Project Maverick – Plan to Manufacture

Technical Lean Specification

Integration –Break Fix data from MAESTRO to Oracle Cloud - (PTM_INV_INT_146)

Document Control Information

Document Information

Document Identification	PTM_INV_INT_146
Document Name	Braek-Fix Data from MAESTRO to Oracle Cloud - (PTM_INV_INT_146)
Project Name	Project Maverick
Client	Dell
Document Author	Deloitte Project Maverick PTM Team
Document Version	V1.0
Document Status	Draft
Date Released	

Document Edit History

Version	Date	Additions/Modifications	Prepared/Revised by
1	04-14-2025	Initial version	Rajaramesh Betha
2	08-07-2025	Updated version	Abhijeet Desai
3	13-12-2025	Updated version	Abhijeet Desai

Document Review/Approval History

Date	Document Version	Name	Organization/Title	Comments
		Srivastava, Saumya	Deloitte (Manager)	

Distribution of Final Document

The following people are designated recipients of the final version of this document:

Name	Organization/Title

Table of Contents

1. Summary

1.1 Purpose/Justification

The purpose of the breakfix Interface from Maestro to Oracle Cloud is to streamline the inventory management process, ensuring that inventory levels are maintained efficiently and accurately between service orgs and Maestro. This interface facilitates the seamless transfer of break-fix shipment and return transactions between the Maestro system and Oracle Cloud, reducing manual intervention and minimizing the risk of errors. These service facilities will need to move materials between organizations due to demand. Maestro will be the application who analyzes the service part requests and orchestrates this material movement across the 3PL locations. The interface will come into play when there is a break-fix shipment or break-fix return transaction from Maestro. Oracle will need to communicate with Maestro to know these transactions and then update its own system as Oracle will be the source of all inventory management in the future state. By integrating these systems, Oracle can achieve real-time visibility into inventory levels, optimize stock replenishment cycles, and enhance overall supply chain efficiency.

Overall, the Interface from Maestro to Oracle Cloud represents a significant enhancement to the inventory management capabilities of an organization, providing a reliable and efficient means of maintaining optimal stock levels and supporting the smooth operation of supply chain processes.

1.2 Interface Description and Overview

The breakfix Interface from Maestro to Oracle Cloud is designed to establish a connection between these two applications regarding the break-fix process as well as inventory modifications. This interface ensures that inventory levels are maintained with seamless transfer of break-fix data and any other changes within inventory data from the Maestro system to Oracle Cloud for part change and sub-inventory transfer.

The interface operates by extracting relevant data from Maestro, such as shipment and return from DLP orgs to Float Orgs to Oracle. This data is then transformed into a format that is compatible with Oracle Cloud, ensuring that it can be accurately and efficiently processed by the receiving system.

Once the data is sent, Oracle Cloud will process the received data either into a break-fix shipment or break-fix return and update the received fields accordingly. This automated process reduces the need for manual intervention, minimizing the risk of errors and ensuring that inventory levels are consistently maintained.

Below systems will be involved in the overall process of the transmission -

1. Maestro - Dell owned system which will be involved in the initiation of services break-fix and service inventory adjustment processes
2. Kafka – the queuing system in the middleware to ensure efficient processing of bulk messages through proper allocation of resources
3. Oracle Integration Cloud Service (OIC) – the middleware integration layer system that will replace the current legacy Prism Eagle middleware system and will hold the transformation and orchestration logic to drive integration flows.

1.3 Business Rules

The Interface from Maestro to Oracle Cloud is designed to ensure seamless and efficient inventory management. The following business rules are essential for the successful implementation and operation of this interface:

1. Data Accuracy: All inventory data transferred from Maestro to Oracle Cloud must be accurate and up to date. This includes item quantities, descriptions, and locations. Any discrepancies must be flagged and resolved promptly to maintain data integrity. Maestro will also include data from HES inventory in the future state. There will need to be recognition of this type of inventory when Oracle is interfaced with Maestro.

2. Real-Time Synchronization: The interface must support real-time updates to reflect the current inventory status. This ensures that any changes in inventory levels, such as new stock arrivals or items issued, are immediately reflected in Oracle Cloud.

3. Error Handling: Robust error handling mechanisms must be in place to identify and address any issues during data transfer. This includes logging errors, notifying relevant personnel, and providing clear instructions for resolving the issues.

Assumptions:

S.no.	Assumption
1	Maestro will orchestrate material movement requests across 3PL locations
2	The messages from Maestro will come in the Kafka queue for the integration
3	There will be no direct communication between OIC and Maestro. All communication will be through Kafka
4	Any transaction processing within Maestro systems is out of scope and this integration will cover only the receiving and processing of messages from Maestro
5	Xsd message specs is not required as the payload is in JSON format.
6	UOM for any item is not included in the xml structure. It is assumed that the UOM is EA by default for all transactions
7	All prerequisite configs like inv orgs, sub-inventories, locations, Account Aliases and DFFs should be configured prior to invoking this interface

BRAZIL ADDITIONS

8	NF validation needs to be performed for both for the Shipment and Return transactions as part of this integration
9	FDG creation would have been created and approved before triggering the actual shipment message to Oracle.

