.	com.mxi.mx.scapi.trigger.partrequestcreatedupdated.PartRequestCreatedUpdatedTrigger

Trigger_cd	Trigger_name	Class_name
MX_PR_PRIORITY:	- Publish part request message after updating the part request priority. - Applicable to both task and adhoc part requests.	com.mxi.mx.scapi.trigger.partrequestcreatedupdated.PartRequestCreatedUpdatedTrigger
MX_PR_EXTERNAL_REFERENCE	- Publish part request message after modifying a part request external reference. - Applicable to both task and adhoc part requests.	com.mxi.mx.scapi.trigger.partrequestcreatedupdated.PartRequestCreatedUpdatedTrigger

Trigger_cd	Trigger_name	Class_name
MX_PR_NEEDED_BY_DATE	- Publish part request message after updating the task scheduled start (which automatically updates the part request needed by date). - Applicable to both task and adhoc part requests.	com.mxi.mx.scapi.trigger.partrequestcreatedupdated.PartRequestCreatedUpdatedTrigger
MX_PR_AVAIL	- Publish part request message part request status has changed to AVAIL. - Applicable to both task and adhoc part requests.	com.mxi.mx.scapi.trigger.partrequestcreatedupdated.PartRequestStatusUpdateTrigger

Trigger_cd	Trigger_name	Class_name
MX_PR_OPEN:	- Publish part request message when part request status has changed to OPEN. -Applicable to both task and adhoc part requests.	com.mxi.mx.scapi.trigger.partrequestcreatedupdated.PartRequestCreatedUpdateTrigger
MX_PR_CREATE	- A new adhoc part request or new split batch part request is created.	com.mxi.mx.scapi.trigger.partrequestcreatedupdated.PartRequestCreatedTrigger

Outboud Event – PartRequestCancelled.

Trigger_cd	Trigger_name	Class_name
MX_PR_CANCEL	- Remove part requirement from task. Part request must be tied with the removed part requirement; Task must be in a work package. - Cancel task part request directly. -For a Part Request with a type of TASK, the Part request must be tied to a task in a work package.	com.mxi.mx.scapi.trigger.partrequestcancelled.PartRequestCancelledTrigger
MX_TS_CANCEL	Task is cancelled	com.mxi.mx.scapi.trigger.partrequestcancelled.PartRequestCancelledTrigger

Outbound Event – InventoryRemovedInstalled.

Trigger_cd	Trigger_name	Class_name
MX_TS_COMPLETE	Publish supply chain inventory removed/installed message for task completion.	com.mxi.mx.scapi.trigger.inventoryremovedinstalled.TaskCompleteTrigger
MX_IN_ATTACH	Publish supply chain inventory removed/installed message on inventory attach.	com.mxi.mx.scapi.trigger.inventoryremovedinstalled.InventoryAttachTrigger
MX_IN_DETACH	Publish supply chain inventory removed/installed message on inventory detach.	com.mxi.mx.scapi.trigger.inventoryremovedinstalled.InventoryDetachTrigger
MX_TS_FINISHLABOUR	Publish supply chain inventory removed/installed message for labour finish.	com.mxi.mx.scapi.trigger.inventoryremovedinstalled.PauseFinishLaboutTrigger
MX_TS_PAUSE	Publish supply chain inventory removed/installed message for labour pause.	com.mxi.mx.scapi.trigger.inventoryremovedinstalled.PauseFinishLabourTrigger

Outbound Event – InventoryConditionUpdated.

Trigger_cd	Trigger_name	Class_name
MX_IN_CONDITION	Publish inventory condition update message when inventory condition changes.	com.mxi.mx.scapi.trigger.inventoryconditionupdated.InventoryConditionUpdatedTrigger

Outbound Event – InventoryTurnedIn.

Trigger_cd	Trigger_name	Class_name
MX_LX_COMPLETE	Publish inventory turned in message after a TURNIN transfer is complete.	com.mxi.mx.scapi.trigger.inventoryturnedin.InventoryTurnedInTrigger

Mandatory Configuration Parameters Setup

The UTL_CONFIG_PARM table in Maintenix contains the Parameters that configure the Supply Chain API functionality.

PARM_NAME	PARM_VALUE	PARM_DESC
ENABLE_EXTSC_INVENTORY_ SUMMARY	TRUE/FALSE	The value must be configured to TRUE in order to hide the existing availability data and show the material availability data. When **ENABLE_EXTSC_ INVENTORY_SUMMARY** is **TRUE**, the following Maintenix pages will display part availability data from the External Inventory Summary (sc_inv_summary and sc_inv_summary_attr) tables instead of the core Maintenix inventory tables for the end user: 1. Part Availability tab in the Part Details page 2. Part Group Details page 3. Parts tab in the Work Package Details page 4. Deferred Fault Part Request page 5. Add Part Requirement page
CHANGE_PARTREQUEST_TO _PRAVAIL	TRUE/FALSE	The value must be configured to TRUE to support the following: When Create Update Inventory inbound message changes the location of the reserved inventory from the remote to the local, the status of the part request will be changed from REMOTE to AVAIL.
DISABLE_REMOTE_ RESERVATION_SHIPMENTS	TRUE/FALSE	The value must be configured to TRUE in order to support the UpdatePartRequest inbound message.

PARM_NAME	PARM_VALUE	PARM_DESC
EXT_SC_BATCH_ENABLED	TRUE/FALSE	The value must be configured to TRUE to support the following: This parameter controls batch inventory allocation behavior when using any external supply chain: TRUE will prevent taking reserved batch inventory from other part requests and reduces the available unreserved batch inventory quantity.
SUPPLY_CHAIN_SYSTEM	EXTERNAL/IFS	The value must be configured to EXTERNAL to support the following: Determines which supply chain system is used with Maintenix.

Mandatory Job Setup

Update the below Maintenix job to populate the deferred fault part requests tab.

JOB_CD	ACTIVE_BOOL
MX_CORE_MVIEW_EXTERNAL_DEFER_ FAULT_PART_REQ	1/0 - Refresh the External Deferred Fault Part Request MView active_bool: The value must be configured to 1 to enable the job.

3

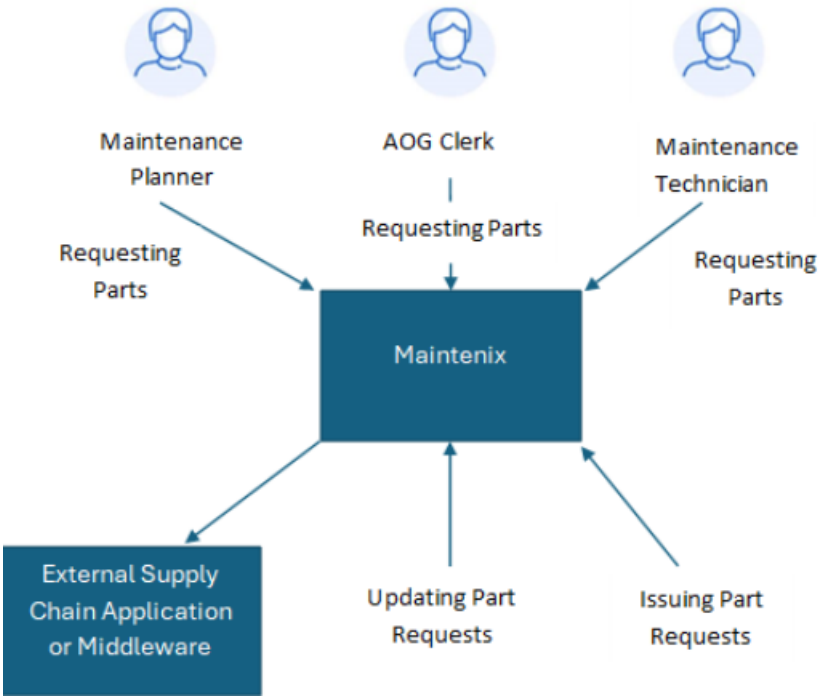
Open API Specification

You can obtain the Supply Chain API Open API Specification from a running Maintenix instance at the URL **http://<host:port>/scapi/openapi.json**. The user that will log in requires the **API_VIEW_REQUEST** permission. It is also bundled with this guide under API Schemas/ scapiopenapi. json.

4

Part Demand

The following diagram shows how the personas and systems interact in the external supply chain:



External Supply Chain API Scenarios.

Each of the supported scenarios contain the following sections:

- **Goal:** A short statement that defines the purpose of the scenario.
- **Pre-conditions:** A list of conditions that must be met before the scenario can be implemented.
- **Messaging**: A basic description of the messaging required for the scenario.
- **Result:** The outcome of the successful execution of the scenario.

Create/Update Part Request

Outbound Message - PartRequestCreatedUpdated

Publishing part requests raised in Maintenix.

Goal: Publish the details of a part request raised in Maintenix to an external system automatically.

Business Scenarios:

- Update the part request quantity by creating a new part requirement or new part request.
- Add a specific part number on the part requirement.

Summary: The **PartRequestCreatedUpdated** outbound message is sent by Maintenix to the Supply Chain system.

Pre conditions:

- A work package exists on an aircraft or a component with non historic JICS that have part requirements with non historic part requests.
- A work package exists on an aircraft or a component with non historic faults that have part requirements with non historic part requests
- A work package exists on an aircraft or component with non historic ad hoc tasks that have part requirements with non historic part requests.
- A work package exists on an aircraft or a component with non historic executable requirements that have part requirements with non historic part requests.
- An ad hoc part request.
- A part request is raised in Maintenix.

Messaging: Maintenix publishes a **PartRequestCreatedUpdated** outbound message for part requests created through work packages and adhoc part requests.


```
{
    "metadata": {
        "correlationId": "1ea225bd-33c8-4e10-a820-26f4ded53f7a",
        "messageId": "381bcb27-0fb7-44dd-8a3b-26bfd1b31549",
        "system": "MAINTENIX",
        "user": "mxintegration",
        "timestamp": "2025-02-28T13:04:35.898Z"
    },
    "requestedParts": [{
            "task": {
                "taskIdentifier": {
                    "barcode": "T40S001EBDP"
                },
                "taskName": "23-11-21-11-JIC (23-11-21-11-JIC)",
                "type": "JIC",
                "creationDate": "2025-02-28T12:55:36.000Z"
            },
```

```
            "masterReqId": "R48U000KB8R",
            "requestId": "R48U000KB8R",
            "requestType": "TASK",
            "priority": "NORMAL",
            "quantity": 1,
            "status": "PRAVAIL",
            "neededByDate": "2025-02-28T12:56:00.000Z",
            "configuration": {
                "assemblyCode": "737-NG",
                "ataCode": "23-11-21-11",
                "partGroup": {
                    "code": "23-11-21-11"
                }
            },
            "partDefinition": [{
                    "part": {
                        "oemPartNumber": "822-0990-004",
                        "manufacturerCode": "4V792"
                    },
                    "standard": false,
                    "specificallyRequested": false,
                    "partName": "TRANSCEIVER ASSY - HF COMM (CMA-
RC)",
                    "standardUnitOfMeasurement": "EA",
                    "otherConditions": false
                }, {
                    "part": {
                        "oemPartNumber": "MC10-25-104",
                        "manufacturerCode": "F5341"
                    },
                    "standard": false,
                    "specificallyRequested": false,
                    "partName": "*CR*MASK ASSY - OXY REGULATOR",
                    "standardUnitOfMeasurement": "EA",
                    "otherConditions": false
                }, {
                    "part": {
                        "oemPartNumber": "FE374300001",
                        "manufacturerCode": "90025"
                    },
                    "standard": false,
                    "specificallyRequested": false,
                    "partName": "ACTUATOR, ASSY (RMSA)",
                    "standardUnitOfMeasurement": "EA",
                    "otherConditions": false
                }
            ],
            "notes": "This part request was created automatically upon
 task creation.",
            "issueToAccountCode": "BOLO_5145800045",
            "reservedInventory": {
                "part": {
                    "oemPartNumber": "822-0990-004",
                    "manufacturerCode": "4V792"
                },
```

```
                "oemSerialNumber": "SN011020",
                "uniqueId": "7DCD5DB3F5D311EFB0376045BD61E2AD"
            },
            "neededLocation": {
                "neededLocationCode": "ADZ/LINE"
            }
        }
    ],
    "rootInventory": {
        "inventoryId": {
            "part": {
                "oemPartNumber": "737-7V3",
                "manufacturerCode": "81205"
            },
            "oemSerialNumber": "30049",
            "uniqueId": "4348A120ABAB11E7BDEEA3C18A64A95E"
        },
        "assemblyName": "AIRLINE",
        "aircraftRegistrationCode": "IFSAD-1371",
        "ownerCode": "IFS",
        "inventoryClassCode": "ACFT"
    },
    "authorizationInfo": {
        "datetime": "2025-02-28T13:04:35.827Z",
        "authority": "mxsystem",
        "firstName": "mxsystem",
        "lastName": "mxsystem"
    },
    "workOrder": {
        "workOrderIdentifier": {
            "barcode": "T40S001EBDM",
            "internalId": "4650:6062694",
            "workOrderNumber": "WO - 6062694",
            "workOrderName": "dd"
        },
        "status": "COMMIT",
        "location": {
            "locationCode": "ADZ/LINE",
            "locationName": "Line Maintenance"
        },
        "scheduledStartDate": "2025-02-28T12:56:00.000Z"
    },
    "airport": {
        "code": "ADZ",
        "name": "Gustavo Rojas Pinilla"
    }
}
```


Result: The **PartRequestCreatedUpdated** outbound message is expected to publish a message that contains information on requested parts.

Note The SC **PartRequestCreatedUpdated** outbound message now supports KIT part numbers. Although KIT contains part numbers, they are currently not included in the **PartRequestCreatedUpdated** outbound message. Aside from that, it functions like any other part.

Publishing part requests updated in Maintenix.

Goal: Publish the details of a part request updated in Maintenix to an external system automatically.

Business Scenarios:

- Update the part request priority.
- Update the part request status.
- Update the part request needed by date.
- Update the part request needed location.

Summary: The **PartRequestCreatedUpdated** outbound message is sent by Maintenix to the Supply Chain system.

Pre conditions:

- A work package exists on an aircraft or a component with non historic JICS that have part requirements with non historic part requests.
- A work package exists on an aircraft or a component with non historic faults that have part requirements with non historic part requests
- A work package exists on an aircraft or component with non historic ad hoc tasks that have part requirements with non historic part requests.
- A work package exists on an aircraft or a component with non historic executable requirements that have part requirements with non historic part requests.
- An ad hoc part request.
- A part request is updated in Maintenix. *For example, Part Request Priority, Part Request Status; Part Request Needed By Date, Part Request Needed Location etc*.

Messaging: Maintenix publishes a **PartRequestCreatedUpdated** outbound message.


```
{
    "metadata": {
        "correlationId": "1ea225bd-33c8-4e10-a820-26f4ded53f7a",
        "messageId": "381bcb27-0fb7-44dd-8a3b-26bfd1b31549",
        "system": "MAINTENIX",
        "user": "mxintegration",
        "timestamp": "2025-02-28T13:04:35.898Z"
    },
    "requestedParts": [{
```

```
            "task": {
                "taskIdentifier": {
                    "barcode": "T40S001EBDP"
                },
                "taskName": "23-11-21-11-JIC (23-11-21-11-JIC)",
                "type": "JIC",
                "creationDate": "2025-02-28T12:55:36.000Z"
            },
            "masterReqId": "R48U000KB8R",
            "requestId": "R48U000KB8R",
            "requestType": "TASK",
            "priority": "NORMAL",
            "quantity": 1,
            "status": "PRAVAIL",
            "neededByDate": "2025-02-28T12:56:00.000Z",
            "configuration": {
                "assemblyCode": "737-NG",
                "ataCode": "23-11-21-11",
                "partGroup": {
                    "code": "23-11-21-11"
                }
            },
            "partDefinition": [{
                    "part": {
                        "oemPartNumber": "822-0990-004",
                        "manufacturerCode": "4V792"
                    },
                    "standard": false,
                    "specificallyRequested": false,
                    "partName": "TRANSCEIVER ASSY - HF COMM (CMA-
RC)",
                    "standardUnitOfMeasurement": "EA",
                    "otherConditions": false
                }, {
                    "part": {
                        "oemPartNumber": "MC10-25-104",
                        "manufacturerCode": "F5341"
                    },
                    "standard": false,
                    "specificallyRequested": false,
                    "partName": "*CR*MASK ASSY - OXY REGULATOR",
                    "standardUnitOfMeasurement": "EA",
                    "otherConditions": false
                }, {
                    "part": {
                        "oemPartNumber": "FE374300001",
                        "manufacturerCode": "90025"
                    },
                    "standard": false,
                    "specificallyRequested": false,
                    "partName": "ACTUATOR, ASSY (RMSA)",
                    "standardUnitOfMeasurement": "EA",
                    "otherConditions": false
                }
            ],
```

```
            "notes": "This part request was created automatically upon
 task creation.",
            "issueToAccountCode": "BOLO_5145800045",
            "reservedInventory": {
                "part": {
                    "oemPartNumber": "822-0990-004",
                    "manufacturerCode": "4V792"
                },
                "oemSerialNumber": "SN011020",
                "uniqueId": "7DCD5DB3F5D311EFB0376045BD61E2AD"
            },
            "neededLocation": {
                "neededLocationCode": "ADZ/LINE"
            }
        }
    ],
    "rootInventory": {
        "inventoryId": {
            "part": {
                "oemPartNumber": "737-7V3",
                "manufacturerCode": "81205"
            },
            "oemSerialNumber": "30049",
            "uniqueId": "4348A120ABAB11E7BDEEA3C18A64A95E"
        },
        "assemblyName": "AIRLINE",
        "aircraftRegistrationCode": "IFSAD-1371",
        "ownerCode": "IFS",
        "inventoryClassCode": "ACFT"
    },
    "authorizationInfo": {
        "datetime": "2025-02-28T13:04:35.827Z",
        "authority": "mxsystem",
        "firstName": "mxsystem",
        "lastName": "mxsystem"
    },
    "workOrder": {
        "workOrderIdentifier": {
            "barcode": "T40S001EBDM",
            "internalId": "4650:6062694",
            "workOrderNumber": "WO - 6062694",
            "workOrderName": "dd"
        },
        "status": "COMMIT",
        "location": {
            "locationCode": "ADZ/LINE",
            "locationName": "Line Maintenance"
        },
        "scheduledStartDate": "2025-02-28T12:56:00.000Z"
    },
    "airport": {
        "code": "ADZ",
        "name": "Gustavo Rojas Pinilla"
    }
}
```

Result: The **PartRequestCreatedUpdated** outbound message is expected to update a message that contains information on requested parts.

Update Part Request

Inbound Message i.e. from the external Supply Chain system to Maintenix - UpdatePartRequest.

Updating the notes on a Part Request in Maintenix

Goal: Update or provide notes on a part request without changing the current status of the part request.

Summary: The external system can send notes that have been manually entered by a user or automatically by the external system on a part request.

This scenario might arise when an external system would like to record an acknowledgment of the part request.

Pre conditions:

- API Permission **SC_API_UPDATE_PART_REQUEST** should be enabled.
- Valid non-historic part request exists in Maintenix.

Messaging: The external supply chain system sends an **UpdatePartRequest** inbound message to update the notes on the part request that is raised. The status of the part request does not change.


```
{
  "metadata": {
    "correlationId": "1ea225bd-33c8-4e10-a820-26f4ded53f7a",
    "messageId": "381bcb27-0fb7-44dd-8a3b-26bfd1b31549",
    "system": "External Supply Chain",
    "user": "mxintegration",
  "timestamp": "2022-11-09T11:55:18.602835000Z"
  },
  "partRequest": {
    "id": {
      "masterReqId": "R48U000KB8R",
      "requestId": "R48U000KB8R",
```

```
      "externalId": "test ext id
    },
    "status": "PROPEN",
    "partRequestNotes": "test note 4"
  }
}
```

Result:

- The part request notes are updated.
- The part request notes can be viewed on the **Part Request Details** page.

Note The **UpdatePartRequest** inbound message now supports KIT inventory. KIT contents are currently not included in the **UpdatePartRequest** message, Aside from that, it functions like any other part.

Estimated Arrival Date Update

Goal: To identify when the reserved inventory will be available.

Summary: The external supply chain system identifies the day by which the reserved inventory will arrive at the warehouse. This could be a date provided by a vendor/supplier from a purchase order or a shipment if the component is sent from another airport.

Pre conditions:

- API Permission **SC_API_UPDATE_PART_REQUEST** should be enabled.
- Valid non-historic part request exists in Maintenix.

Messaging: The external supply chain system sends an **UpdatePartRequest** inbound message to update the Estimated Arrival Date on the part request.

```
{
  "metadata": {
    "correlationId": "1ea225bd-33c8-4e10-a820-26f4ded53f7a",
    "messageId": "381bcb27-0fb7-44dd-8a3b-26bfd1b31549",
    "system": "External Supply Chain",
    "user": "mxintegration",
```

```
  "timestamp": "2022-11-09T11:55:18.602835000Z"
  },
  "partRequest": {
    "id": {
      "masterReqId": "R48U000KB8R",
      "requestId": "R48U000KB8R"
    },
    "status": "PROPEN",
    "estimatedArrival": "2024-10-15T17:39:28.160Z"
  }
}
```

Result:

- The part request estimated arrival date is updated.
- The updated estimated arrival date can be viewed on the **Task Details** page and the **Part Request Details** page.

Note The **UpdatePartRequest** inbound message now supports KIT inventory. KIT contents are currently not included in the **UpdatePartRequest** message. Aside from that, it functions like any other part.

Reserve Serialized Inventory on a Part Request

Goal: To reserve serialized inventory on a part request.

Summary: The external supply chain system identifies the available serialized inventory.

Pre-conditions:

- API Permission **SC_API_UPDATE_PART_REQUEST** should be enabled.

Messaging: An **UpdatePartRequest** inbound message is sent to reserve serialized inventory through a part request.

```
{
    "metadata": {
        "correlationId": "correlationId-123",
```

```
        "messageId": "MessageId-123",
        "system": "mxi",
        "timestamp": "2025-08-19T11:55:18.602835000Z",
        "user": "mxi"
    },
    "partRequest": {
        "id": {
            "masterReqId": "R48U00025NG",
            "requestId": "R48U00025NG"
        },
        "status": "PRAVAIL",
        "inventory": {
            "serialized": {
                "part": {
                    "oemPartNumber": "SERIALIZEDPART",
                    "manufacturerCode": "C0890"
                },
                "oemSerialNumber": "SERIALNO",

           }
        }
    }
}
```

Result:

- The inventory is expected to be reserved for the part request.
- The reserved inventory (local and remote) can be viewed on the **Part Request Details** page.

Note The **UpdatePartRequest** inbound message now supports KIT inventory. KIT contents are currently not included in the **UpdatePartRequest** message. Aside from that, it functions like any other part.

Reserve Batch on a Part Request

- Batch inventory is created with the batch number provided by the Supply Chain system in **oemBatchTag**. This process takes place when reserving a batch inventory from the inbound **UpdatePartReqest** message.
- The **oemBatchTag** is mandatory and the batch number will be generated based on its value in the message.
- As all batch numbers in Maintenix are in upper case, the **oemBatchTag** value will be converted to UPPER CASE and stored in the batch number field. This batch number will appear on the SN/BN field on the work capture page for technicians.
    - The system will match each batch inventory with the following combination: part number, manufacturer code and the batch number in upper case.

    - If no batch inventory exists or the batch inventory exists at a different location:
        - Create a new batch inventory using the upper case value of oemBatchTag as the batch number. E.g. TEST BATCH TAG.
        - Reserve the inventory for the part request.
        - Inventory Details Page: OEM Batch Number refers to the value of oemBatchTag provided in the UpdatePartRequest message.
        - Inventory Details Page: OEM Lot Number refers to the value of oemLotTag provided in the UpdatePartRequest message.
    - If a RFI batch inventory exists for the same part at the same location:
        - Increase the quantity of the existing batch inventory by the amount specified in the message. Validate the updated batch inventory. Reserve the updated inventory for the part request.
- Multiple part requests (across different tasks, work packages, or aircraft) can be reserved against the same batch inventory, each maintaining its own reserved quantity.

Goal: To reserve batch inventory on a part request.

Summary: The external supply chain system identifies the available batch inventory.

Pre condition: API Permission **SC_API_UPDATE_PART_REQUEST** should be enabled.

Message: An **UpdatePartReqest** message is sent to reserve a batch on a part request.

```
{
    "metadata": {
        "correlationId": "correlationId-123",
        "messageId": "MessageId-123",
        "system": "mxi",
        "timestamp": "2022-12-17T11:55:18.602835000Z",
        "user": "mxi"
    },
    "partRequest": {
        "id": {
            "masterReqId": "R48U000KB79",
            "requestId": "R48U000KB79"
        },
        "status": "PRAVAIL",
        "inventory": {
            "batch": {
                "part": {
                    "oemPartNumber": "24H0168-005",
                    "manufacturerCode": "00000"
                },
                "oemBatchTag": "BatchTag-04",
                "quantity": 6
            }
        }
    }
```

```
}
```

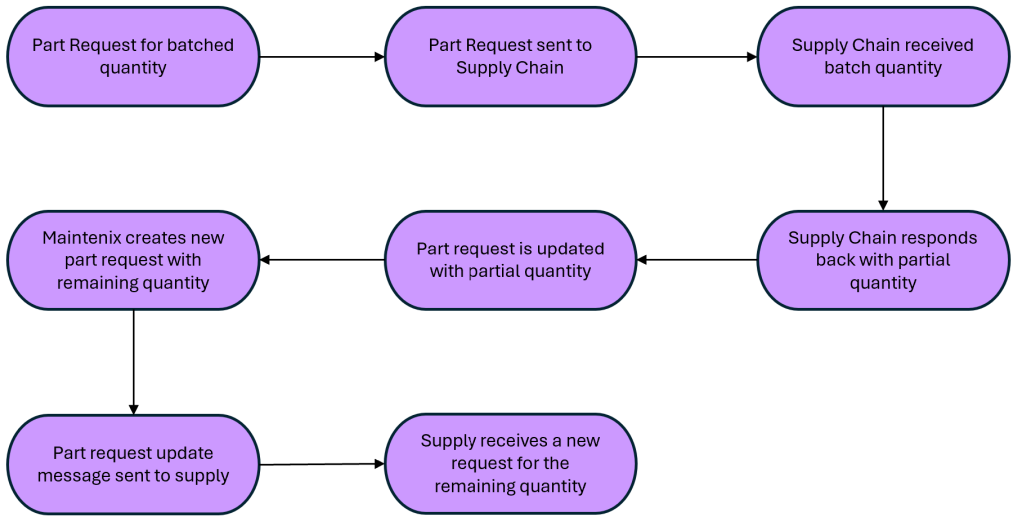
Result:

- The inventory is expected to be reserved for the part request.
- The reserved inventory (local and remote) can be viewed on the **Part Request Details** page.

Split Batch

The Split Batch process takes place when the Supply Chain system is unable to fulfill the quantity required by the part request in one single batch. Instead, a partial quantity will be fulfilled. Typically, the total quantity will be fulfilled by multiple batches.

The process involves Maintenix splitting the part request and creating a new one for the remaining quantity. These new part requests are connected by a Master Part Request ID which is shown on the Part Request Details.



Step 1 : Requesting multiple batched inventory.

Goal: To fulfill a part request with multiple batch numbers.

Summary: A part request is sent to the external supply chain system for a batch part.

Messaging: A **PartRequestCreatedUpdated** message is sent to the supply chain system.

```
{
 "metadata": {
 "correlationId": "e0abf728-d8fe-449b-9f67-311b30fc6899",
```

```
"messageId": "eb9472d3-a3db-4b18-97eb-7de2cce04f2e",
 "system": "MAINTENIX",
"user": "mxintegration",
 "timestamp": "2025-05-04T09:27:26.475Z"
 },
"requestedParts": [
{
 "task": {
"taskIdentifier": {
 "barcode": "T40S001EB7A"
 },
 "taskName": "DEMO_BATCH_SPLIT",
 "type": "ADHOC",
 "creationDate": "2025-04-30T16:10:16.000Z"
 },
 "masterReqId": "R48U000KBDA",
 "requestId": "R48U000KBDA",
 "requestType": "TASK",
 "priority": "NORMAL",
 "quantity": 100,
 "status": "PROPEN",
 "neededByDate": "2025-04-29T08:14:00.000Z",
 "configuration": {
"assemblyCode": "COMHW",
 "ataCode": "COMHW",
 "partGroup": {
 "code": "CB200-40"
 }
```

```
},
"partDefinition": [
{
"part": {
 "oemPartNumber": "CB200-40",
 "manufacturerCode": "76381"
 },
 "standard": true,
 "specificallyRequested": false,
 "partName": "ACRYLIC ADHESIVE",
 "standardUnitOfMeasurement": "EA",
 "otherConditions": false
 }
],
 "notes": "This part request was created when the part was added to
 task 'DEMO_BATCH_SPLIT
[T40S001EB7A]'.",
"issueToAccountCode": "CONS_5125050090",
 "neededLocation": {
 "neededLocationCode": "ADZ/LINE"
},
 "requestedBy": {
 "firstName": "Admin",
 "lastName": "Admin"
}
 }
 ],
 "rootInventory": {
```

```
 "inventoryId": {
 "part": {
 "oemPartNumber": "737-7V3",
 "manufacturerCode": "81205"
},
 "oemSerialNumber": "28607",
 "uniqueId": "4348A11FABAB11E7BDEEA3C18A64A95E"
},
 "assemblyName": "AIRLINE",
"aircraftRegistrationCode": "IFSAD-1372",
 "ownerCode": "IFS",
 "inventoryClassCode": "ACFT"
},
 "authorizationInfo": {
 "datetime": "2025-05-04T09:27:26.405Z",
 "authority": "ADMIN",
 "notes": "Added part requirement(s) to task.",
 "firstName": "Admin",
 "lastName": "Admin"
 },
 "workOrder": {
 "workOrderIdentifier": {
 "barcode": "T40S001EB70",
"internalId": "4650:6062408",
 "workOrderNumber": "WO - 6062408",
 "workOrderName": "WP_BATCH_SPLIT_TEST"
},
```

```
"status": "IN WORK",
 "location": {
 "locationCode": "ADZ/LINE",
 "locationName": "Line Maintenance"
 },
 "scheduledStartDate": "2025-04-29T08:14:00.000Z"
 },
 "airport": {
 "code": "ADZ",
 "name": "Gustavo Rojas Pinilla"
 }
}
```

Result: The supply chain system receives and processes the request. This is detailed in Step 2.

Step 2: Supply Chain responds with a partial quantity.

Goal: To provide the partial quantity that can be fulfilled.

Summary: Once Maintenix sends the batch quantity to the supply chain system, it is reviewed and the partial quantity that can be fulfilled is provided.

Preconditions:

- API Permission **SC_API_CREATE_UPDATE_PART_REQUEST** should be enabled
- Valid non-historic part request exists in Maintenix.

Messaging: A **PartRequestCreatedUpdated** message is sent for the remaining quantity that can be updated by the Supply Chain system.

```
{
    "metadata": {
        "correlationId": "correlationId-123",
        "messageId": "MessageId-123",
        "system": "mxi",
        "timestamp": "2022-11-09T11:55:18.602835000Z",
```

```
        "user": "scapi"
    },
    "partRequest": {
        "id": {
            "masterReqId": "R48U000KBDA",
            "requestId": "R48U000KBDA",
            "externalId": "test ext id"
        },
        "status": "PRAVAIL",
        "estimatedArrival": "",
        "partRequirementNotes": "Test",
        "inventory": {
            "batch": {
                "part": {
                    "oemPartNumber": "CB200-40",
                    "manufacturerCode": "76381"
                },
                  "oemBatchTag": "External SC Batch Number",
                   "quantity":35
                }
           }
      }
}
```

Result:

- The part request is updated with the partial quantity in Maintenix. The part request status changes to **AVAIL**
  - A **PartRequestCreatedUpdated** message is sent to the supply chain system with the status as **AVAI**L More details can be found in **Step 3**.
- Maintenix splits the part request and creates a new part request for the remaining quantity. Theses part requests are connected by a Master Part Request ID which is shown on the Part Request Details.

- A **PartRequestCreatedUpdated** message is sent for the remaining quantity that can be updated by Supply which will contain both the master part request ID and the new created part request ID: **PartRequestCreatedUpdated.requestedPart.masterReqId**

  PartRequestCreatedUpdated.requestedPart.ReqId

Step 3: Part Request Update message sent to Supply Chain.

This scenario includes two messages that are triggered by **Step 2**.

Message 1: A **PartRequestCreatedUpdated** message is sent with the status as **AVAIL.**

- **PRAVAIL**


```
{
"metadata": {
  "correlationId": "731d5a08-e526-4b90-927b-02b8061eefa4",
  "messageId": "2cfb58e5-1b30-4f1a-9bf6-3e80ccebb73c",
  "system": "MAINTENIX",
  "user": "mxintegration",
  "timestamp": "2025-04-10T12:18:31.401Z"
},
"requestedParts": [
  {
   "task": {
    "taskIdentifier": {
     "barcode": "T40S0001DND"
    },
    "taskName": "321002203-REPL (ANTI SKD NORM REPLACEMENT)",
    "type": "REPL",
    "creationDate": "2025-04-10T06:45:42.000Z"
   },
   "masterReqId": "R48U00025MG",
   "requestId": "R48U00025MG",
```

```
"externalId": "test ext id",
"requestType": "TASK",
"priority": "NORMAL",
"quantity": 35,
"status": "PRAVAIL",
"neededByDate": "2002-02-21T08:27:00.000Z",
"configuration": {
 "assemblyCode": "CONSUM",
 "ataCode": "CONSUM",
 "partGroup": {
  "code": "0912ALTEST"
 }
},
"partDefinition": [
 {
  "part": {
   "oemPartNumber": "0912ALTEST",
   "manufacturerCode": "00624"
  },
  "standard": true,
  "specificallyRequested": false,
  "partName": "0912 AL Test Part A",
  "standardUnitOfMeasurement": "EA",
  "otherConditions": false
 }
],
```

```
  "notes": "This part request was created when the part was
 added to task '321002203-REPL (ANTI SKD NORM REPLACEMENT)
 [T40S0001DND]'.",
  "issueToAccountCode": "DEFAULT_EXPENSE",
  "reservedInventory": {
   "part": {
    "oemPartNumber": "0912ALTEST",
    "manufacturerCode": "00624"
   },
   "oemSerialNumber": "EXTERNAL SC BATCH NUMBER",
   "uniqueId": "E9714FB8160511F08CF14438E8172362"
  },
  "neededLocation": {
   "neededLocationCode": "YYZ/LINE"
  },
  "requestedBy": {
   "firstName": "MXI",
   "lastName": "MXI"
  }
 }
],
"rootInventory": {
 "inventoryId": {
  "part": {
   "oemPartNumber": "B767-232",
   "manufacturerCode": "1000"
  },
  "oemSerialNumber": "222",
  "uniqueId": "A00EE40D404B4CD1AED630860836A83D"
```

```
  },
  "assemblyName": "Boeing 767-200",
  "aircraftRegistrationCode": "222",
  "ownerCode": "A319/320",
  "inventoryClassCode": "ACFT"
},
"authorizationInfo": {
  "datetime": "2025-04-10T12:18:31.032Z",
  "authority": "mxi",
  "firstName": "MXI",
  "lastName": "MXI"
},
"workOrder": {
  "workOrderIdentifier": {
   "barcode": "T0000NRW",
   "internalId": "5000000:86774",
   "workOrderNumber": "WO - 86774",
   "workOrderName": "Maintenance Visit - Time Limted Parts"
  },
  "status": "IN WORK",
  "location": {
   "locationCode": "YYZ/LINE",
   "locationName": "Line"
  },
  "scheduledStartDate": "2002-02-21T02:27:00.000Z"
},
"airport": {
```

```
  "code": "YYZ",
  "name": "Toronto"
}
}
```

Message 2: A **PartRequestCreatedUpdated** message is sent for the remaining quantity that can be updated by the supply chain system. This will contain both the master requirement ID and the new created part request ID.