10	FDG creation will happen through a separate integration and maestro will receive confirmation once the FDG is approved
11	FDC NF will be available in custom staging table before the Return trigger is processed
12	Despite a deffective part returning can have different price in BRL, according to DELL Costing Team, the transaction cost to be used in PAC is the inventory costing, not related to the cost from the Nota Fiscal.
13	Need to send the Ack/nack response back as part of this integration

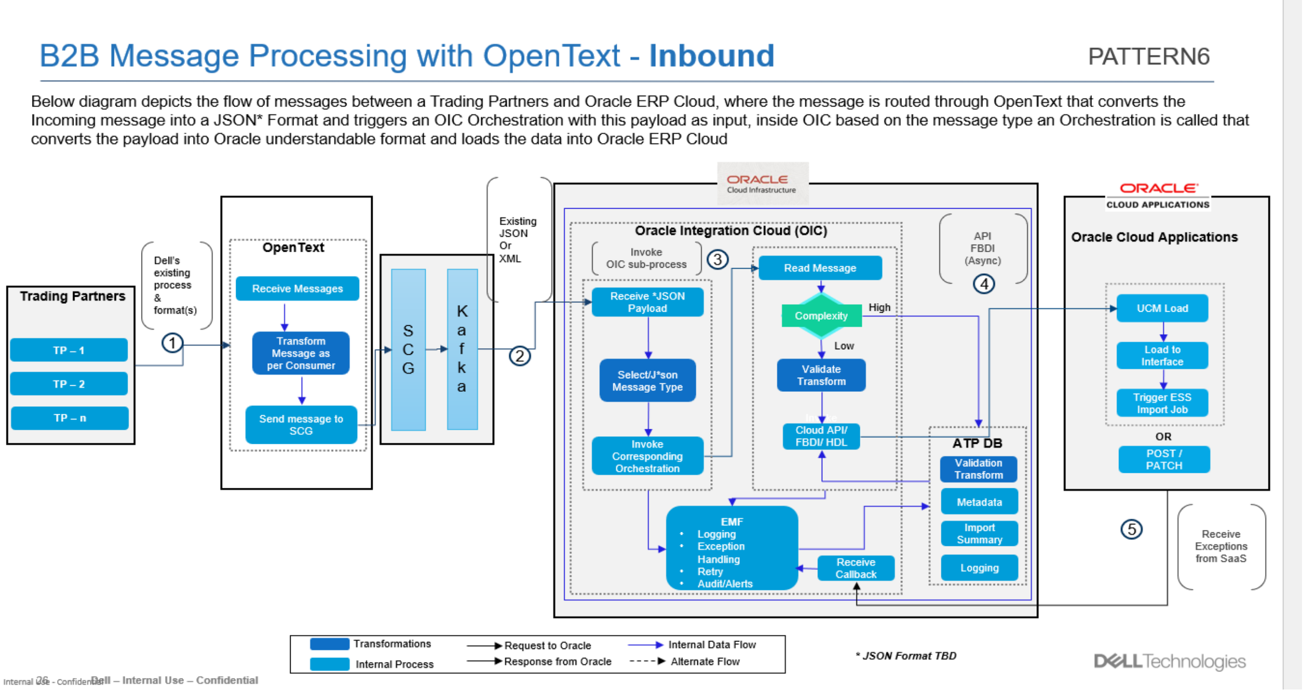
This interface will cover below user story IDs –

Integration	**User Story**
Braek-Fix Data from MAESTRO to Oracle Cloud	MAV-5435

1.4 Non-Functional Requirements

Encryption / Decryption - Data Security	No
Date/time format alignment on Payload	2024-12-10T09:00:02.000+00:00
Sequence of Processing	Kafka -> OIC -> Oracle Cloud ERP
Determination of if the interface will be trigger/Invoke or both	Realtime
Full Payload Processing or Partial Processing Allowed	Full Processing
Frequency - Real Time/ Near Real Time/ Batch	Real Time
Volume	80K Per day
Payload Size	TBD

Below will be the overall tech architecture process for inbound interfaces for Project Maverick -



B2B Message Processing with OpenText - **Inbound**

PATTERN6

Below diagram depicts the flow of messages between a Trading Partners and Oracle ERP Cloud, where the message is routed through OpenText that converts the Incoming message into a JSON* Format and triggers an OIC Orchestration with this payload as input, inside OIC based on the message type an Orchestration is called that converts the payload into Oracle understandable format and loads the data into Oracle ERP Cloud

* *JSON Format TBD*

Internal Use - Confidential 26 Dell – Internal Use – Confidential

1.5 Definitions and Acronyms

Acronym	Meaning	Description

ERP	Enterprise Resource Planning	A type of software used by organizations to manage business activities.
API	Application Programming Interface	A set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.
PaaS	Platform as a Service	A set of Cloud services that delivers infrastructure and middleware components in the cloud that enables developers and IT administrators to build and manage mobile apps and web applications.