- **PROPEN** (remaining quantity : **65**).

```
{
 "metadata": {
 "correlationId": "124b0936-a615-4b49-bdea-805559e0cfe0",
 "messageId": "1d01a328-e27a-4b40-b808-a96706167f5c",
 "system": "MAINTENIX",
 "user": "mxintegration",
 "timestamp": "2025-05-01T02:01:52.622Z"
 },
 "requestedParts": [
  {
   "task": {
    "taskIdentifier": {
     "barcode": "T40S001EB7A"
    },
    "taskName": "DEMO_BATCH_SPLIT",
    "type": "ADHOC",
    "creationDate": "2025-04-30T16:10:16.000Z"
   },
   "masterReqId": "R48U000KBD4",
   "requestId": "R48U000KBD8",
   "requestType": "TASK",
   "priority": "NORMAL",
   "quantity": 65,
   "status": "PROPEN",
   "neededByDate": "2025-04-29T08:14:00.000Z",
   "configuration": {
    "assemblyCode": "COMHW",
    "ataCode": "COMHW",
    "partGroup": {
     "code": "190-15167-002"
    }
   },
   "partDefinition": [
    {
     "part": {
      "oemPartNumber": "190-15167-002",
      "manufacturerCode": "00000"
     },
     "standard": true,
     "specificallyRequested": false,
```

```
     "partName": "ANGLE RIB",
     "standardUnitOfMeasurement": "EA",
     "otherConditions": false
    }
   ],
   "notes": "This part request was created when the part was added
to task 'DEMO_BATCH_SPLIT [T40S001EB7A]'.",
   "issueToAccountCode": "CONS_5125050090",
   "neededLocation": {
    "neededLocationCode": "ADZ/LINE"
   },
   "requestedBy": {
    "firstName": "Admin",
    "lastName": "Admin"
   }
  }
 ],
 "rootInventory": {
 "inventoryId": {
   "part": {
    "oemPartNumber": "737-7V3",
    "manufacturerCode": "81205"
   },
   "oemSerialNumber": "28607",
   "uniqueId": "4348A11FABAB11E7BDEEA3C18A64A95E"
  },
 "assemblyName": "AIRLINE",
 "aircraftRegistrationCode": "IFSAD-1372",
 "ownerCode": "IFS",
 "inventoryClassCode": "ACFT"
 },
 "authorizationInfo": {
 "datetime": "2025-05-01T02:01:52.466Z",
 "authority": "ADMIN",
 "notes": "This part request was created when the part was added to
 task 'DEMO_BATCH_SPLIT [T40S001EB7A]'.",
 "firstName": "Admin",
 "lastName": "Admin"
 },
 "workOrder": {
 "workOrderIdentifier": {
   "barcode": "T40S001EB70",
   "internalId": "4650:6062408",
   "workOrderNumber": "WO - 6062408",
   "workOrderName": "WP_BATCH_SPLIT_TEST"
  },
 "status": "IN WORK",
 "location": {
   "locationCode": "ADZ/LINE",
   "locationName": "Line Maintenance"
  },
 "scheduledStartDate": "2025-04-29T08:14:00.000Z"
 },
 "airport": {
 "code": "ADZ",
```

```
  "name": "Gustavo Rojas Pinilla"
  }
}
```

Result: The supply chain system is expected to process a new request with the remaining quantity.

Prevent Stealing Reserved Batch From Another Part Request

When reserving batch inventory for a part request in Maintenix, if the available quantity is insufficient and the inventory is already reserved for other part requests, the Maintenix core reservation logic will unreserve it from those requests to fulfill the current one. The Supply Chain API employs a different approach: it splits the part request rather than taking inventory from other request.

The following example details how stealing reserved batch inventory can be prevented.

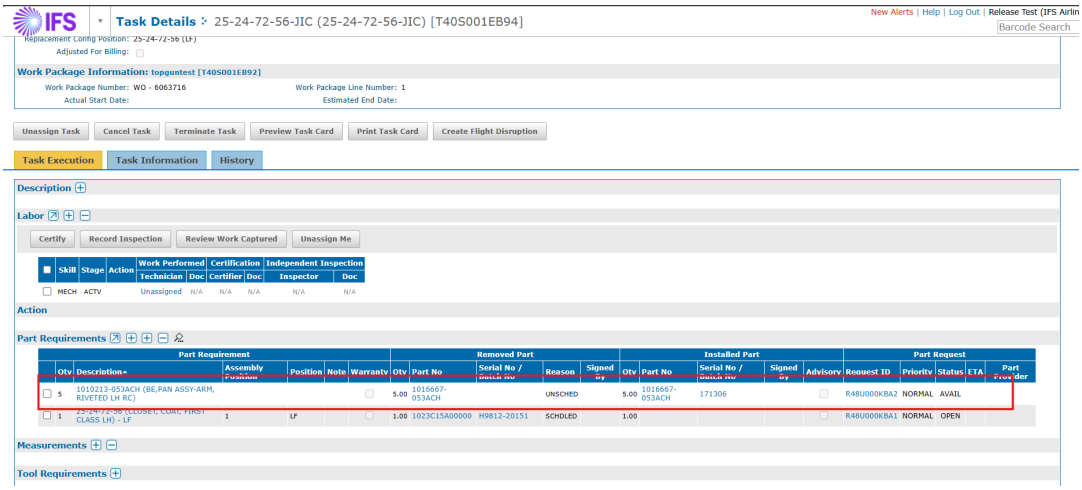
Goal: To prevent reserved batch inventory from being stolen to fulfill part requests.

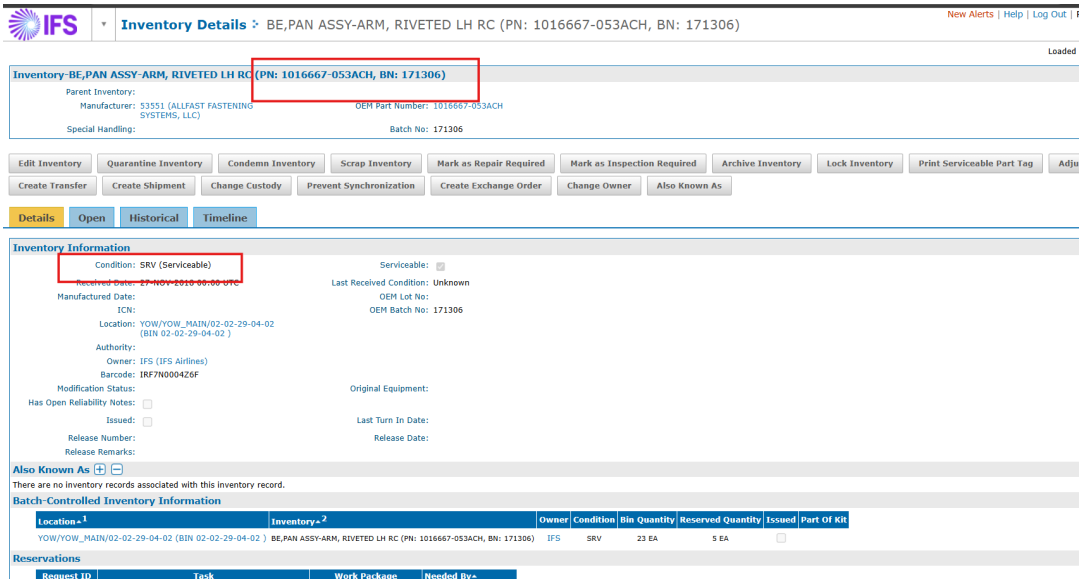
Summary: Part requests are split instead of stealing reserved Inventory from another request.

Pre-condition:

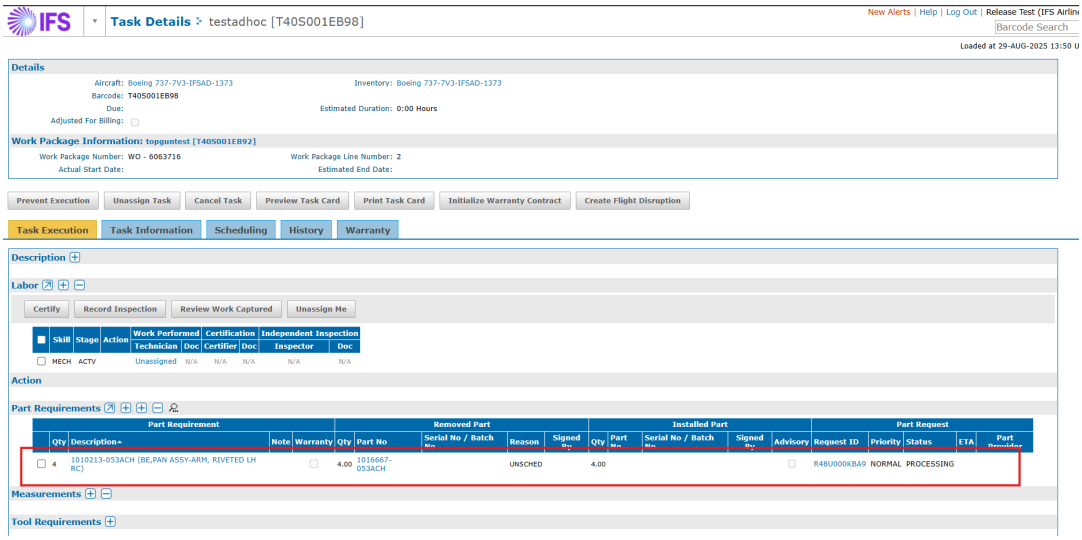
- **EXT_SC_BATCH_ENABLED** config parameter is enabled.

Step 1 : Create a workpackage -> task -> part requirement 1 with quantity of 5 and reserve all 5 batch inventory.





Step 2 : Create another task -> part requirement 2 a with a quantity of 4.



Step 3 : Send an **UpdatePartRequest** inbound message to reserve inventory for part requirement 2, but with lower quantity (quantity = 1) than the expected one.

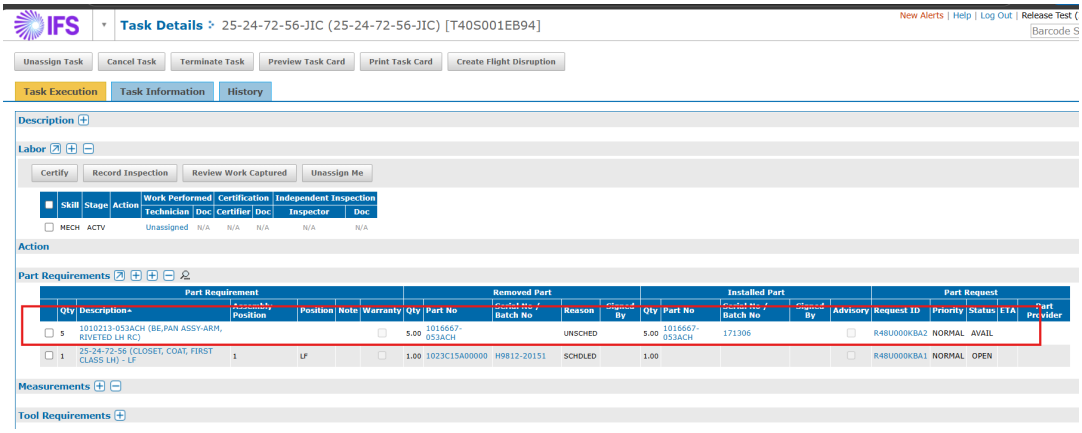
Messaging: An **UpdatePartRequest** message is sent from External Supply Chain to Maintenix

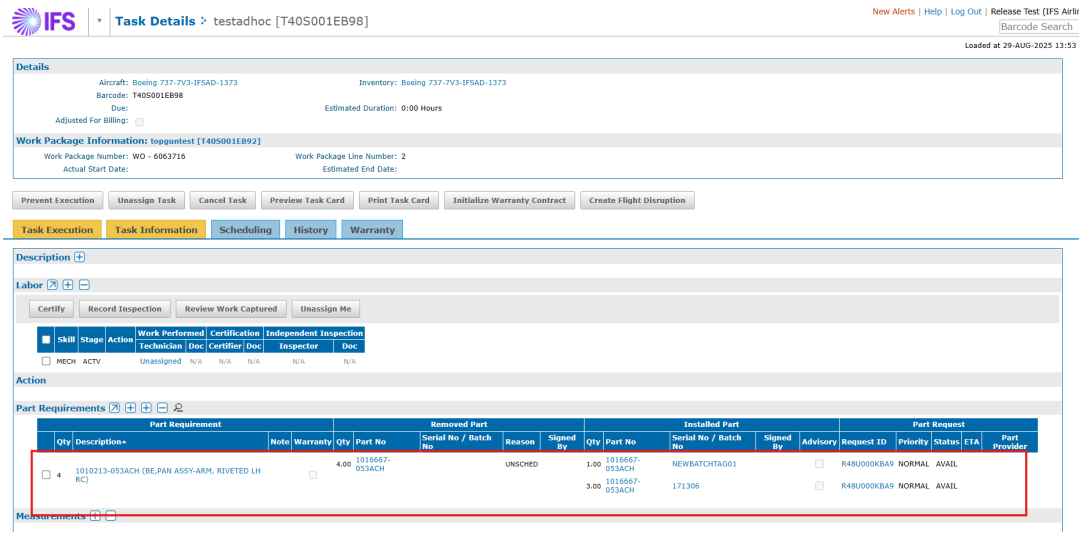
```
{
    "metadata": {
        "correlationId": "correlationId-123",
        "messageId": "MessageId-123",
        "system": "mxi",
        "timestamp": "2022-12-17T11:55:18.602835000Z",
        "user": "mxi"
    },
    "partRequest": {
        "id": {
```

```
            "masterReqId": "R48U000KBA9",
            "requestId": "R48U000KBA9"
        },
        "status": "PRAVAIL",
        "inventory": {
            "batch": {
                "part": {
                    "oemPartNumber": "1016667-053ACH",
                    "manufacturerCode": "53551"
                },
                "oemLotTag": "lottag01",
                "oemBatchTag": "newBatchTag01",
                "quantity": "1"
            }
        }
    }
}
```


Result : A part request is fulfilled without utilizing already reserved inventory.

Part requirement 1 should be unaltered and split batch is created for part requirement 2 with partial reservation.





Reduce Inventory Count on Unreserve

When batch inventory is unreserved in Maintenix, Maintenix can reserve it for other part request. In the Supply Chain API solution, unreserved batch inventory in Maintenix will not be eligible for reservation by other part requests. Therefore, the Supply Chain API will reduce the batch inventory count on unreserved inventory.

The following example details how unreserved batch inventory will not be made eligible.

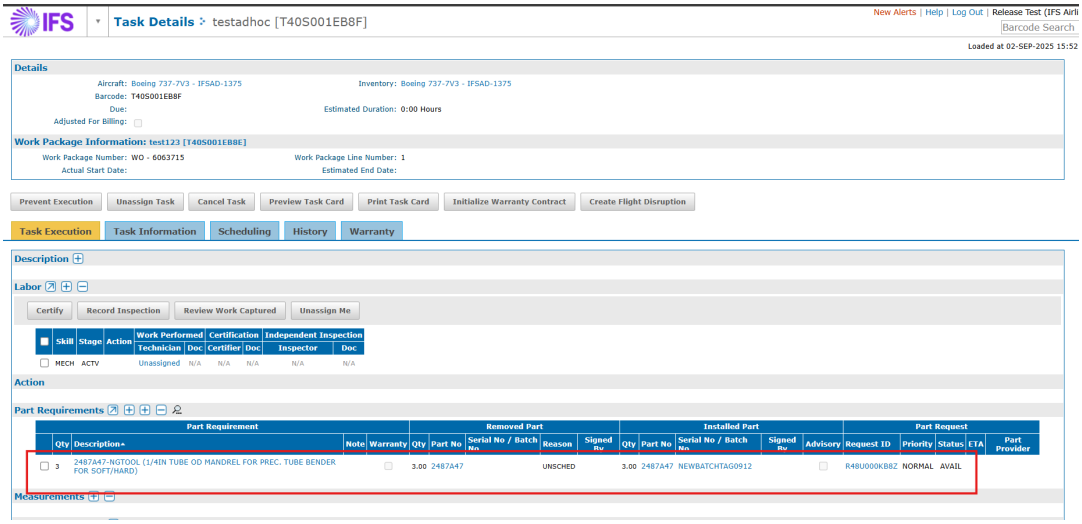
Goal: To reduce the batch inventory count on unreserved inventory.

Summary: Unreserved inventory will not be available for reservation.