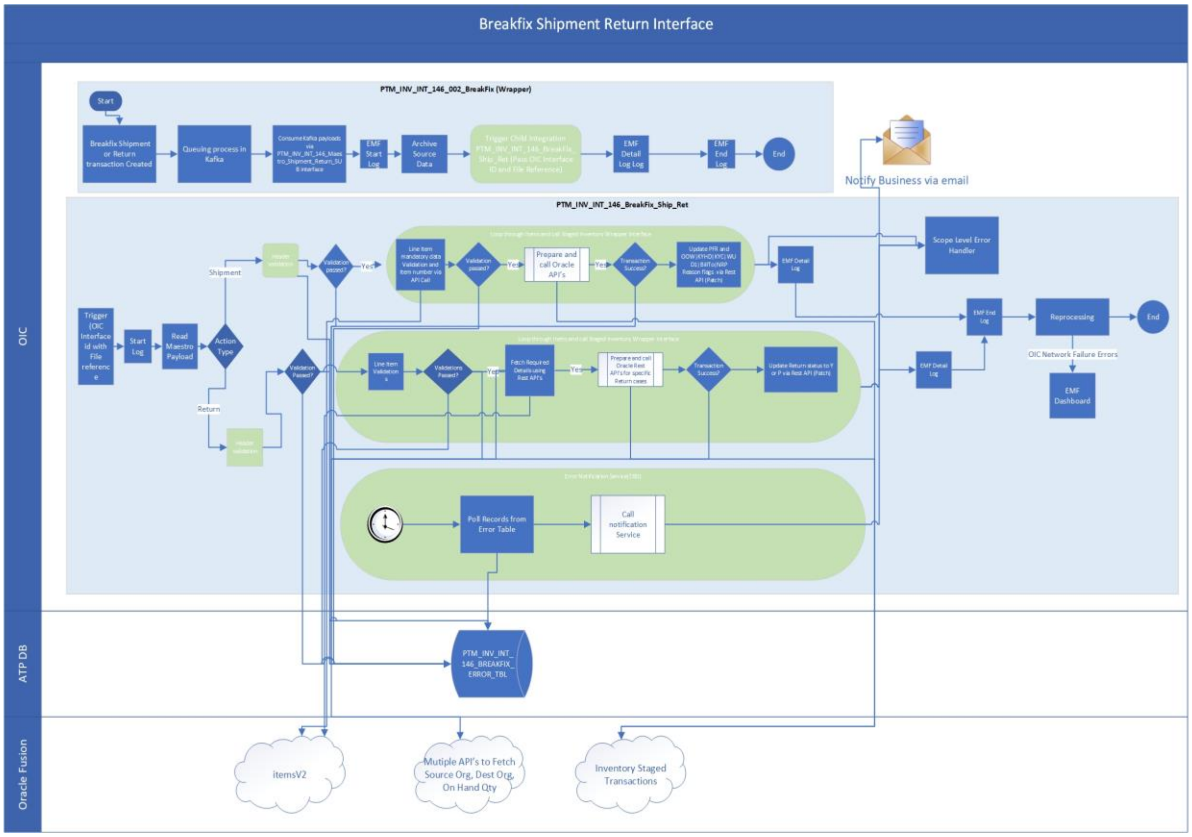
1.6 Sample Output

2. Technical Specification

2.1 Process Description

Breakfix Interface will sync the Shipment and Return transaction from Maestro to Oracle Cloud

2.2 Design Approach


Breakfix Shipment Return Interface
OIC
ATP DB
Oracle Fusion
PTM_INV_INT_146_002_BreakFix (Wrapper)
Start
Breakfix Shipment or Return transaction Created
Queuing process in Kafka
Consume Kafka payloads via PTM_INV_INT_146_Maestro_Shipment_Return_SUB interface
EMF Start Log
Archive Source Data
Trigger Child Integration PTM_INV_INT_146_BreakFix_Ship_Ret (Pass OIC Interface ID and File Reference)
EMF Detail Log Log
EMF End Log
End
Notify Business via email
PTM_INV_INT_146_BreakFix_Ship_Ret
Trigger (OIC Interface id with File reference)
Start Log
Read Maestro Payload
Action Type
Shipment
Return
Validation passed?
Yes
Line item mandatory data Validation and Item number via API Call
Validation passed?
Yes
Prepare and call Oracle API's
Yes
Transaction Success?
Update PFR and OOW|KYHD|KYC|WU D1|BillTo|NRP Reason flags via Rest API (Patch)
EMF Detail Log
Validation Passed?
Line item Validations
Validations Passed?
Yes
Fetch Required Details using Rest API's
Yes
Prepare and call Oracle Rest API's for specific Return cases
Transaction Success?
Update Return status to Y or P via Rest API (Patch)
EMF Detail Log
Scope Level Error Handler
EMF End Log
Reprocessing
End
OIC Network Failure Errors
EMF Dashboard
Poll Records from Error Table
Call notification Service
PTM_INV_INT_146_BREAKFIX_ERROR_TBL
itemsV2
Mutiple API's to Fetch Source Org, Dest Org, On Hand Qty
Inventory Staged Transactions

- Process starts with Breakfix shipment or Return Transaction creation at Maestro
- Message transformed to json format
- OIC process flow is initiated by consuming Json payloads from Kafka.

• Key Logic

Shipment

- The logic for shipment is that we move the inventory from DLP org to Float org.
- The destination Org Code is fetched from "Ship-to" location"
- Destination Subinventory is always "Usable"
- MessageTypeID would be "PartsStatusUpdateShipped"
- For break-fix, we need to create in-transit shipment. Once the in-transit shipment is created, it needs to be received at the destination immediately.

- We need to capture fields like Source Code, DPS Number, Transaction Reference is NPR Reason, Ship Date, Waybill Number
- Transaction date is defaulted to "sysdate"
- We also need to capture OOW, KYC, KYHD, WUD1 and Bill To flag values in Attribute 12
- For float management, we need to store the DPS number (coming from attribute 16 of transaction history DFF) as Lot number in the destination float org.
- For Float Management, we need to store Part Number, Float Org, DLP org and sub inventory, Quantity, Serial Numbers, Lot Number (DPS Number)
- If another transaction comes in with the same combination of Item and Lot details, The write-off needs to happen based on the earliest date.
- For HES Onsite Inv Orgs (Organization Type is HES and Facility type is Onsite). We would not be getting the serial numbers in the shipment payload. Hence, we need to pick the serial numbers which "Resides in stores" and has the earliest "Receipt Date"
- For HES Onsite Inv Orgs, we need to send the ACK Message with the Work Order Number (Dispatch ID), PPID Details, Satus as "Shipped" , Source Org and Sub-ineventory, Destination Org and Sub-inventory