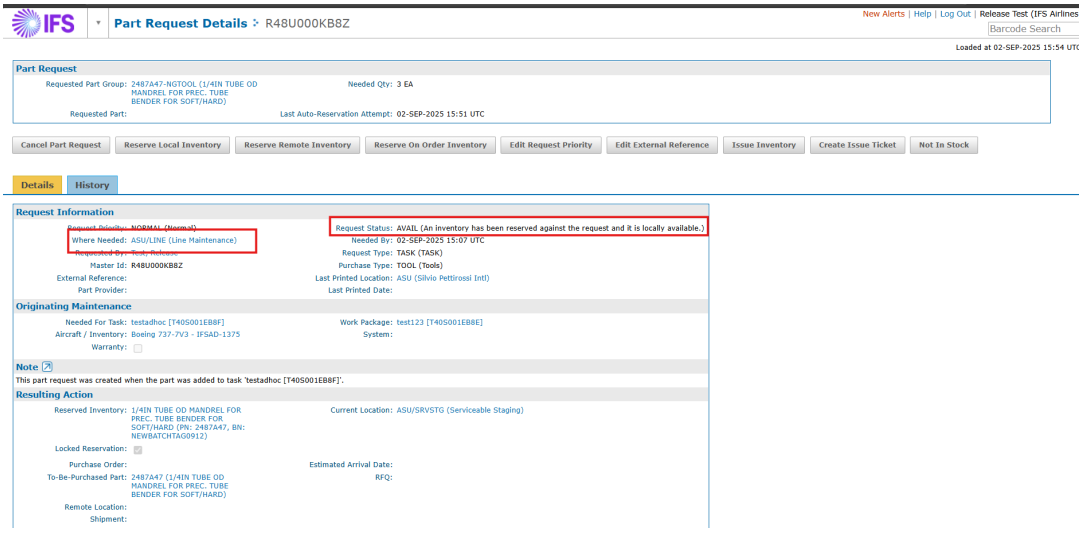
Pre-condition:

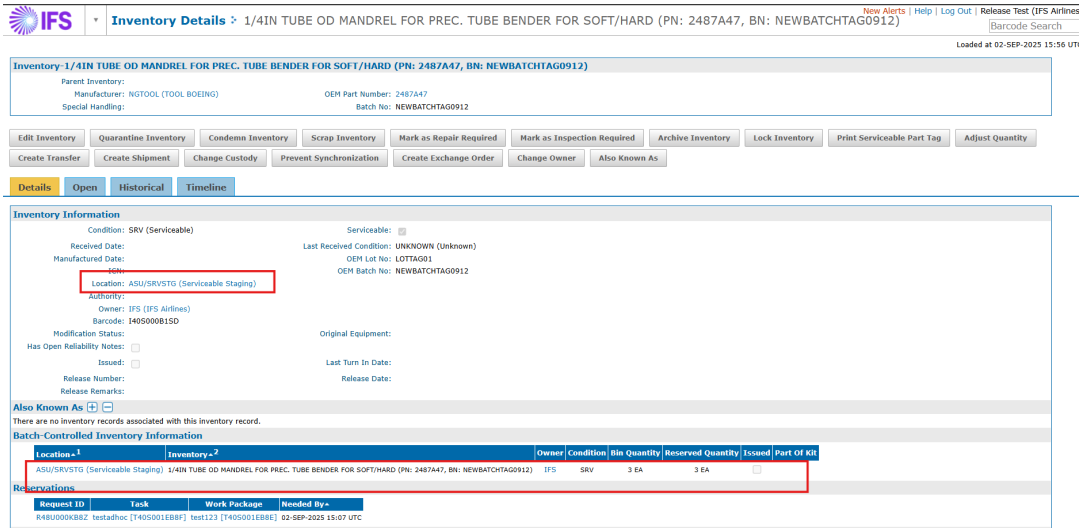
- **EXT_SC_BATCH_ENABLED** config parameter is enabled.

Step 1 : Create a workpackage -> task -> part requirement with a quantity of 3.

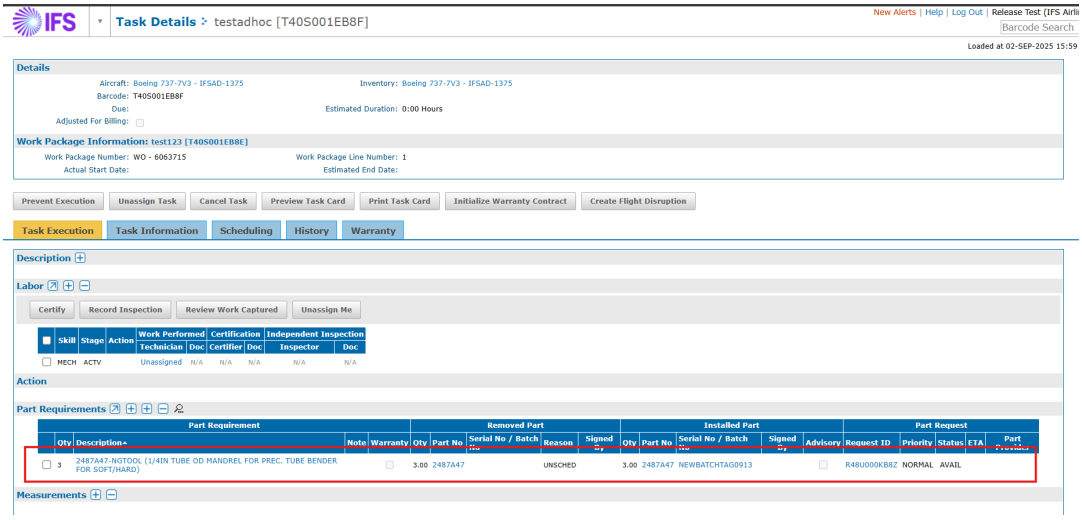


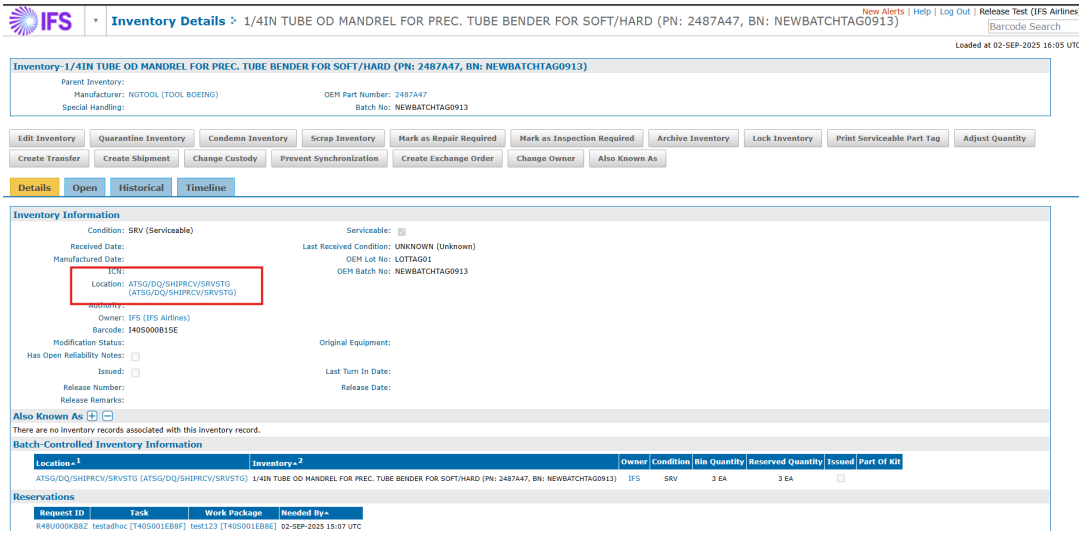
Step 2 : Schedule the workpackage under location A and reserve all 3 batch inventory for the part request.





Step 3 : Reschedule the workpackage to location B and reserve all batch inventory for the same part request.





Step 4 : Reschedule the workpackage back to location A and reserve only 2 (a lower quantity than the expected one) batch inventory for the same part request.

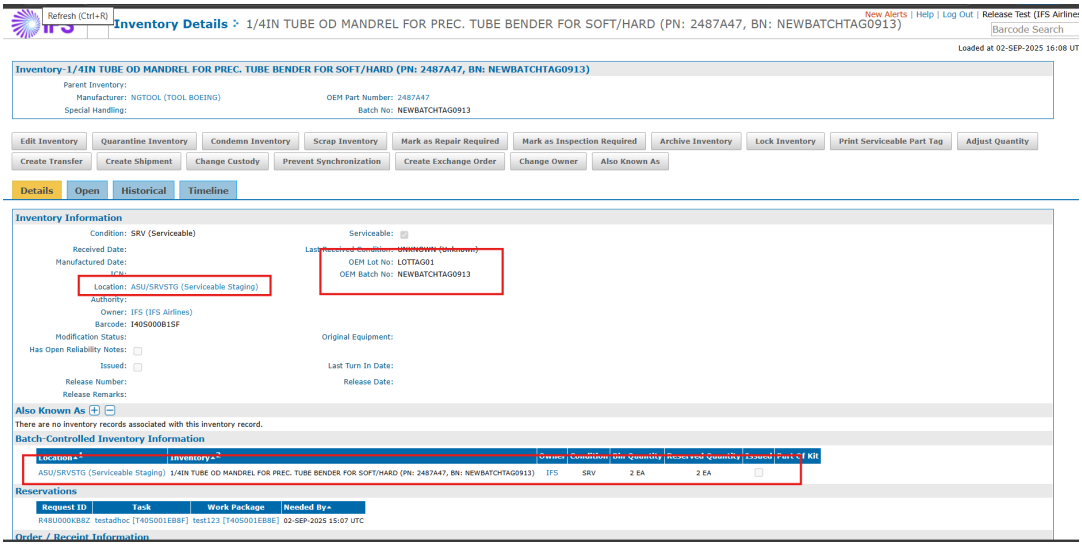
Send an **UpdatePartRequest** inbound message to reserve inventory for part requirement, but with lower quantity (quantity = 2) than the expected one.

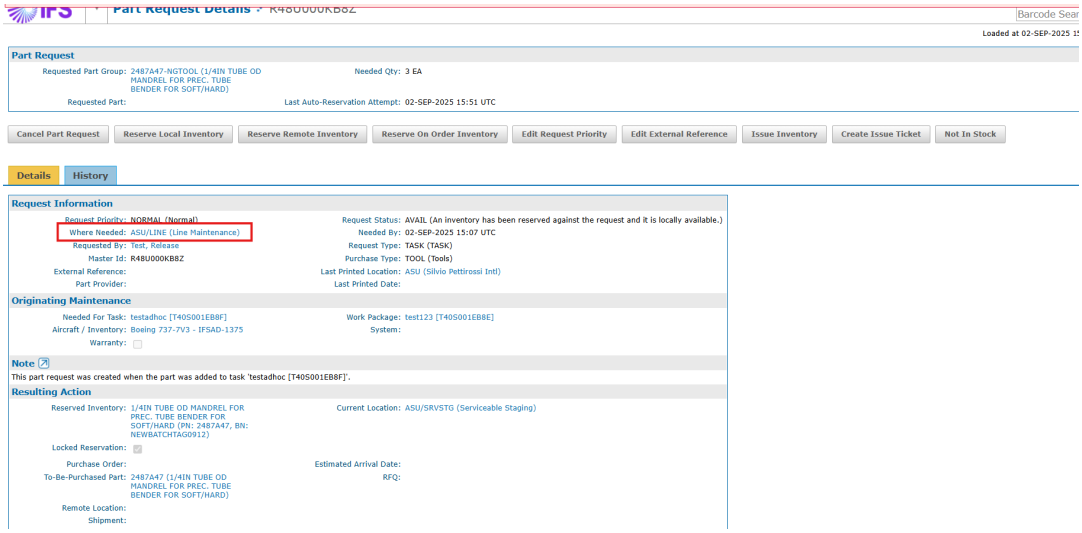
Messaging: An **UpdatePartRequest** message is sent from the External Supply Chain system to Maintenix.

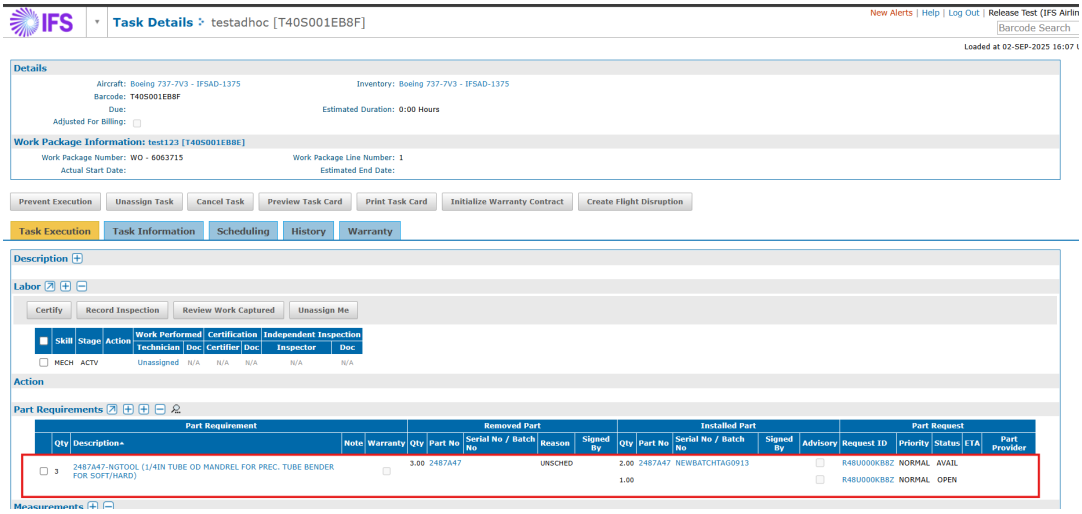
```
{
    "metadata": {
        "correlationId": "correlationId-123",
        "messageId": "MessageId-123",
        "system": "mxi",
        "timestamp": "2022-12-17T11:55:18.602835000Z",
        "user": "mxi"
    },
    "partRequest": {
        "id": {
            "masterReqId": "R48U000KB8Z",
            "requestId": "R48U000KB8Z"
        },
        "status": "PRAVAIL",
        "inventory": {
            "batch": {
                "part": {
                    "oemPartNumber": "2487A47",
                    "manufacturerCode": "NGTOOL"
                },
                "oemLotTag": "lottag01",
                "oemBatchTag": "newBatchTag0913",
                "quantity": "2"
            }
        }
    }
```

```
}
```

Result: The part request is fulfilled by utilizing reserved batch inventory. Split batch is created for the part request and all previous inventories are unavailable for reservation.







Cancel Part Request

Outbound Message - PartRequestCancelled

Goal: To extract and publish the **PartRequestCancelled** message to an external supply chain system.

Summary: Maintenix can notify the External Supply Chain system that a non-historical Part Request has been cancelled.

Pre conditions:

- Valid non-historic part request exists in Maintenix.

Messaging: Maintenix publishes a **PartRequestCancelled** outbound message once the trigger is enabled.

```
    "metadata": {
        "correlationId": "9a88fbda-6cac-45e2-a1bc-ec89928469bb ",
        "messageId": "MessageId-123",
        "system": "mxi",
        "timestamp": "2022-11-09T11:55:18.602835000Z",
        "user": "scapi"
    },
    "requestedParts": [

        {
            "masterReqId": "R48U000KB55",
            "requestId":   "R48U000KB55",
            "taskBarCode": "T40S001EB80"

        },

        {
```

```
            "masterReqId": "R48U000KB5W",
            "requestId":   "R48U000KB5W",
            "taskBarCode": "T40S001EB80"
        }

        ],

  "autorizationInfo": {

         "datetime": "2024-10-15T17:39:28.160Z",
         "authority": "mxi",
         "notes": "Canceled because task '36-13-03-01-JIC (36-13-03-01-
JIC) [T40S001EB80]' was canceled. "
         }
}
```

Result: Once a part request has been cancelled, Maintenix publishes **PartRequestCancelled** outbound message.

Inventory Item Issued

Inbound Message - IssueInventory

Notifying Maintenix that an inventory item was issued

Goal: To notify Maintenix that inventory was issued to fill a part request.

Summary: The Supply Chain System can notify Maintenix that the inventory was issued to fill a part request. This closes the part request.

It can take place when one of the following preconditions are met.

Pre conditions:

- API Permission **SC_API_INVENTORY_ISSUED** is enabled.
- Valid TRK/SER/ASSY part exists in Maintenix.
- Valid inventory exists in Maintenix.

Messaging: The external supply chain system sends an **IssueInventory** inbound message once inventory has been issued for the requested part.

```
    "metadata": {
        "correlationId": "{{correlation_id}}",
        "system": "External Supply Chain",
        "user": "mxintegration",
        "timestamp": "2024-11-20T17:50:10.282Z"
    },
    "notes": "{{inventory_type}} Inventory is Ready For Issue",
```

```
    "requestId": "{{request_id}}"
}
```

Result:

- The part request status is set to **ISSUED** as a result of which the boolean in the inventory will be updated accordingly. If provided, optional notes will be recorded in the history as well.
- The issued inventory can be viewed on the **Inventory Details** page.

Note The **IssueInventory** inbound message now supports KIT inventory.. KIT contents are currently not included in the **IssueInventory** message, Aside from that, it functions like any other part.

Inventory Turned In

Outbound Message - InventoryTurnedIn

Inventory Turned In

Goal: To return inventory back to the supply chain.

Summary: Inventory is returned to the supply chain so that it can be repaired and available for future use.

Pre conditions:

- API Permission **SC_API_INVENTORY_TURNED_IN** is enabled.
- PubSub is enabled.