BRAZIL ADDITIONS

- At the time of Shipment from DLP to Float Org, we need to perform the FDG validation based on the NF details received as part of the shipment trigger
- For Brazil, we need to perform **Account Alias Issue and Account Alias Receipt** and not the Intransit Shipment transaction like how it is happening now for Global process.
- Need to validate the Item and quantity in the FDG nota fiscal based on the transaction message received from Maestro
- The **Account Alias Issue and Account Alias Receipt** needs to be performed only if the FDG validation is successful, if not we should not proceed with the transaction and error message needs to be sent.
- For float management and reporting purposes, we need to store in a DFF the nota fiscal information (RA_CUSTOMER_TRX_ALL > CUSTOMER_TRX_ID) used to transfer items to the float organization.
- Remaining all DFF's which are being captured in Global process, should be captured AS-IS in addition to the NF value.
- Need to send the Ack/nack response back as part of this integration
- Need to Capture two descriptive flexfields while performing the transactions
  1. Fiscal Document Number in Nota Fiscal DFF – API Name "notaFiscal"
  2. Customer TRX ID into Customer TRX ID (Brazil) DFF – API Name "customerTrxIdBrazil".
- **Customer Trx Id** needs to be fetched in the below manner.
  - Identify the Nota Fiscal
  - Each NF will have a Transaction Number

  - Transaction Number for this kind of NF transactions are nothing but the AR Invoice Number
  - Need to identify the Customer Trx Id for that AR Invoice# and capture it in the DFF.

- **INDIA ADDITIONS**

  Need to Capture two descriptive flexfields while performing the transactions

-

  1. Invoice Number in Invoice Number DFF – API Name "invoiceNumber", Table Column "Attribute Number 1"
  2. Delivery Challan Number into Delivery Challan Number DFF – API Name "deliveryChallanNumber". Table Column "Attribute Number 3"

Break-fix return:

- The main columns we need to identify would be Action (which would be "returned"), DPS Number, Line Number, Item Shipped and Item received, Quantity received, Dest Org and Dest Subinventory. If Item shipped is Null, it need to be defaulted to Item received.
- MessageTypeID would be "PartsStatusUpdateReturned"
- We need to use LOT Number (which is Dispatch ID field) and item number combination to identify the Float Org and Subinventory.
- In All of the below scenarios, we need consider serial numbers if serial numbers are availble while doing the respective activity.
- If the serial numbers are different for both Shipment and Return transactions, then, first the existing Serial Numbers need to be issued out using Account Alias "Part Change" and then do Account Alias receipt with New Serial Numbers with Account Alias "Part Change"
- If the Serial Number which needs to be issued out is not mentioned, issue out the serial number in the FIFO format.

BRAZIL ADDITIONS

- At the time of Return from Float Org to DLP, we need to perform the FDC validation based on NF details received as part of the return trigger.
- Maestro will provide NF Access Key of the Nota Fiscal being used in the receipt transaction. This data will be used to find the Nota Fiscal captured by TR DFe and made available in staging tables.
- For Brazil, we need to perform **Account Alias Issue and Account Alias Receipt** and not the Intransit Shipment transaction like how it is happening now for Global process.
- Need to validate the Item and quantity in the FDC nota fiscal based on the transaction message received from Maestro (through NF Access Key)

- The FDC validation should happen based on the NF info available in the below custom staging tables
  STP_AP_NFE_HDR_STG_T
- STP_AP_NFE_LINE_STG_T
- Once the validation is successful, then the XML file available in the STP_AP_NFE_HDR_STG_T column "CLOB" needs to be pulled and placed in the UCM folder "scm/BrazilSEFAZSupplierMessages/Import".
- Once the file is loaded in UCM integration need to submit the ESS job "Run the Import Brazil Electronic Documents process"
- As part of next step, integration need to trigger Standard ESS Job ‘‘Import and Validate Electronic Fiscal Document’.
- Perform the Release Hold Step using the API

  fscmRestApi/resources/11.13.18.05/inboundFiscalDocumentHolds/{FiscalDocumentHoldId}

- FDC will use a Fiscal Flow of ‘Bookkeeping’
- Need to validate if the NF is in "Captured" status in FDC tables. Then only we need to proceed with the inventory transaction step
- The **Account Alias Issue and Account Alias Receipt** needs to be performed only if the FDC validation is successful, if not we should not proceed with the transaction and error message needs to be sent.
- For float management and reporting purposes, we need to store in a DFF the nota fiscal information (CMF_FISCAL_DOC_HEADERS > DOCUMENT_HEADER_ID) used to transfer items from the float organization.
- Remaining all DFF's which are being captured in Global process, should be captured AS-IS in addition to the NF value.
- Need to send the Ack/nack response back as part of this integration
- Need to Capture two descriptive flexfields while performing the transactions
  1. Fiscal Document Number in Nota Fiscal DFF – API Name "notaFiscal"
  2. Document Header Id (NF Document Header Id) into Customer TRX ID (Brazil) DFF – API Name "customerTrxIdBrazil".

- There are multiple cases for breakfix return and below are the scenarios:
- The Serial/Lot mismatch scenarios are listed here: **Break Fix New Scenarios.xlsx**

  Case1:

- When Item shipped is same as Item received in return payload and when the Lot number exists in the float org with no write-off, this is considered as a happy path scenario.
- In this case, we first need to identify whether the item and LOT combination that we received as part of return payload were also received as part of shipment. To check if they were received as part of shipment, we need to check the on-hand table for item and lot-combination. If the

combination exists, we need do the in-transit shipment and receipt of item from Float Org to DLP Org for the Item which is there in the Item Received field.
BRAZIL ADDITION
Instead of in-transit shipment and receipt of item from Float Org to DLP Org, we need to perform Account Alias Issue in Float Org and Account Alias Receipt in DLP Org.