Messaging: Maintenix publishes an **InventoryTurnedIn** outbound message.

```
{
    "metaData": {
        "correlationId": "b95b0d72-b833-4e4e-886c-9249b6b61662",
        "messageId": "e32edf96-41a5-4517-8742-d8cc90b178ea",
        "system": "MAINTENIX",
        "user": "mxintegration",
        "timestamp": "2024-12-20T16:03:26.617Z"
    },
```

```
"transferId": "X48U00025AS",
"partRequestId": {
    "masterReqId": "R0002AUL",
    "requestId": "R0002AUL"
},
"transferFromLocation": {
    "code": "NA/CA/ATL/NB/YFC/LINE",
    "name": "Line"
},
"inventory": {
    "inventoryConditionCode": "RFI",
    "ownerCode": "A319/320",
    "id": {
        "batch": {
            "part": {
                "oemPartNumber": "1_HCTABSY",
                "manufacturerCode": "MXI"
            },
            "oemLotTag": "YSLOT_42",
            "oemBatchTag": "YSBATCH_9",
            "quantity": 4
        }
    }
},
"authorizationInfo": {
    "datetime": "2024-12-20T16:03:17.447Z",
    "authority": "mxi"
}
```

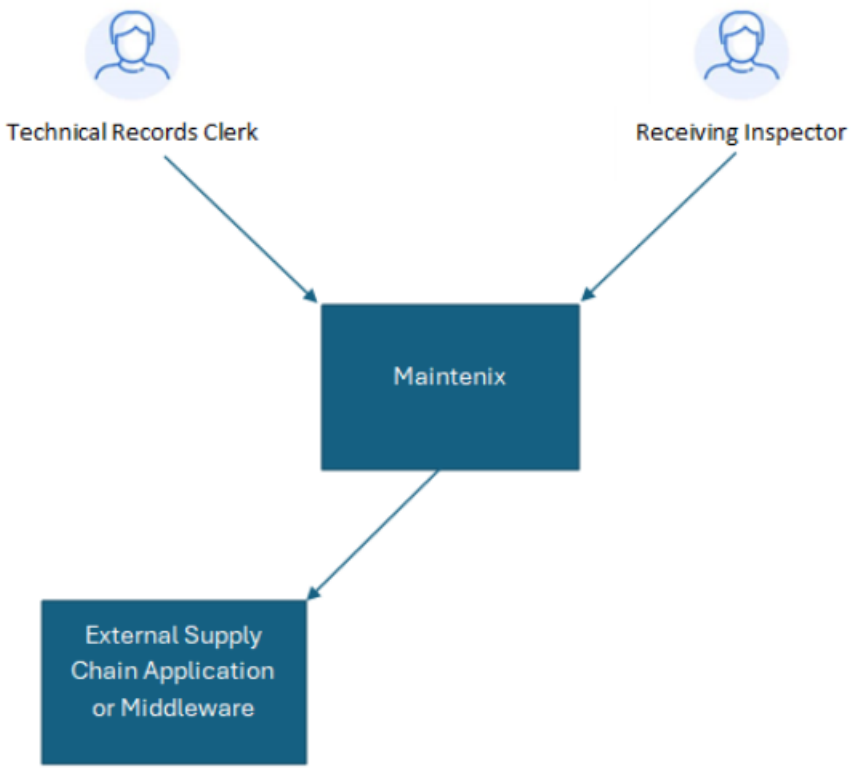
```
}
```

Result: Once Maintenix completes a TURN IN transfer, an **InventoryTurnedIn** outbound message is expected to be published.

5

Inventory Synchronization & Availability

The following diagram shows how the personas and systems interact in the external supply chain:



SC API Scenarios.

Each of the supported scenarios contain the following sections:

- **Goal:** A short statement that defines the purpose of the scenario.
- **Pre-conditions:** A list of conditions that must be met before the scenario can be implemented.

- **Messaging**: A basic description of the messaging required for the scenario.
- **Result:** The outcome of the successful execution of the scenario.

Create/Update Inventory

Inbound Message - CreateUpdateInventory

Create New Inventory

Goal: To create an inventory.

Summary: A new serial (TRK, SER or ASSY) is created in Maintenix

Pre conditions:

- API Permission **SC_API_CREATE_UPDATE_INVENTORY** is enabled..
- Valid TRK/SER/ASSY part exists in Maintenix.
- Valid location exists in Maintenix.
- Valid owner exists in Maintenix.

Messaging: The external supply chain system sends a **CreateUpdateInventory** inbound message.

```
{
    "metadata": {
        "correlationId": "{{correlation_id}}",
        "system": "External Supply Chain",
        "user": "mxintegration",
        "timestamp": "2024-09-20T17:50:10.282Z"
    },
    "action": "CREATE",
    "locationCode": "IFS",
    "ownerCode": "AIR",
    "inventoryId": {
       "part": {
            "oemPartNumber": "{{oemPartNumber}}",
            "manufacturerCode": "{{manufacturerCode}}"
        },
        "oemSerialNumber": "{{oemSerialNumber}}"
    },
    "ICN": "test"
}
```

Result:

- Once the **CreateUpdateInventory** inbound message is sent from the External Supply Chain system to Maintenix, the inventory will be created.

- The inventory created can be viewed on the **Inventory Details** page.

Update Inventory Location

Goal: Update the inventory location.

Summary: Location of the inventory can be updated within an airport and between the airports.

Pre conditions:

- API Permission **SC_API_CREATE_UPDATE_INVENTORY** is enabled.
- Either TRK/ SER/KIT/ASSY inventory should exist in Maintenix.
- Should be a loose inventory.
- Valid location exists in Maintenix.

Messaging: The inventory location can be updated using a natural key or unique ID.

1. Using Natural Key.

```
{
    "metadata": {
        "correlationId": "90b6f44b-9de8-4ae8-b86d-715b2af0a9d1",
        "system": "External Supply Chain",
        "user": "mxintegration",
        "timestamp": "2025-04-25"
    },
    "action": "LOCATION",
    "locationCode":"BAQ/LINE",
    "inventoryId": {
        "part":{
            "oemPartNumber":"820973-4",
            "manufacturerCode":"001JK"
        },
        "oemSerialNumber":"SERIAL TEST"
    }
}
```

```
2. Using Unique ID.
{
    "metadata": {
        "correlationId": "90b6f44b-9de8-4ae8-b86d-715b2af0a9d1",
        "system": "External Supply Chain",
        "user": "mxintegration",
        "timestamp": "2025-04-25"
    },
    "action": "LOCATION",
    "locationCode":"BSB/LINE",
    "inventoryId": {
        "uniqueId":"3738B3AC1E8211F0A406000D3AE89034"
    }
}
```

Result:

- The inventory location is updated.
- The **CreateUpdateInventory** inbound message is sent once the inventory location is updated.
- The updated inventory location can be viewed on the **Inventory Details** page.

Update Inventory Details

Goal: Update inventory properties: ICN, received date and manufactured date.

Summary: The inventory properties such as received date, ICN and manufactured date can be updated.

Pre-conditions:

- API Permission **SC_API_CREATE_UPDATE_INVENTORY** is enabled.
- Either TRK/ SER/ASSY inventory should exist in Maintenix.
- Should be a loose inventory.

```
{
```

```
    "metadata": {
        "correlationId": "90b6f44b-9de8-4ae8-b86d-715b2af0a9d1",
        "system": "External Supply Chain",
        "user": "mxintegration",
        "timestamp": "2025-04-21"
    },
    "action": "UPDATE",
    "inventoryId": {
        "part":{
            "oemPartNumber":"820973-4",
            "manufacturerCode":"001JK"
        },
        "oemSerialNumber":"SERIAL TEST"
    },
    "manufacturedDate":"2025-04-21",
    "receivedDate":"2025-04-23",
    "ICN":"icn test"
}
```

```
{
    "metadata": {
        "correlationId": "90b6f44b-9de8-4ae8-b86d-715b2af0a9d1",
        "system": "External Supply Chain",
        "user": "mxintegration",
        "timestamp": "2025-04-21"
```

```
    },
    "action": "UPDATE",
    "inventoryId": {
        "uniqueId":"3738B3AC1E8211F0A406000D3AE89034"
    },
    "manufacturedDate":"2025-04-22",
    "receivedDate":"2025-04-25",
    "ICN":"icn test two"
}
```

Result:

- Once the **CreateUpdateInventory** inbound message is sent from the external supply chain system to Maintenix, the details will be be updated.
- The updated details can be viewed on the **Inventory Details** page.

Update Inventory Condition

Goal: Update the inventory condition.

Summary: Inventory condition can be updated to RFI/ REPREQ/INSPREQ/ARCHIVE/ CONDEMN/QUAR/SCRAP.

Pre conditions:

- API Permission **SC_API_CREATE_UPDATE_INVENTORY** is enabled.
- Either TRK/SER/ASSY inventory should exist in Maintenix.
- Should be a loose inventory.

Messaging: The inventory condition can be updated using the unique ID or natural key.

```
1. Using unique ID.


{
    "metadata": {
        "correlationId": "90b6f44b-9de8-4ae8-b86d-715b2af0a9d1",
```

```
        "system": "External Supply Chain",
        "user": "mxintegration",
        "timestamp": "2025-04-21"
    },
    "action": "CONDITION",
    "conditionCode":"REPREQ",
    "inventoryId": {
        "uniqueId":"3738B3AC1E8211F0A406000D3AE89034"
    },
    "details":{
        "notes":"condition has been changed to REPREQ",
        "reason":"ACRDMG"
    }
}
```

2. Using natural key.

```
{
    "metadata": {
        "correlationId": "90b6f44b-9de8-4ae8-b86d-715b2af0a9d1",
        "system": "External Supply Chain",
        "user": "mxintegration",
        "timestamp": "2025-04-21"
    },
    "action": "CONDITION",
    "conditionCode":"ARCHIVE",
    "inventoryId": {
        "part":{
            "oemPartNumber":"820973-4",
```

```
            "manufacturerCode":"001JK"
        },
        "oemSerialNumber":"SERIAL TEST"
    },
    "details":{
        "notes":"condition has been changed to ARCHIVE",
        "reason":"ACRTV"
    }
}
```

Result:

- Once the **CreateUpdateInventory** inbound message is sent from the external supply chain system to maintenix, the inventory condition is updated.
- The updated conditions can be viewed on the **Inventory Details** page.

Update Inventory Ownership

Goal: Update the ownership of the inventory.

Summary: The owner of the Inventory can be changed.

Pre-conditions:

- API Permission **SC_API_CREATE_UPDATE_INVENTORY** is enabled.
- TRK/ SER/ASSY inventory should exist in Maintenix.
- Should be a loose inventory.
- Valid owner exists in Maintenix.

Messaging: The Inventory ownership can be updated using the unique ID or natural key.

```
1. Using unique ID.


{
    "metadata": {
        "correlationId": "90b6f44b-9de8-4ae8-b86d-715b2af0a9d1",
        "system": "External Supply Chain",
```

```
        "user": "mxintegration",
        "timestamp": "2025-04-26"
    },
    "action": "OWNERSHIP",
    "ownerCode":"CM",
    "inventoryId": {
        "uniqueId":"3738B3AC1E8211F0A406000D3AE89034"
    },
    "details": {
        "notes": "Inventory owner has been changed ",
        "reason": "ACSUP"

    }
}
```

2. Using natural key.

```
{
    "metadata": {
        "correlationId": "90b6f44b-9de8-4ae8-b86d-715b2af0a9d1",
        "system": "External Supply Chain",
        "user": "mxintegration",
        "timestamp": "2025-04-26"
    },
    "action": "OWNERSHIP",
    "ownerCode":"P5",
    "inventoryId": {
        "part":{
```

```
            "oemPartNumber":"820973-4",
            "manufacturerCode":"001JK"
        },
        "oemSerialNumber":"SERIAL TEST"
    },
    "details": {
        "notes": "Inventory owner has been changed ",
        "reason": "ACSUP"
    }
}
```

Result:

- Once the **CreateUpdateInventory** inbound message is sent from the external supply chain system to Maintenix, the inventory ownership is updated.
- The updated ownership can be viewed on the **Inventory Details** page.

Unarchive Existing Inventory

Goal: Unarchive the inventory.

Summary: Archived inventory within Maintenix will be reindicted once the message is sent.

Pre-conditions:

- Archived Inventory that exists in Maintenix with inventory class TRK, SER or ASSY.

Messaging: The external supply chain system sends a **CreateUpdateInventory** inbound message.to reindict existing inventory.

```
{
    "metadata": {
        "correlationId": "{{correlation_id}}",
        "system": "External Supply Chain",
        "user": "mxintegration",
        "timestamp": "2024-09-20T17:50:10.282Z"
    },
    "action": "CREATE",
    "locationCode": "GUA",
    "ownerCode": "IFS",
    "inventoryId": {
```

```
        "part": {
             "oemPartNumber": "{{oemPartNumber}}",
             "manufacturerCode": "{{manufacturerCode}}"
         },
         "oemSerialNumber": "{{oemSerialNumber}}"
    }
}
```

Result:

- Once the **CreateUpdateInventory** inbound message is sent, the relevant inventory will be unarchived.
- The reindicted inventory can be viewed on the **Inventory Details** page.

Update Serialized Inventory as KIT Content

Goal: Update the serialized inventory.

Summary:

Pre conditions:

- API Permission **SC_API_CREATE_UPDATE_INVENTORY** is enabled.
- Either TRK/SER/ASSY inventory should exist in Maintenix.

Messaging: The external supply chain system sends a **CreateUpdateInventory** inbound message.to update serialized inventory.

```
{
  "metadata": {
    "correlationId": "60.1.1234567",
    "messageId": "88888",
    "system": "MM",
    "user": "johnsmith",
    "timestamp": "2022-05-10T11:55:18.602835Z"
  },
  "action": "KIT",
  "inventoryId": {
    "oemSerialNumber": "SERPARTNo",
    "part": {
      "oemPartNumber": "SERPARTNO",
      "manufacturerCode": "BOEING"
    }
  },
   "kitInventoryId": {
    "oemSerialNumber": "KITINSERNO",
    "part": {
      "oemPartNumber": "KITPART",
      "manufacturerCode": "BOEING"
    }
  },
```

```
}
```

Result: Once the CreateUpdateInventory inbound message is sent, the relevant serialized inventory will be updated with the KIT Content.

Update Serialized Inventory as KIT Content: Error Case/Special Case

Goal:

Summary:

Pre-conditions:

Messaging:

Result:

Inventory Condition Updated

Outbound message - InventoryConditionUpdated

Inventory Condition Change

Goal: Notify the supply chain system when there is an inventory condition change.

Summary: The external supply chain is notified when there is a change in the inventory condition.

Pre condition:

- The inventory item must be in a detached state (not installed on a parent assembly or sub-assembly).
- The inventory must belong to one of the following inventory classes: Assembly (ASSY), Tracked (TRK), Serial-Controlled (SER).
- Condition changes must transition to one of the system defined states: RFI (Ready for Issue), INSPREQ (Inspection Required), REPREQ (Repair Required), ARCHIVE, SCRAP, QUAR (Quarantine), CONDEMN (Condemned), or INREP (In Repair).

Messaging: The external supply chain system sends an **InventoryConditionUpdated** outbound message once the condition change is made.

```
{
   "metaData": {
```

```
        "correlationId": "5d028618-c62c-47ff-b7a8-45658ad69a45",
        "messageId": "3dd6a61d-8ff2-439d-ba67-e49107414411",
        "system": "MAINTENIX",
        "user": "mxintegration",
        "timestamp": "2025-04-11T06:21:09.128Z"
    },
    "inventory": {
        "part": {
            "oemPartNumber": "1001246-3",
            "manufacturerCode": "73030"
        },
        "oemSerialNumber": "2012050022",
        "uniqueId": "87606CC4ABAB11E7BDEEA3C18A64A95E"
    },
    "conditionCode": "REPREQ",
    "eventDate": "2025-04-11T06:21:08.000Z",
    "authorizationInfo": {
        "datetime": "2025-04-11T06:21:08.000Z",
        "authority": "ADMIN",
        "reason": "MISPPWK",
        "notes": "Needs Paperwork!"
    }
}
```


Result:

- Once a condition change triggered by a user action takes place in the inventory, an outbound message is published in the external supply chain system.

- The customer is expected to receive a notification from Maintenix whenever a user action causes a condition change that meets the specified precondition.

Update Inventory Summary

Update Inventory Summary

Update Inventory Summary

Inbound Message - UpdateInventorySummary

Goal: To update the Maintenix front end to display Inventory Summary information based on the material availability information.

Summary: The Update Inventory Summary inbound message is sent by the Supply Chain system to IFS Maintenix to update the quantity and location of inventory in Maintenix.

When the **ENABLE_EXTSC_INVENTORY_SUMMARY** config param is enabled, the following Maintenix pages will display data from the Inventory Summary table instead of the core Maintenix inventory tables.

1. Part Availability tab in the Part Details page.
2. Availability tab in the Part Group page.
3. Parts Tab in the Work Package Details page.
4. Deferred Fault Part Request Page: **UTL_JOB with JOB_CD = MX_CORE_MVIEW_EXTERNAL_DEFER_FAULT_PART_REQ** is enabled.
5. Add Part Requirement Search page

Pre conditions:

- API Permission **SC_API_UPDATE_INVENTORY_SUMMARY** should be enabled.
- When an update inventory summary message is received and it contains one or more new part number, owner, location combination.

Messaging: The external supply chain sends an **UpdateInventorySummary** inbound message.

```
{
"metadata": {
"correlationId": "correlationId-PCR",
"messageId": "88888",
"system": "QANTAS",
"user": "173385",
"timestamp": "2025-02-04T11:55:18.602835Z"
```

```
},
"parts": [
{
"part": {
"oemPartNumber": "CFM56-7EMU33X",
"manufacturerCode": "58828"
},
"locationCode": "MEL",
"ownerCode": "SES",
"serviceableTransfersCount": 1,
"availableCount": 2,
"reservedCount": 2,
"issuedCount": 50,
"serviceableInKitCount": 1,
"unservCount": 15,
"quarantineCount": 2,
"awaitingInspectionCount": 10,
"inRepairCount": 10,
"condemnCount": 10,
"unserviceableTransfersCount": 10,
"unserviceableInKitCount": 10,
"onOrderCount": 10
},
{
"part": {
"oemPartNumber": "RP235-00",
"manufacturerCode": "00001"
```

```
},
"locationCode": "HKG",
"ownerCode": "QFA",
"serviceableTransfersCount": 10,
"availableCount": 20,
"reservedCount": 20,
"issuedCount": 20,
"serviceableInKitCount": 20,
"unservCount": 2,
"quarantineCount": 22,
"awaitingInspectionCount": 20,
"inRepairCount": 20,
"condemnCount": 20,
"unserviceableTransfersCount": 20,
"unserviceableInKitCount": 20,
"onOrderCount": 20
},
{
"part": {
"oemPartNumber": "TY2760-01A",
"manufacturerCode": "K0377"
},
"locationCode": "LAX",
"ownerCode": "QFA",
"serviceableTransfersCount": 10,
"availableCount": 20,
"reservedCount": 20,
"issuedCount": 20,
```

```
"serviceableInKitCount": 20,
"unservCount": 20,
"quarantineCount": 220,
"awaitingInspectionCount": 20,
"inRepairCount": 20,
"condemnCount": 20,
"unserviceableTransfersCount": 20,
"unserviceableInKitCount": 20,
"onOrderCount": 20
}
]
}
```

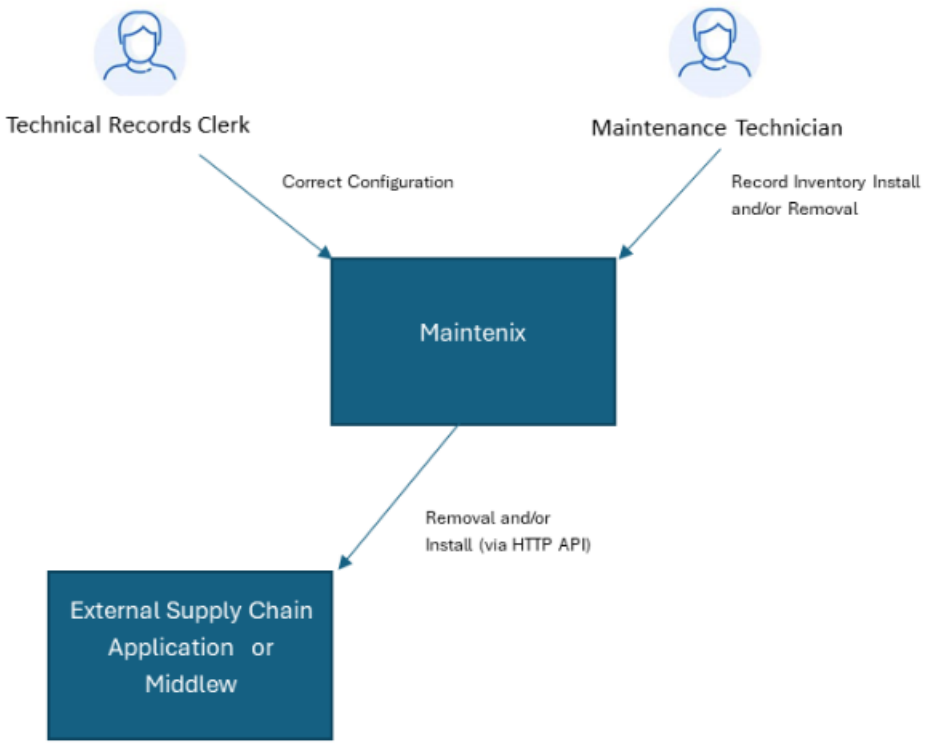

Result:

- Once the **UpdateInventorySummary** inbound message is sent, the quantity of inventory in Maintenix is updated in the Inventory summary table.
- Part Availability data will be displayed on the pages mentioned in the **Summary** section above.

6

Removals and Installs

The following diagram shows how the personas and systems interact in the external supply chain:



SC API Scenarios.

Each of the supported scenarios contain the following sections:

- **Goal:** A short statement that defines the purpose of the scenario.
- **Pre-conditions:** A list of conditions that must be met before the scenario can be implemented.
- **Messaging**: A basic description of the messaging required for the scenario.

- **Result:** The outcome of the successful execution of the scenario.

Pre-conditions and Result

- **MX_TS_PAUSE/ MX_TS_FINISHLABOUR/ MX_IN_ATTACH /MX_IN_DETACH/ MX_TS_COMPLETE** triggers are enabled.
- PubSub is enabled.
- An outbound destination for **InventoryRemovedInstalled** is configured in PubSub.
- A work package exists against an aircraft (not applicable to **MX_IN_ATTACH** and **MX_IN_DETACH**).
- The removed component must be in the inventory class of:
- Tracked (TRK), Serialized (SER), Assembly (ASSY) or Batch Repairable (BATCH with the Part Number Repairable flag enabled)
- The installed part must be in the inventory class of:
- Tracked (TRK), Serialized (SER), Assembly (ASSY) or Batch (BATCH).

Result: The message is sent to the destination configured in PubSub.

Component change - through labour pause

Goal:

- A component is to be removed from an aircraft (ACFT) after the labour row has been paused while executing a task (a job card, executable requirement, adhoc task or fault).
- A component is to be installed from an aircraft (ACFT) after the labour row has been paused while executing a task (a job card, executable requirement, adhoc task or fault).
- A component is to be removed from an off-wing complex component (subassembly, complex TRK) after the labour row has been paused while executing a task (a job card, executable requirement, adhoc task or fault).
- A component is to be installed from an off-wing complex component (subassembly, complex TRK) after the labour row has been paused while executing a task (a job card, executable requirement, adhoc task or fault).

Messaging: This sample message can take place during a labor pause or complete.


```
{
    "metaData": {
        "correlationId": "617df29a-1abc-49af-8ca9-30eac5181e8f",
        "messageId": "ff8bc74d-2d29-472b-8a02-96abfa283b07",
        "system": "MAINTENIX",
        "user": "mxintegration",
        "timestamp": "2025-03-10T16:11:03.528Z"
```

```
    },
    "rootInventory": {
        "inventoryId": {
            "part": {
                "oemPartNumber": "B767-232",
                "manufacturerCode": "1000"
            },
            "oemSerialNumber": "222",
            "uniqueId": "A00EE40D404B4CD1AED630860836A83D"
        },
        "aircraftRegistrationCode": "222"
    },
    "task": {
        "workorderLocationCode": "YYZ/LINE",
        "taskName": "TestAdhocTask06",
        "barcode": "T40S0001DPQ",
        "classCode": "ADHOC",
        "fault": false
    },
    "inventories": {
        "removedInventory": [{
                "inventoryDetails": {
                    "inventoryId": {
                        "part": {
                            "oemPartNumber": "398786-5",
                            "manufacturerCode": "1000"
                        },
                        "oemSerialNumber": "1861-60-609",
                        "uniqueId": "690E4B7AABAD4EC2AACEDF9C79FF8050"
                    },
                    "configSlotCode": "211001832",
                    "positionName": "1.1",
                    "positionId": 1,
                    "currentUsage": [{
                            "parameter": "H",
                            "tsn": 8967.2,
                            "tso": 8967.2,
                            "tsi": 8967.2
                        }, {
                            "parameter": "C",
                            "tsn": 678.0,
                            "tso": 678.0,
                            "tsi": 678.0
                        }
                    ],
                    "ownerCode": "A319/320"
                },
                "inventoryClassCode": "TRK",
                "inventoryConditionCode": "REPREQ",
                "removedQuantity": 1,
                "removedReasonCode": "IMSCHD",
                "autogeneratedWpBarcode": "T40S0001DPU",
                "turnInTransferId": "X48U00025BY"
            }
        ],
```

```
        "installedInventory": [{
                "inventoryDetails": {
                    "inventoryId": {
                        "part": {
                            "oemPartNumber": "398786-5",
                            "manufacturerCode": "1000"
                        },
                        "oemSerialNumber": "SN000293",
                        "uniqueId": "DC396F91FDC911EFB715DC4628AF39CA"
                    },
                    "configSlotCode": "211001832",
                    "positionName": "1.1",
                    "positionId": 1,
                    "currentUsage": [{
                            "parameter": "C",
                            "tsn": 0.0,
                            "tso": 0.0,
                            "tsi": 0.0
                        }, {
                            "parameter": "H",
                            "tsn": 0.0,
                            "tso": 0.0,
                            "tsi": 0.0
                        }
                    ],
                    "ownerCode": "A319/320"
                },
                "inventoryClassCode": "TRK",
                "installedQuantity": 1.0,
                "partRequestId": "R48U00025N0",
                "requestActionCode": "REQ"
            }
        ]
    },
    "authorizationInfo": {
        "datetime": "2025-03-10T16:11:03.338Z",
        "authority": "mxsystem"
    }
}
```


Component Removal - through labour row completion.

Goal:

- A component is to be removed from an aircraft (ACFT) after the labour row has been completed while executing a task (a job card, executable requirement, adhoc task or fault).
- A component is to be installed from an aircraft (ACFT) after the labour row has been completedwhile executing a task (a job card, executable requirement, adhoc task or fault).

- A component is to be removed from an off-wing complex component (subassembly, complex TRK) after the labour row has been completed while executing a task (a job card, executable requirement, adhoc task or fault).
- A component is to be installed from an off-wing complex component (subassembly, complex TRK) after the labour row has been completed while executing a task (a job card, executable requirement, adhoc task or fault).

Messaging: This sample message can take place during a labor pause or complete.


```
{
    "metaData": {
        "correlationId": "617df29a-1abc-49af-8ca9-30eac5181e8f",
        "messageId": "ff8bc74d-2d29-472b-8a02-96abfa283b07",
        "system": "MAINTENIX",
        "user": "mxintegration",
        "timestamp": "2025-03-10T16:11:03.528Z"
    },
    "rootInventory": {
        "inventoryId": {
            "part": {
                "oemPartNumber": "B767-232",
                "manufacturerCode": "1000"
            },
            "oemSerialNumber": "222",
            "uniqueId": "A00EE40D404B4CD1AED630860836A83D"
        },
        "aircraftRegistrationCode": "222"
    },
    "task": {
        "workorderLocationCode": "YYZ/LINE",
        "taskName": "TestAdhocTask06",
        "barcode": "T40S0001DPQ",
        "classCode": "ADHOC",
        "fault": false
    },
    "inventories": {
        "removedInventory": [{
                "inventoryDetails": {
                    "inventoryId": {
                        "part": {
                            "oemPartNumber": "398786-5",
                            "manufacturerCode": "1000"
                        },
                        "oemSerialNumber": "1861-60-609",
                        "uniqueId": "690E4B7AABAD4EC2AACEDF9C79FF8050"
                    },
                    "configSlotCode": "211001832",
                    "positionName": "1.1",
                    "positionId": 1,
                    "currentUsage": [{
                            "parameter": "H",
                            "tsn": 8967.2,
                            "tso": 8967.2,
```

```
                        "tsi": 8967.2
                    }, {
                        "parameter": "C",
                        "tsn": 678.0,
                        "tso": 678.0,
                        "tsi": 678.0
                    }
                ],
                "ownerCode": "A319/320"
            },
            "inventoryClassCode": "TRK",
            "inventoryConditionCode": "REPREQ",
            "removedQuantity": 1,
            "removedReasonCode": "IMSCHD",
            "autogeneratedWpBarcode": "T40S0001DPU",
            "turnInTransferId": "X48U00025BY"
        }
    ],
    "installedInventory": [{
            "inventoryDetails": {
                "inventoryId": {
                    "part": {
                        "oemPartNumber": "398786-5",
                        "manufacturerCode": "1000"
                    },
                    "oemSerialNumber": "SN000293",
                    "uniqueId": "DC396F91FDC911EFB715DC4628AF39CA"
                },
                "configSlotCode": "211001832",
                "positionName": "1.1",
                "positionId": 1,
                "currentUsage": [{
                        "parameter": "C",
                        "tsn": 0.0,
                        "tso": 0.0,
                        "tsi": 0.0
                    }, {
                        "parameter": "H",
                        "tsn": 0.0,
                        "tso": 0.0,
                        "tsi": 0.0
                    }
                ],
                "ownerCode": "A319/320"
            },
            "inventoryClassCode": "TRK",
            "installedQuantity": 1.0,
            "partRequestId": "R48U00025N0",
            "requestActionCode": "REQ"
        }
    ]
},
"authorizationInfo": {
    "datetime": "2025-03-10T16:11:03.338Z",
    "authority": "mxsystem"
```

```
    }
}
```

Component Install - through 'Inventory Attach' function.

Goal:

- A component is to be installed on an aircraft (ACFT) after completing an inventory attach transaction.
- A component is to be installed on an off-wing complex component (subassembly, complex TRK) after completing an inventory attach transaction.

Messaging: This message is used to attach when a subassembly is installed on an ACFT by inventory configuration change.

```
{
"metaData": {
"correlationId": "20486d46-bac5-49e2-b027-a88521de7716",
"messageId": "6d7bd8ce-8d8e-4f66-b1d8-f13b40578939",
"system": "MAINTENIX",
"user": "mxi",
"timestamp": "2024-09-19T16:35:11.559Z"
},
"rootInventory": {
"inventoryId": {
"part": {
"oemPartNumber": "B767-232",
"manufacturerCode": "1000"
},
"oemSerialNumber": "222"
},
"aircraftRegistrationCode": "222"
},
```

```
"inventories": {
"installedInventory": [{
"inventoryDetails": {
"inventoryId": {
"part": {
"oemPartNumber": "CF6-80A2",
"manufacturerCode": "2000"
},
"oemSerialNumber": "SN000283"
},
"configSlotCode": "CF6-80",
"positionName": "1.LT",
"positionId": 1,
"currentUsage": [{
"parameter": "C",
"tsn": 0.0,
"tso": 0.0,
"tsi": 0.0
}, {
"parameter": "H",
"tsn": 0.0,
"tso": 0.0,
"tsi": 0.0
}
]
},
"inventoryClassCode": "ASSY",
"installedQuantity": 1.0
```

```
}
]
}
}
```

Component Removal - through 'Inventory Detach' function.

Goal:

- A component is to be removed from an aircraft (ACFT) after completing an inventory detach transaction.
- A component is to be removed from an off-wing complex component (subassembly, complex TRK) after completing an inventory detach transaction.

Messaging: This message is used to detach when a subassembly is removed on ACFT by inventory configuration change.

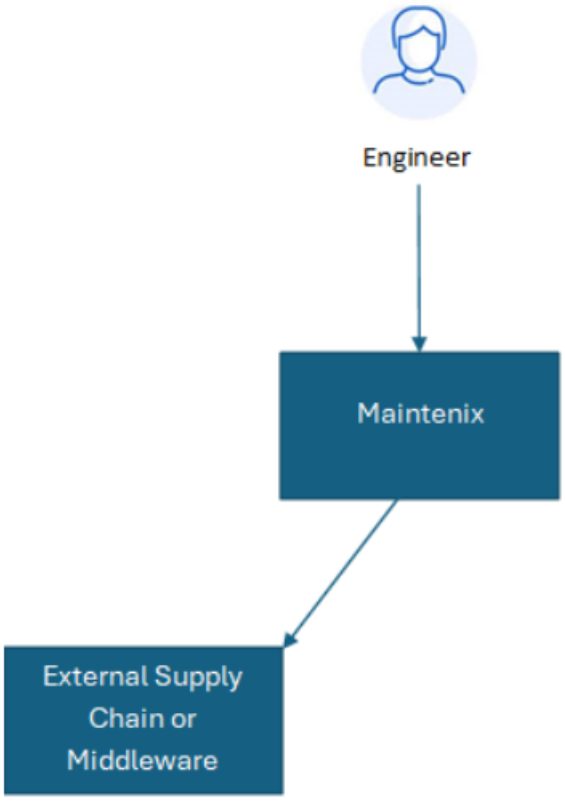
```
{
"metaData": {
"correlationId": "315ef5d4-f7cc-4581-98b3-251789733ea6",
"messageId": "161e4997-dfa9-42fe-8321-191829c8956f",
"system": "MAINTENIX",
"user": "mxi",
"timestamp": "2024-09-19T16:23:56.360Z"
},
"rootInventory": {
"inventoryId": {
"part": {
"oemPartNumber": "B767-232",
"manufacturerCode": "1000"
},
"oemSerialNumber": "222"
```

```
},
"aircraftRegistrationCode": "222"
},
"inventories": {
"removedInventory": [{
"inventoryDetails": {
"inventoryId": {
"part": {
"oemPartNumber": "CF6-80A2",
"manufacturerCode": "2000"
},
"oemSerialNumber": "SN000283"
},
"configSlotCode": "CF6-80",
"positionName": "1",
"positionId": 1,
"currentUsage": [{
"parameter": "C",
"tsn": 0.0,
"tso": 0.0,
"tsi": 0.0
}, {
"parameter": "H",
"tsn": 0.0,
"tso": 0.0,
"tsi": 0.0
}
]
```

7

Part Synchronization

The following diagram shows how the personas and systems interact in the external supply chain:



Material Adapter API Scenarios.

Each of the supported scenarios contain the following sections:

- **Goal:** A short statement that defines the purpose of the scenario.
- **Pre-conditions:** A list of conditions that must be met before the scenario can be implemented.
- **Messaging:** A basic description saying what is required for the scenario.

- **Result:** The outcome of the successful execution of the scenario.

Create New Part Number

Goal: To create a new part number in Maintenix.

Summary: When the external system is the master materials management system, the external system must create a new part number in Maintenix to synchronize Maintenix with the external system.

Pre-conditions: The specified part number does not exist in Maintenix.

Messaging: The external supply chain system sends an inbound ***create-part-definition-1.1*** message. Maintenix returns a ***part_definition_created-1.1*** outbound message.

- When you send a create-part-definition message, the OEM part number and manufacturer combination must not exist in Maintenix already.
- If the manufacturer provided in the message does not exist in Maintenix, Maintenix creates that manufacturer.

Result: The part number is created in Maintenix.

Approve Part Number

Goal: To approve a part number in Maintenix.

Summary: When the external system is the master materials management system, the external system must approve part numbers in Maintenix to synchronize Maintenix with the external system.