Case2:

- When Item shipped is same as Item received in return payload and when the combination of item and LOT number doesn't exist in float org that were received as part of shipment payload, we need to check if it is a case of write-off or not. This means the combination of LOT and Item number doesn't exist in the on-hand table.
- If it is not write-off, this means that the LOT and item combination doesn't exist in on-hand table, because it can be of some other issue or consumption transaction but not a write-off transaction, we need to do Account Alias receipt at the DLP Org (shipment source org) by using the Item received value and the Alias would be "Additional Exchange Receipts". If the serial number used to perform AER account alias transaction is already existing in some other Lot, first issue out the serial number from that lot using AER account alias and then perform AER transaction at DLP org.
- While performing the account alias issue transaction, the work order number of return message should be populated in Work Order Number DFF for tracking
- If it is because of write-off that the LOT and item combination doesn't exist in on-hand table because of a write-off transaction. Then, first, we do Account Alias Receipt at the float org, and we use the same Alias that was used while writing off and we need to use item received for this transaction. Then, do an in-transit shipment from float org to DLP Org for the Item Received. If the serial numbers are different between shipment and return transactions, first, we need to issue out the older serial numbers using "Part Change" account Alias and bring in new serial numbers with "Part Change" account Alias receipt. We need to consider the serial number for this process.

Case3: Part Change

- If Item shipped and Item Received is not the same and lot and item combination exists in the float org for shipped item, and item shipped and transaction qty is less than or equal to open float qty, then the transaction is a part change quantity
- In this case, Account Alias issue need to be performed with the item shipped at the float org with the serial number that was shipped and the account alias is "Part Change" and we need to do Account Alias receipt with the Item Received at Float Org with the serial number and the account alias is "Part Change". These 2 should happen parallelly.
- The Second step for this case would be a in-transit shipment and receipt from float to DLP Org for Item Received.
- If the serial number that is brought in using Part Change account alias transaction already exists in some other lot, then issue out the serial number from that lot using AER account alias.
- While performing the account alias issue transaction, the work order number of return message should be populated in Work Order Number DFF for tracking
  BRAZIL ADDITION
  Instead of in-transit shipment and receipt of item from Float Org to DLP Org, we need to perform Account Alias Issue in Float Org and Account Alias Receipt in DLP Org for the item received

Case4: Part Change + WriteOff Reversal:

- If shipped item and received item are not the same on the return payload and if item shipped and Lot combination doesn't exist due to prior write-off

- In this case, first we need to Account Alias Receipt for Item shipped at the Float Org and the Alias and serial number should be same as the one used for Wite off
- The second step is to do Account Alias Issue for shipped Item and serial number at the float Org with Account Alias as “Part Change” and Account Alias Issue for Item Received at Float Org with Alias as Part Change. Both these transactions need to go in Parallel.
- The last step is to do a in-transit shipment and receipt from Float to DLP Org for the Item received.
- If the serial number that is brought in using Part Change account alias transaction already exists in some other lot, then issue out the serial number from that lot using AER account alias.
- While performing the account alias issue transaction, the work order number of return message should be populated in Work Order Number DFF for tracking

Case5: RNS

- If Breakfix return comes without a breakfix shipment (or) If breakfix return is sent post 180 days of Breakfix Shipment (NON PFR) or If Breakfix Return is sent post 360 days of Breakfix Shipment (PFR)
- In this case, Account Alias Receipt needs to happen at DLP Org for Item Received with Alias as “RNS”
- If the serial number with which RNS account alias transaction is to be performed already exists in some other lot, then issue out the serial number from that lot using RNS account alias and then do the account alias receipt as RNS at DLP org.
- While performing the account alias issue transaction, the work order number of return message should be populated in Work Order Number DFF for tracking~~.~~

HES Debrief :

For ISG related transactions, we may have 2 more message types coming in from Maestro.
The 2 Message types are:
1. PartsUsageUpdate
2. ServiceCallClosure

For both these Message types, we need to perform the account alias transactions below based on the combination of “Debrief Type”, “Disposit+++ion Type”, “Disposition Reason”.

If the serial number received in Debrief message for install exists in some other lot, then using the float org provided in the debrief message, find out the lot the serial number resides in and perform the transaction for INSTALL based on the below table. The work order number of the debrief message should be populated in the DFF
Latest Payloads for debrief transactions: **Debrief Latest Payloads**

All the transactions need to be performed on the float org of the shipment transaction and in the “USABLE” Sub-inventory.

S. No	Debrief Type	Disposition Type	Disposition Reason	Action to be performed	Transaction Type and Logic	DFFs
1	INSTALL	INSTALLED		INSTALL Transaction	Account alias issue from Float Org. Account Alias is "Field Service Usage"	**Source:** For transaction 1- l_work_order||'-'||l_record_id||'-'||'INSTALL' **Reference:** MAESTRO-PART_USAGE **Serial Number Attributes**: 1. Attribute1=ReturnWaybillNumber
2	REMOVE	LEFT ONSITE	PART RETENTION	REMOVE followed by Part Retention Issue	Account alias Receipt from Float Org. Account Alias is "Field Service Recovery" followed by Account Alias Issue with Alias name "Site Adjustment"	**Source:** For Transaction 1- l_work_order||'-'||l_record_id||'-'||'REMOVE' **Reference** MAESTRO-PART_USAGE For Transaction 2- **Source:** transaction_source_name = l_work_order||'-'||l_record_id||'-'||'PR_ISSUE' **Reference** MAESTRO-PART_USAGE

						Serial Number Attributes: 1. ReturnWaybillNumber for first transaction
3	REMOVE	LEFT ONSITE	DISK ERASURE	REMOVE Transaction	Account alias Receipt from Float Org. Account Alias is "Field Service Recovery"	**Source:** l_work_order||'-'||l_record_id||'-'||'REMOVE' **Reference** MAESTRO-PART_USAGE **Serial Number Attributes**: 1. ReturnWaybillNumber
4	REMOVE	RETURNED TO WAREHOUSE	DEFECTIVE	REMOVE Transaction	Account alias Receipt from Float Org. Account Alias is "Field Service Recovery"	**Source:** transaction_source_name = l_work_order||'-'||l_record_id||'-'||'REMOVE' **Reference** MAESTRO-PART_USAGE **Serial Number Attributes**: 1. ReturnWaybillNumber
5	REMOVE	RETURNED TO WAREHOUSE	CUSTOMER FA	REMOVE Transaction	Account alias Receipt from Float Org. Account Alias is "Field Service Recovery"	**Source:** l_work_order||'-'||l_record_id||'-'||'REMOVE'