Pre-conditions: The specified part number exists in Maintenix.

Messaging: The external supply chain system sends an inbound ***activate_part_definition-1.0*** message. Maintenix returns an ***part_definition_activated-1.0*** outbound message.

- If the part number specified in the message exists in Maintenix in **BUILD** status, Maintenix approves the part and changes its status to **ACTV**.
- If the part number exists in Maintenix in **ACTV** status, then Maintenix ignores the message.
- You can also use the ***update-part-definition-status-1.1*** inbound message to approve a part number.
- You can configure the Part Activated alert to notify users when a part number has been approved.

Result: The part number is approved in Maintenix.

Obsolete Part Number

Goal: To reject a part number in Maintenix.

Summary: When the external system is the master materials management system, the external system must reject part numbers in Maintenix to synchronize Maintenix with the external system.

Pre-conditions: The specified part number exists in Maintenix.

Messaging: The external supply chain system sends an inbound ***update-part-definition-status-1.1*** message. Maintenix returns a ***part_definition_obsolesced-1.0*** outbound message.

Result: The part number is rejected in Maintenix.

Part Definition Created

Goal: To send a ***part_definition_created*** outbound message to the external supply chain system.

Summary: The message is sent as a notification service message or as a response to a create-part-definition inbound operational service message.

Pre-conditions:

- Set the ACTIVE_BOOL to 1 for the following trigger in Maintenix's UTL_TRIGGER table.

TRIGGER_ID	TRIGGER_CD	TRIGGER_NAME
99953	MX_MAT_PART_DEF_CREATED	Send Part Definition Created

Message Triggers:

The ***part_definition_created*** notification message is published in an external system when a new part definition is created in Maintenix.

Part Definition Activated

Goal: To send a ***part_definition_activated*** outbound message to the external supply chain system.

Summary: The ***part_definition_activated*** outbound message is sent to the external system as a notification service message, or as a response to a ***activate_part_definition-1.0*** or ***update-part-definitionstatus- 1.1*** inbound operational service message from an external system.

Pre-conditions:

Note There are no triggers to configure the ***part_definition_activated*** message. If the Materials API license exists, then Maintenix automatically triggers a message when a part number is approved.

Message triggers:

The ***part_definition_activated*** notification message is published to external systems when a part definition is approved in Maintenix.

Part Definition Obsolesced

Goal: To send a ***part_definition_obsolesced*** outbound message to the external supply chain system.

Summary: The ***part_definition_obsolesced*** message is sent to the external system as a notification service message, or as a response to a update-part-definition-status-1.1 inbound operational service message from an external system.

Pre-conditions:

Note There are no triggers to configure for the ***part_definition_obsolesced*** message. If the Materials API license exists, then Maintenix automatically triggers a message when a part number is rejected.

Message triggers:

The ***part_definition_obsolesced*** notification message is published to external systems when a new part definition is rejected, meaning that its status is changed to **OBSLT** in Maintenix.

Part Definition Updated

Goal: To send a ***part_definition_updated*** outbound message to the external supply chain system.

Summary: The ***part_definition_updated*** message is sent as a notification service message.

Pre-conditions:

Note There are no triggers to configure for the ***part_definition_updated*** message. If the Materials API license exists, then Maintenix automatically triggers a message when a part number is updated.

Message triggers:

The ***part_definition_updated*** notification message is sent to the external system when the part details are updated, or when a part's inventory class is transformed as follows:

- A part is promoted from batch controlled to serial controlled.
- A part is promoted from serial controlled to tracked.
- A part is demoted from tracked to serial controlled
- A part is demoted from serial controlled to batch controlled.

8 FAQs

1. What happens when a planner reschedules a work package from location A to location B?
   - The attribute 'Where Needed' on each part request will be updated to location B.
   - If scheduled dates have changed, the attribute 'Needed By' will also be updated.
   - If the part request status is equal to OPEN, there is no change to the status.
   - If the part request status is equal to AVAIL or REMOTE:
     - The status will change from AVAIL or REMOTE to OPEN
     - The outbound message PartRequestCreatedUpdated is sent from Maintenix to the external Supply Chain system.
     - From this point on, it is the standard reservation interaction between the 2 systems.
2. Can the external Supply Chain system send multiple batches to fulfill a single part request?
   - **Yes.** The inbound **UpdatePartRequest** message allows the Supply Chain system to reserve inventory in batches with a quantity greater than 1.
3. How are unused consumables handled?
   - Users can create an unexpected turn-in for unused consumables. Once the turn-in is completed, a **TurnedIn** outbound message is triggered by the Supply Chain API and sent from Maintenix to the external Supply Chain system. This message contains information about the returned unused consumable inventory.
4. How are removed components handled?
   - When components (Tracked, Serialized, Assembly or Batch Repairable) are removed from an aircraft or complex assembly, an **InventoryRemovedInstalled** outbound message is triggered by the Supply Chain API and sent from Maintenix to the external Supply Chain system. This message contains information about the removed component.
5. Are Kits supported?
   - **Yes, but with some limitations:**

- **Kit Content Baseline Definition:**
  - Define the Kit part number with **Inventory Class = KIT** in Maintenix.
  - Define Kit Contents, including part numbers and their respective quantities, within the Kit definition.
- The **PartRequestCreatedUpdated** outbound message supports KIT part numbers. Although a KIT includes multiple part numbers, these are not individually listed in the outbound message.
- The **UpdatePartRequest** inbound message supports updating/reserving KIT inventory. Kit contents are not currently included in this message, but KITs otherwise function like serialized parts.
- The **IssueInventory** inbound message now supports issuing KIT inventory.

6. Are ad-hoc part requests supported?
   - **Yes.** Ad-hoc part requests are supported by the Supply Chain API.
7. Are tools supported?
   - **No.** Tool requests made via the Tool Requirement section in the Maintenix GUI are **not** supported by the Supply Chain API.
8. What are the changes in Maintenix front end?
   - When the **ENABLE_EXTSC_INVENTORY_SUMMARY** config param is enabled, the following Maintenix pages will display data based on the inventory summary information of an external supply chain instead of the core Maintenix inventory tables.
     - Part Availability tab in the Part Details page.
     - Availability tab in the Part Group page.
     - Parts Tab in the Work Package Details page.
     - Deferred Fault Part Request Page.
     - Add Part Requirement Search page
     -
9. Will this API work for ICAM?
   - **Yes, limited to the technical upgrade.** The Supply Chain API can be installed and used in an ICAM environment to send and receive messages from an external Supply Chain system.

9
Appendix

Pubsub-subscriptions.properties sample

Sample pubsub-subscriptions.properties file for an HTTPS endpoint.

```
# HTTP subscription
#
# Channel: TEST1
# Application: fooApp
# Server: foo (defined in servers.properties or mx-servers.properties)
# Path: /fooPath (appended to the server url)
#subscription.TEST1.fooApp.serverId=foo
#subscription.TEST1.fooApp.path=/fooPath

# JMS subscription using basic authentication
#
# Channel: TEST2
# Application: bazApp
# Server: baz (defined in servers.properties or mx-servers.properties)
# Destination: com.mxi.mx.jms.someQueue (Queue or Topic)
#subscription.TEST2.bazApp.serverId=baz
#subscription.TEST2.bazApp.destination=com.mxi.mx.jms.someQueue
```

Supply Chain JMS Pubsub Subscriptions

subscription.MX_REMOVAL_INSTALL.scapiApp.serverId=scapi

subscription.MX_REMOVAL_INSTALL.scapiApp.path=/scapimock

subscription.MX_PART_REQUEST_CANCELLED.scapiApp.serverId=scapi

subscription.MX_PART_REQUEST_CANCELLED.scapiApp.path=/scapimock

subscription.MX_PART_REQUEST_CREATED_UPDATED.scapiApp.serverId=scapi

subscription.MX_PART_REQUEST_CREATED_UPDATED.scapiApp.path=/scapimock

subscription.MX_INVENTORY_TURNED_IN.scapiApp.serverId=scapi

subscription.MX_INVENTORY_TURNED_IN.scapiApp.path=/scapimock

subscription.MX_INVENTORY_CONDITION_UPDATED.scapiApp.serverId=scapi

subscription.MX_INVENTORY_CONDITION_UPDATED.scapiApp.path=/scapimock

Servers.properties sample

The URL and credentials for the External Supply Chain system (scapi in the example above) are configured in the servers.properties file. Below is a sample:

HTTP subscription using basic auth

#

Server: foo

Server Url: http://localhost:8080

Application: fooApp

Application Credentials: user1/password

#

#foo.url=http://localhost:8080

#foo.fooApp.username=user1

#foo.fooApp.password=${encrypted:password}

#

Supply Chain mock endpoint properties

#

scapi.url=https://45caf35f-156e-4ccf-b704-e76fdab5c1d0.mock.pstmn.io

scapi.scapiApp.username=mxi

scapi.scapiApp.password=password

Example of Authentication (IAM/OIDC) within IFS Cloud

The following example illustrates the configuration of Authentication (IAM/OIDC) for inbound message processing within the IFS Cloud environment. Postman is utilized as a simulation tool to emulate an external supply chain system sending the inbound message.

1. Follow the instructions in **Chapter 2: Setting Up External Supply Chain API**.

2. Setup Postman.

Create an environment in in Postman and provide sufficient details about the environment that you will test against.

- Click the **Environments** button on the far left of the screen.
- In the Environments column, click the plus (**+**) sign to create a new environment. Name the environment.
- In the new environment, create two new environment variables. The variable for the external server can be called **server** and the variable for the environment can be called **realm**.

> **Note** The name of the environment can vary depending on the internal URL.

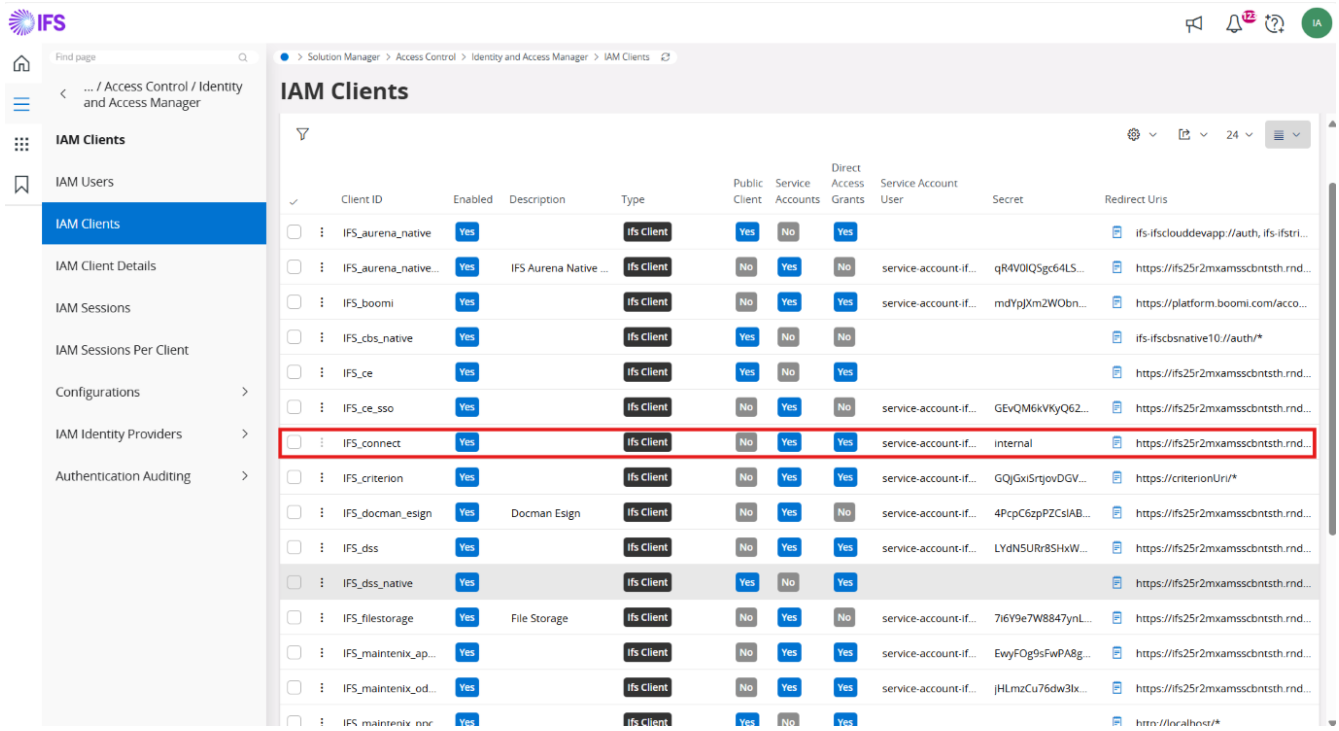
Create a new collection for your integration points.

- Select the **Collections** button on the far left of the screen. Then click the plus (**+**) sign in the middle pane.
- Navigate to the pane on the right side of the screen and select the **Authorization** tab. Change the authorization type to **OAuth 2.0**.
- Next, a new token should be configured. This can be done by selecting the **Edit Token** configuration button in the same tab.

Connections to IFS Cloud are done using the IAM a component/module in IFS Cloud. To configure **OAuth 2.0** the Client Secret (password) used by the **IFS_connect** users must be found.

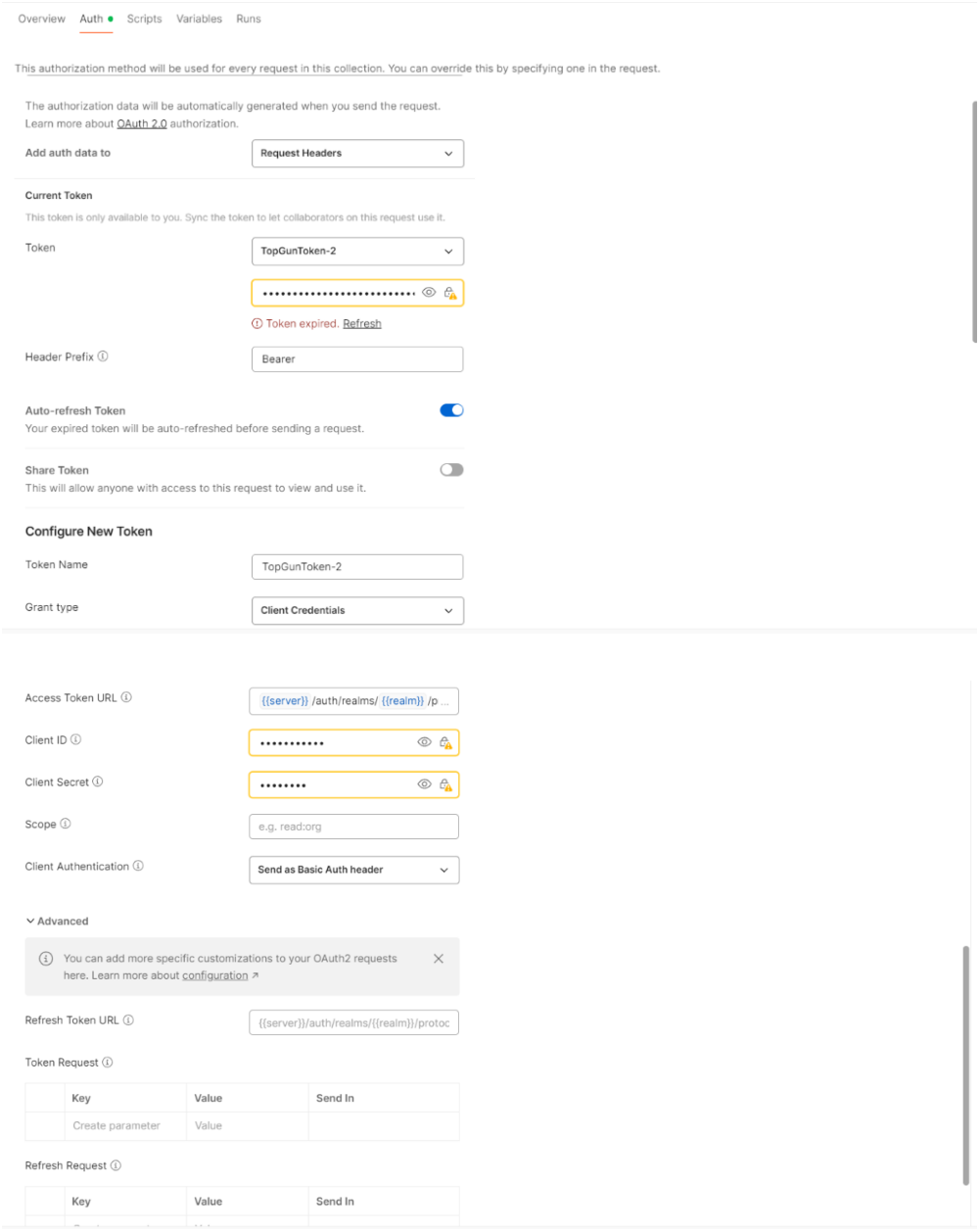
- In IFS Cloud navigate to **Solution Manager → Users and Permissions → Identity and Access Manager → IAM Client Details** and look for **IFS_connect.**

> **Note** Permission might be required to access this screen.



Next, return to the **Configure New Token** process on Postman.