						Reference MAESTRO-PART_USAGE **Serial Number Attributes**: 1. ReturnWaybillNumber 2. FA_FLAG = “Y”
6	REMOVE	RETURNED TO WAREHOUSE	MANDATORY FA	REMOVE Transaction	Account alias Receipt from Float Org. Account Alias is "Field Service Recovery"	**Source:** l_work_order||'-'||l_record_id||'-'||'REMOVE' **Reference** MAESTRO-PART_USAGE **Serial Number Attributes**: 1. ReturnWaybillNumber 2. FA_FLAG = “Y”
7	REMOVE	LEFT ONSITE FOR PICKUP	DISK ERASURE	REMOVE Transaction	Account alias Receipt from Float Org. Account Alias is "Field Service Recovery"	**Source:** l_work_order||'-'||l_record_id||'-'||'REMOVE' **Reference** MAESTRO-PART_USAGE **Serial Number Attributes**: 1. ReturnWaybillNumber
8	REMOVE	LEFT ONSITE	CUSTOMER FA	REMOVE Transaction	Account alias Receipt from Float Org. Account Alias is	**Source:**

		FOR PICKUP			"Field Service Recovery"	l_work_order||'-'||l_record_id||'-'||'REMOVE' **Reference** MAESTRO-PART_USAGE **Serial Number Attributes**: 1. ReturnWaybillNumber 2. FA_FLAG = “Y”
9	REMOVE	LEFT ONSITE FOR PICKUP	MANDATORY FA	REMOVE Transaction	Account alias Receipt from Float Org. Account Alias is "Field Service Recovery"	**Source:** l_work_order||'-'||l_record_id||'-'||'REMOVE' **Reference** MAESTRO-PART_USAGE **Serial Number Attributes**: 1. ReturnWaybillNumber 2. FA_FLAG = “Y”
10	REMOVE	RETURNED TO DELL LOGISTICS	DEFECTIVE	REMOVE Transaction	Account alias Receipt from Float Org. Account Alias is "Field Service Recovery"	**Source:** l_work_order||'-'||l_record_id||'-'||'REMOVE' **Reference** MAESTRO-PART_USAGE **Serial Number Attributes**: 1.

						ReturnWaybillNumber
11	REMOVE	RETURNED TO DELL LOGISTICS	DISK ERASURE	REMOVE Transaction	Account alias Receipt from Float Org. Account Alias is "Field Service Recovery"	**Source:** l_work_order||'-'||l_record_id||'-'||'REMOVE' **Reference** MAESTRO-PART_USAGE **Serial Number Attributes**: 1. ReturnWaybillNumber
12	REMOVE	RETURNED TO DELL LOGISTICS	CUSTOMER FA	REMOVE Transaction	Account alias Receipt from Float Org. Account Alias is "Field Service Recovery"	**Source:** l_work_order||'-'||l_record_id||'-'||'REMOVE' **Reference** MAESTRO-PART_USAGE **Serial Number Attributes**: 1. ReturnWaybillNumber 2. FA_FLAG = “Y”
13	REMOVE	RETURNED TO DELL LOGISTICS	MANDATORY FA	REMOVE Transaction	Account alias Receipt from Float Org. Account Alias is "Field Service Recovery"	**Source:** l_work_order||'-'||l_record_id||'-'||'REMOVE' **Serial Number Attributes**: 1. ReturnWaybillNumber 2. FA_FLAG = “Y”

14	DOA	RETURNED TO DELL LOGISTICS	DOA	INSTALL Transaction from Usable followed by REMOVE Transaction to Defective	Account alias issue from Float Org. Account Alias is "Field Service Usage" followed by Account alias Receipt from Float Org. Account Alias is "Field Service Recovery"	**Source:** For Transaction 1 - transaction_source_name = l_work_order||'-'||l_record_id||'-'||'INSTALL **Reference** MAESTRO-PART_USAGE For Transaction 2 - transaction_source_name = l_work_order||'-'||l_record_id||'-'||'REMOVE' **Reference** MAESTRO-PART_USAGE **Serial Number Attributes**: 1. ReturnWaybillNumber for both the transactions
15	UNUSED	RETURNED TO DELL LOGISTICS	UNUSED	UNUSED Transaction	No Transaction	
16	UNUSED	LOST	UNUSED	LOST Transaction	Account Alias Issue with Account Alias as "Lost Part"	**Source:** transaction_source_name = l_work_order||'-'||l_record_id||'-'||'LOST'

						Reference MAESTRO-PART_USAGE **Serial Number Attributes**: 1. ReturnWaybillNumber
17	DOA	RETURNED TO WAREHOUSE	DOA	INSTALL Transaction from Usable followed by REMOVE Transaction to Defective	Account alias issue from Float Org. Account Alias is "Field Service Usage" followed by Account alias Receipt from Float Org. Account Alias is "Field Service Recovery"	**Source:** **For Transaction 1 -** l_work_order||'-'||l_record_id||'-'||'INSTALL **Reference** MAESTRO-PART_USAGE **For Transaction 2 -** l_work_order||'-'||l_record_id||'-'||'REMOVE' **Serial Number Attributes**: 1. ReturnWaybillNumber for both the transactions
18	DOA	LEFT ONSITE	PART RETENTION	Part Retention Issue	Account Alias Issue with Alias name "Site Adjustment"	**Source:** transaction_source_name = l_work_order||'-'||l_record_id||'-'||'PR_ISSUE' **Reference** MAESTRO-PART_USAGE