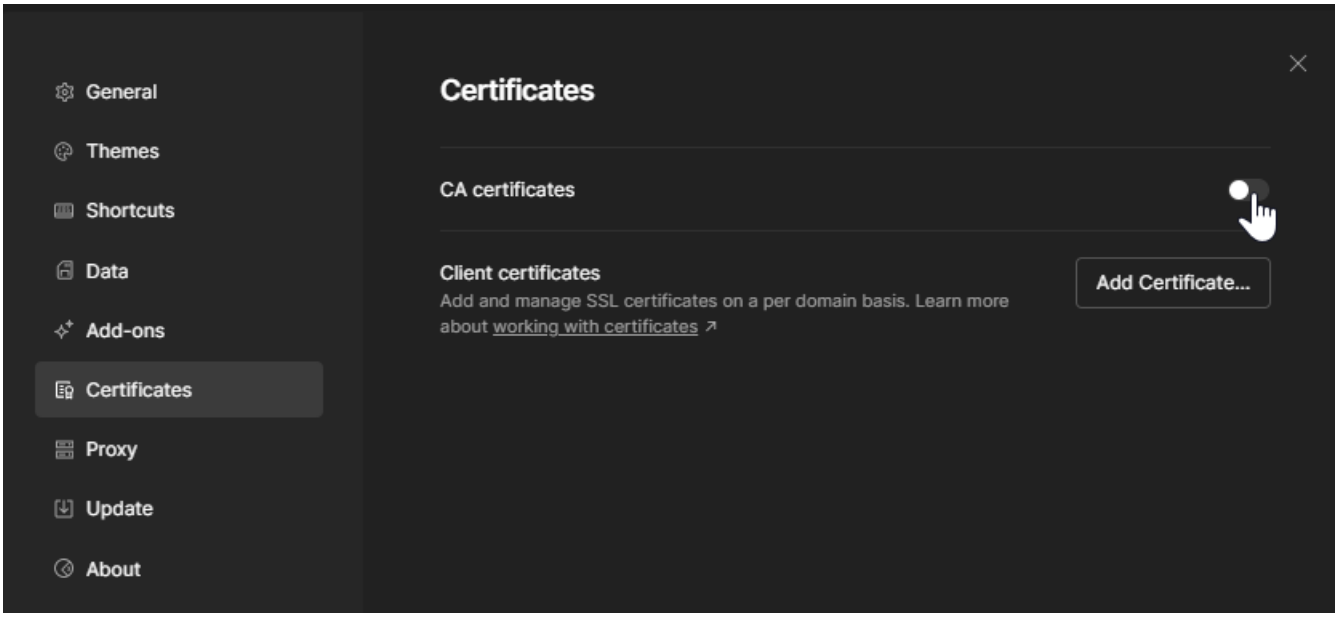
- Name the token. In the example below, it is named **SWA Token 1** (highlighted).
- Next enter the **Access Token URL**. The environment variables that were previously set up (server and realm) will be used here. The URL should look like ***{{server}}/auth/realms/{{realm}}/protocol/openid-connect/token***.
  - The **Client ID** is always set to **IFS_connect**.
  - The **Client Secret** (highlighted) is the password copied from the IFS IAM Client page.
  - The **Scope** should be set to **openid**.



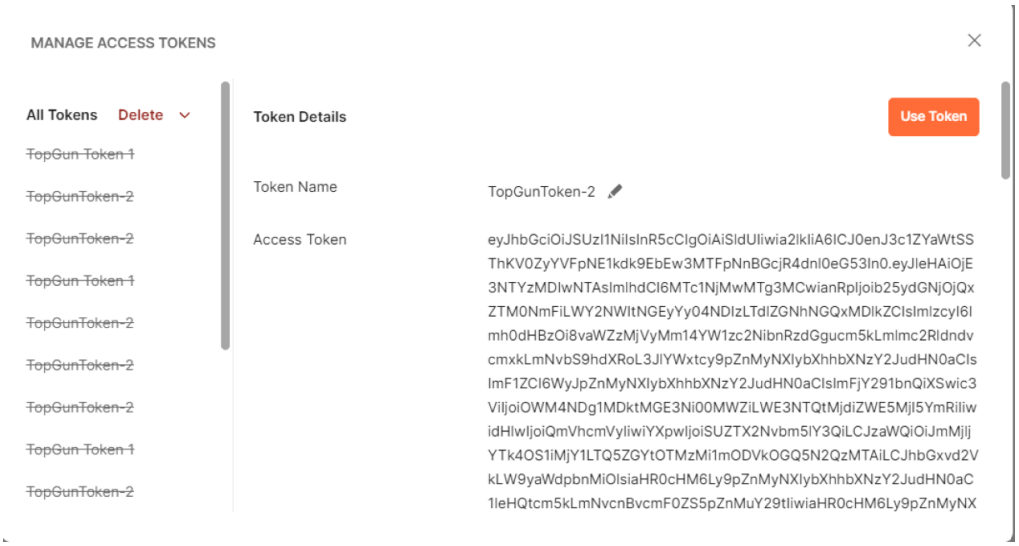
- Click on the **Get New Access Token** button.



- Once you click on the button, there is a possibility you could run into a certificate error (***Error: unable to verify the first certificate***). This error can be fixed by adding a cert to your Postman with the following steps:
  - Download the cert located here to your computer.
  - Open Postman and navigate to **Settings**.
  - Next, navigate to **Certificates** and turn on **CA certificates** by clicking on the toggle.

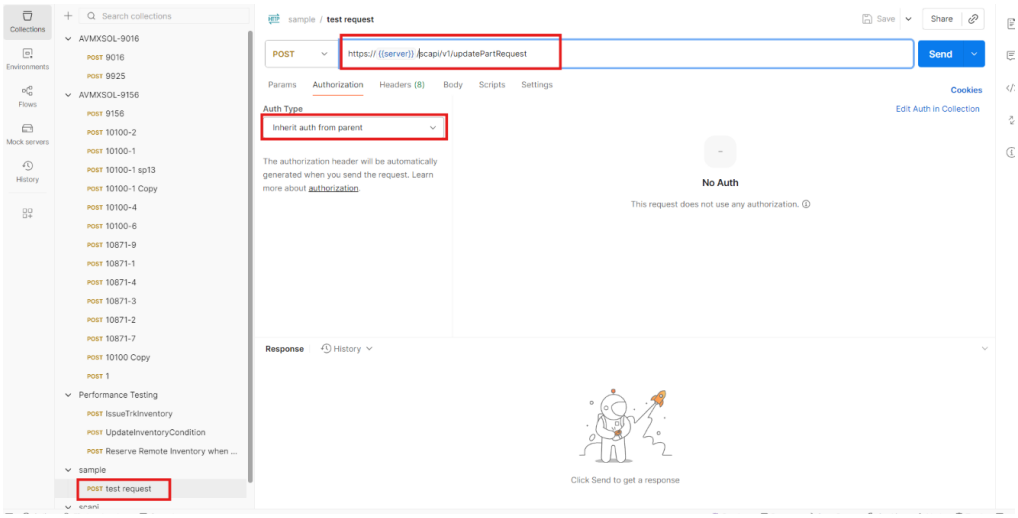


- A new field that allows you to add a cert will show up. Add the cert you downloaded by navigating to the location where you saved it on your computer (step 10 a). Once added, go back to your collection and click on the **Get New Access Token** button.
- Once executed successfully, you should be able to view the **Manage Access Tokens** pop-up (image below).
- This token is what **OAuth 2.0** will use to authenticate you when you make API calls. Click on the **Use Token** button.



- Since all of the steps above have been completed, an API request can now be created. For example (pictured below), a create flight message has been sent.
  - In the collections column select the **Add Request** link to create a new request.

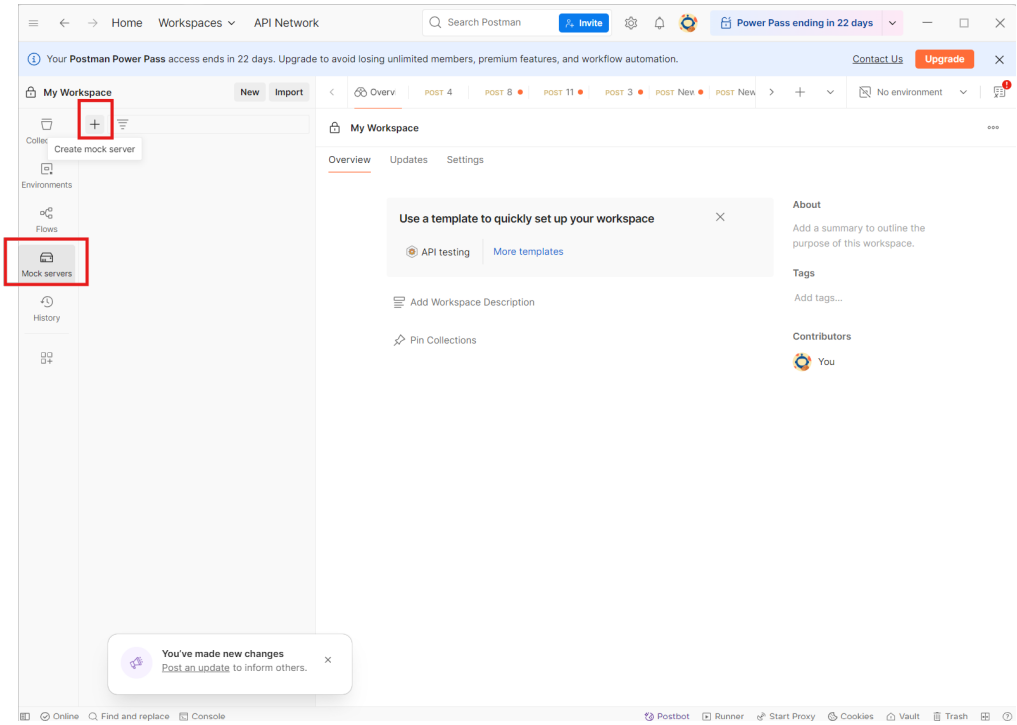
- Clicking on the **Authorization** tab should show that you have **inherit from parent** set. The URL displayed for the endpoint contains the "server" variable.



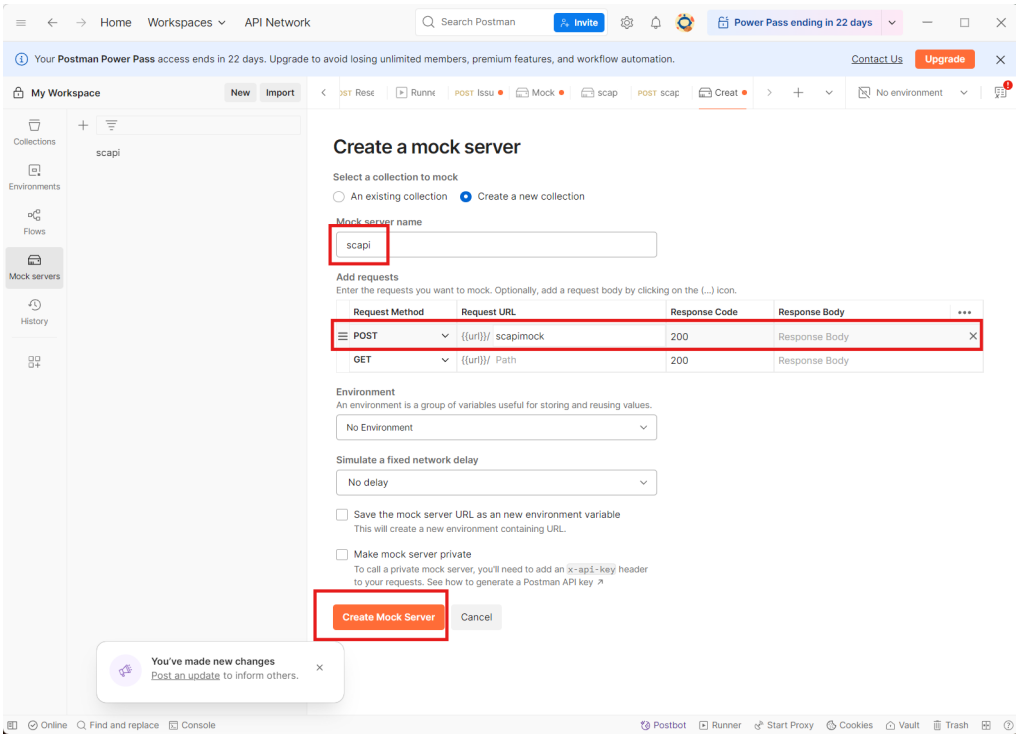
Example of PubSub Configuration Within IFS Cloud

The following example illustrates the configuration of PubSub for Outbound message processing within the IFS Cloud environment. Postman is utilized as a simulation tool to emulate an external supply chain system receiving the outbound message.

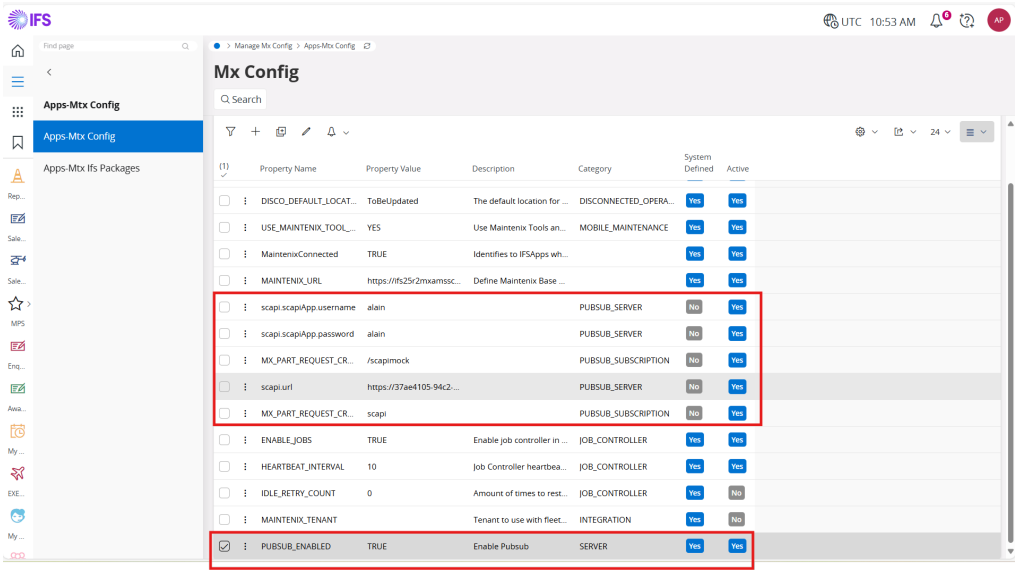
1. Follow the instructions in **Chapter 2: Setting Up External Supply Chain API**.
2. Create a new mock server using Postman.
   a. Open Postman app.
   b. Navigate to **`My Workspace > Mock Servers`**.Create mock server.



c. Once the mock server has been created, copy its URL.

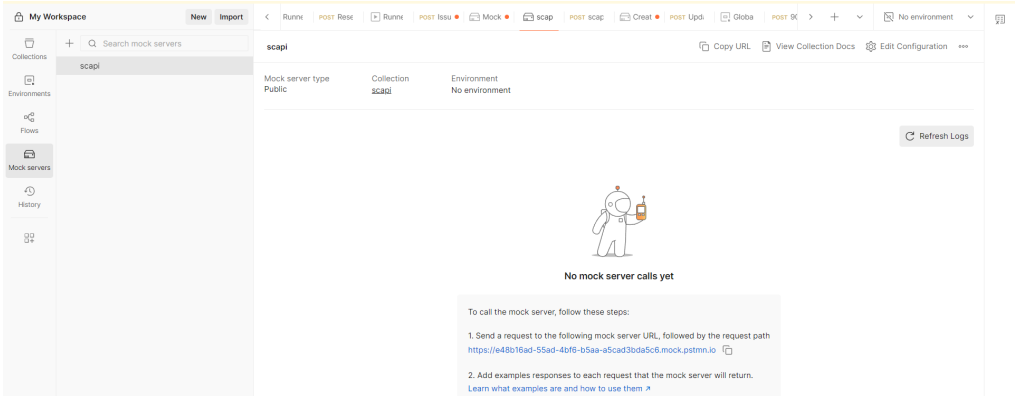


3. Navigate to **Manage Mx Config** in IFS Cloud and update as shown below.




```
scapi.url -> <Mock Server URL>
scapi.scapiApp.username -> <username of the IFS Cloud Web>
scapi.scapiApp.password -> <password of the IFS Cloud Web>
PUBSUB_ENABLED -> TRUE
----Add relavant serverId & path
MX_PART_REQUEST_CREATED_UPDATED.scapiApp.serverId -> scapi
MX_PART_REQUEST_CREATED_UPDATED.scapiApp.path -> /scapimock
```

4. Then, outbound message can be viewed in the mock server.



ABOUT IFS

IFS develops and delivers enterprise software for customers around the world who manufacture and distribute goods, maintain assets, and manage service-focused operations. The industry expertise of our people and solutions, together with commitment to our customers, has made us a recognized leader and the most recommended supplier in our sector. Our team of 3,500 employees supports more than 10,000 customers world-wide from a network of local offices and through our growing ecosystem of partners.

#forthechallengers

ifs.com

WHERE WE ARE

AMERICAS

+1 888 437 4968

ASIA PACIFIC

+65 63 33 33 00

EUROPE EAST

+48 22 577 45 00

EUROPE CENTRAL

+49 9131 77 340

UK & IRELAND

+44 1494 428 900

FRANCE, BENELUX AND IBERICA

+33 3 89 50 72 72

MIDDLE EAST AND AFRICA

+971 4390 0888

NORDICS

+46 13 460 4000

COPYRIGHT © 2026 INDUSTRIAL AND FINANCIAL SYSTEMS, IFS AB. IFS AND ALL IFS PRODUCTS AND SERVICES NAMES ARE TRADEMARKS OF IFS. ALL RIGHTS RESERVED. THIS DOCUMENT MAY CONTAIN STATEMENTS OF POSSIBLE FUTURE FUNCTIONALITY FOR IFS'S PRODUCTS AND TECHNOLOGY. SUCH STATEMENTS ARE FOR INFORMATION PURPOSES ONLY AND SHOULD NOT BE INTERPRETED AS ANY COMMITMENT OR REPRESENTATION. THE NAMES OF ACTUAL COMPANIES AND PRODUCTS MENTIONED HEREIN MAY BE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.