19	UNUSED	LEFT ONSITE	RTR	Error: No transactions needed for UNUSED - RTR	No Transaction	
20	UNUSED	RETURNED TO WAREHOUSE	UNUSED	UNUSED Transaction	No Transaction	
21	UNUSED	LEFT ONSITE	PART RETENTION	Part Retention Issue	Account Alias Issue with Alias name "Site Adjustment"	**Source:** l_work_order||'-'||l_record_id||'-'||'PR_ISSUE' **Reference** MAESTRO-PART_USAGE
22	NO ACTION TAKEN			Error: No transactions needed for NO ACTION TAKEN	No Transaction	
23	UNUSED	LEFT ONSITE FOR PICKUP	UNUSED	UNUSED Transaction	No Transaction	

Case Summery :

Case No	Create Transaction				DFF Value 1	DFF Value 2	Both DFF	Route No	Source Code
	Alias Receipt (T1)	***Receipt Alias Name***	***Alias Issue (T2)***	***Issue Alias Name***					
1			Y	Field Service Usage	Y			23	INSTALL

2	Y (T1)	Field Service Recovery	Y (T2)	Site Adjustment	**Y (T1)**	NA		2	REMOVE (T1) and PR_ISSUE (T2)
3	Y	Field Service Recovery			Y			3	REMOVE
4	Y	Field Service Recovery			Y			4	REMOVE
5	Y	Field Service Recovery					Y	5	REMOVE
6	Y	Field Service Recovery					Y	6	REMOVE
7	Y	Field Service Recovery			Y			7	REMOVE
8	Y	Field Service Recovery					Y	8	REMOVE
9	Y	Field Service Recovery					Y	9	REMOVE
10	Y	Field Service Recovery			Y			1	REMOVE
11	Y	Field Service Recovery			Y			10	REMOVE
12	Y	Field Service Recovery					Y	11	REMOVE
13	Y	Field Service Recovery					Y	12	REMOVE
14	Y (T2)	Field Service Recovery	Y (T1)	Field Service Usage	**Y (T1 and T2)**			13	INSTALL
15	**No Transaction and no DFF update**							14	
16			Y	Lost Part	Y			15	LOST
17	Y (T2)	Field Service Recovery	Y (T1)	Field Service Usage	**Y (T1 and T2)**			16	INSTALL
18			Y	Site Adjustment	NA			17	PR_ISSUE
19	**No Transaction and no DFF update**							18	
20	**No Transaction and no DFF update**							19	
21			Y	Site Adjustment	NA			20	PR_ISSUE

22	**No Transaction and no DFF update**	21	
23	**No Transaction and no DFF update**	22	

2.3 Integration Design

Integration Design	
Integration System Name	Inventory Adjustment from Maestro to Oracle Cloud
Integration Type	Inbound
Integration Tool	OIC
Data Source	Maestro
Target	Oracle Cloud Inventory Management
Delivery / Retrieval Service	Kafka Stream
Output/Inbound Filename	NA
Pattern	Pattern 1
File Format	JSON
Expected Volume	Average Volume of the Inventory transactions
Run Frequency/Schedule	Real time sync with Legacy System, as we receive payloads from Maestro
Encryption/Decryption	NA
Payload Size	NA
Integration Notification	Email to business for failure transactions.

2.4 OIC Objects

2.4.1 OIC Integrations

OIC Integration Name	Description
PTM_INV_INT_146_002_BreakFix	Triggering kafka topic to read file and call child integration.
PTM_INV_INT_146_002_BreakFix_Ship_Ret	Manage Shipment and Return details related to breakFix cases(child).

2.4.2 OIC Lookups

OIC Lookup Name	Description
PTM_INV_INT_146_EMAIL_LOOKUP	Send email notification using this lookup.

Key	Value
Integration_name	to_email , from_email

2.4.3 OIC Connections

OIC Connection Name	Connection Type	Description
PTM_SDM_KAFKA_Conn	Apache Kafka Adapter	It is used to trigger/integrate PTM and SDM systems via streaming data.
PTM_OIC_REST_Conn	Rest Adapter	Used for real time rest API calls.
PTM_OIC_OBJ_STORAGE_Conn	Oracle Object Storage	Connection to oracle object storage for storing or retrieving files, used for file-based integration.
PTM_OIC_OAUTH_REST _TRIGGER_Conn	Rest Adapter (OAuth Trigger)	Rest trigger connection secured with OAuth, used to expose integrations flow for external systems to call securely.
PTM_ATP_DB_Conn	Oracle ATP Adapter	Connection to oracle Autonomous transaction processing database to read / write structured PTM data.
PTM_ERP_REST_Conn	ERP cloud Rest Adapter	Used to integrate PTM with oracle ERP cloud via Rest APIs.
PTM_ERP_Conn	ERP cloud Adapter	General oracle Erp cloud connection used for both inbound and outbound integration with PTM.

2.5 Database Objects

Schema	Object Name	Object Type	Purpose
PTM_INV_INT_146	PTM_INV_INT_146_BREAKFIX_ERROR_TBL	Table	Stores breakfix erro record

2.6 Data File Format

2.7 Web Services

Short Name	Rest/Soap Webservice	URL
getOrgName	REST	/fscmRestApi/resources/11.13.18
getSourceOnhandQty	REST	/fscmRestApi/resources/11.13.18

create_RecieveShipment_reciept	REST	/fscmRestApi/resources/11.13.18
GetInventoryItomLots	REST	/fscmRestApi/resources/11.13.18
UpdateInventoryLots	REST	/fscmRestApi/resources/11.13.18
ItemShipedValidation	REST	/fscmRestApi/resources/11.13.18
SerialNumbersValidation	REST	/fscmRestApi/resources/11.13.18
getDLPOrgName	REST	/fscmRestApi/resources/11.13.18
getItemShippedLotDffs	REST	/fscmRestApi/resources/11.13.18
createAccountAliasIssue_PartChange	REST	/fscmRestApi/resources/11.13.18
ReceiveRemainingSipments_case5	REST	/fscmRestApi/resources/11.13.18
getInventororyShipmentsCompletedTransaction	REST	/fscmRestApi/resources/11.13.18
MoveFileToError	REST	/n/{namespaceName}/b/{bucketNam

2.8 BI Publisher Report Details

Report Name	Report Path	Parameter Name
NA	NA	NA

2.9 FBDI Processing

NA

2.10 File Layout / Report Output

NA

2.11 Validation and Error Handling

1. In the event of transmission failures or successful transmissions at the Oracle end, notifications will be sent to the IS, Source System, Oracle Technical teams, and L1 Business Group. The system will have the capability to reprocess error records based on PO details. If there's an Inbound XML issue, the file must be resent from the Source System or reprocessed by running the interface again if it's a setup issue in Oracle.

2. If integration fails due to technical issues such as server issues or OIC exceptions, the data should be queued for retry. The system should have the capability to

automatically retrigger a failed record. Any system errors will be logged within OIC integration and notified to IT, ending the program in error.

3. Notifications will be sent to IT and Business stakeholders via email, detailing the error. These notifications should be consistent across platforms and based on error density and criticality. A summary email with Vehicle record and on-hand transaction Summary details will be sent to IT/Business teams.

4. Consolidated error notifications will be sent once a day for all errors encountered to business/tech stakeholders.
5. Data Validation:
   - Quantity cannot be NULL, 0 or Negative
   - DPS Number, Line Number, Part Number, Source Org, Source subinv, destination location cannot be NULL
   - Orgs should be active and destination location should be a valid float Org
6. Error Handling:
   - Error Logging: All validation errors should be logged with detailed information, including the field name, error type, and a descriptive message. This log should be accessible for troubleshooting and audit purposes.
   - Error Notification: Implement a notification system to alert relevant stakeholders (e.g., system administrators, data managers) of validation errors. Notifications can be sent via email or integrated messaging systems.
   - Retry Mechanism: For transient errors (e.g., network issues, temporary unavailability of Oracle Cloud services), implement a retry mechanism with configurable retry intervals and maximum retry attempts.
7. Data Validation:
   - Quantity cannot be NULL, 0 or Negative
   - DPS Number, Line Number, Part Number, Source Org, Source subinv, destination location cannot be NULL
   - Orgs should be active and destination location should be a valid float Org

2.12 Notification Requirements

Sr. No.	Error Scenario	Description	Action
1	NA	NA	NA

1. The error handling framework utilizes the XXXCommonErrorHandler_IS integration to log errors in the 'XXX_COMMON_AUDIT_LOG_TBL' DB table and trigger notifications based on the InterfaceID.
2. This REST-based integration also sends emails with error reports attached, with email details stored in the 'XXXCommonNotification_DVM' DVM.

3. The 'XXX_INTERFACE_TRX_TBL' keeps track of processes for an integration based on InterfaceID and FileID.
4. Log files are generated post-program completion, detailing either normal completion or error messages with specific data.
5. Non-cancelled records due to errors are consolidated in an error CSV and sent as an OIC notification attachment.
6. For file issues, it is recommended to contact the XXX administrator for Pick Pro to fix the file and resend the output for the errored record.
7. In case of missing system configuration in Oracle, the SCM team will analyze the execution status report and liaise with IT to fix the setup and reprocess the failed records or the file.

3. Application Setup and Technical Requirements

3.12 Programs/Report and Parameters

Integration Name: PTM_INV_INT_146_BreakFix_Ship_Ret

Parameter Name:

3.13 ESS Program, Scheduling and Executables

ESS Job Execution Status (Callback Response)	Description	User Action
SUCCEEDED	Request completed and was successful.	Check the details of the completed process and proceed with any post processing.
ERROR	Request ran and resulted in error.	Download the details of the error and correct the data.
WARNING	Request ran and resulted in a warning.	Download the details of the process. Check the reason for the warnings and take the necessary action to correct the input data.
CANCELED	Request was cancelled.	Resubmit the request if required.

3.14 Security and Controls Requirements

- Encryption/Decryption Details for PI Data

3.15 Archiving & Purging

3.16 Rollback and Recovery

3.17 Miscellaneous

4. File and Directory Locations

ENVIRONMENT: Development	
Azure Service Bus	Kafka Topic
Folder Location for Output Data File	/external/oic/dil/TOReceipt/messages
File Name for Input Data File	NA
Input Data File Format	□ TXT □ CSV ☒ **XML** □ Other
Folder Location for Error File (if any)	NA
File Name for Error File (if any)	NA
Folder Location for Archive File (if any)	NA
File Name for Archive File (if any)	NA
Error File Format (if any)	□ TXT □ CSV ☒ **XML** □ Other

5. Technical Unit Test

Sr. No	Conditions to be tested	Expected Result	Executed?

6. Interface/Conversion Reporting Requirements

6.1 Data Validation & Error Reporting

Error Reporting is done through NACK messages sending through Kafka.

Data Validation are handled in OIC.

6.2 Data Reconciliation

7. Open and Closed Issues

7.1

Open Issues

Issue Id	Description	Opened By	Responsible	Due Date
1.				
2.				

7.2 Closed Issues

Issue Id	Description	Resolution	Signoff	Closed Date
1.				
2.				

8. Appendix