Validation Configuration

Last Modified on 07/21/2026 7:13 am EDT

V1.0

Overview

In Genpact Deductions Recovery, a case is considered valid or invalid or partially valid for deductions based on some pre-defined rules. The system offers a four-way matching process using YAML rules to validate the deduction case line-items as valid or invalid or partially valid.

The *document-validation-type.yaml* file defines the respective fields from the header, line-level, POD and the backup data, which are considered for validation.

Starting from V1.0, the validation rules support additional fields from the source files per business requirements. Packing list data can also be added to the validation rules per business request. However, it is not included in the default validation rules.

The validation supports data normalization to correctly convert and compare units of measure (UOM), and logic to handle cases that require human intervention.

The *system-validation-ruleset.yaml* file configures the rules for validation of the extracted header, line-item POD and backup data. Based on fulfillment of these pre-defined rules the deduction case is marked as Valid, Invalid or Partially valid and processed accordingly.

The *caseform-validation-statuses.yaml* file defines the available validation status for a case in the portal and the system.
The *caseform-invalid-reasons.yaml* file configures the deduction invalid reasons that can be chosen and will appear in the case details in the portal.

Sample YAML settings

caseform-validation-statuses.yaml

```
kind: document
metadata:
  name: deduction/v1/documents/caseform-validation-statuses
spec:
    values:
      - "valid"
      - "invalid"
      - "partial"
      - "empty"
```

Parameter	Description

Parameter	Description
values	The values for the validation status. • Valid • Invalid • Partial • Empty

caseform-invalid-reasons.yaml

```
kind: document
metadata:
  name: deduction/v1/documents/caseform-invalid-reasons
spec:
    values:
        - "POD does not support shortages"
        - "POD support partial shortages"
        - "Deducted SKU is not invoiced"
        - "Customer deducted at higher price"
        - "UOM issue"
```

Parameter	Description
values	The reasons for an invalid status.

document-validation-types.yaml

```
kind: document
metadata:
  name: deduction/v1/documents/document-validation-types
spec:

  # --------------------------
  # HEADER
  # --------------------------
  - name: "Header"
    fields:
      - fieldName: PoNumber
        type: "System.String"
        defaultNull:
      - fieldName: InvoiceNumber
        type: "System.String"
        defaultNull:
      - fieldName: RptReasonCode
        type: "System.String"
        defaultNull:
      - fieldName: RptDeductionStatus
        type: "System.String"
        defaultNull:
      - fieldName: RptCustomerId
        type: "System.String"
```

```
      defaultNull:
    - fieldName: RptCustomerName
      type: "System.String"
      defaultNull:
    - fieldName: RptAmount
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: InvInvoiceBillQuantity
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: InvPoNumber
      type: "System.String"
      defaultNull:
    - fieldName: BolNumber
      type: "System.String"
      defaultNull:
    - fieldName: InvCarrier
      type: "System.String"
      defaultNull:
    - fieldName: InvCarrierReferenceNumber
      type: "System.String"
      defaultNull:
    - fieldName: VldTotalValidAmount
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: VldTotalInvalidAmount
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: VldValidationStatus
      type: "System.String"
      defaultNull:
    - fieldName: BckDeductedQuantity
      type: "System.Decimal"
      defaultNull: 0

# ---------------------------
# POD
# ---------------------------
- name: "POD"
  fields:
    - fieldName: DocumentProvider
      type: "System.String"
      defaultNull:
    - fieldName: IsCustomerSigned
      type: "System.Boolean"
      defaultNull: false
    - fieldName: CustomerSignatureText
      type: "System.String"
      defaultNull:
    - fieldName: IsCarrierSigned
      type: "System.Boolean"
      defaultNull: false
    - fieldName: CarrierSignatureText
      type: "System.String"
      defaultNull:
    - fieldName: IsSubjectToCount
      type: "System.Boolean"
      defaultNull: false
    - fieldName: FreightChargeTerms
      type: "System.String"
      defaultNull:
    - fieldName: TotalPackagesQuantity
      type: "System.Decimal"
      defaultNull: 0
```

```
    - fieldName: TotalLbsQuantity
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: BolNumberList
      type: "System.String"
      defaultNull:
    - fieldName: PodIsHandwrittenSource
      type: "System.Boolean"
      defaultNull: false
    - fieldName: RelatedBols
      type: "System.String"
      defaultNull:
    - fieldName: PodTotalShortage
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: PodTotalDamage
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: PodTotalOverage
      type: "System.Decimal"
      defaultNull: 0

# --------------------------
# PURCHASE ORDER
# --------------------------
- name: "PurchaseOrder"
  fields:
    - fieldName: PoNumber
      type: "System.String"
      defaultNull:
    - fieldName: BolNumberList
      type: "System.String"
      defaultNull:
    - fieldName: TotalPackagesQuantity
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: TotalLbsQuantity
      type: "System.Decimal"
      defaultNull: 0

# --------------------------
# LINES
# --------------------------
- name: "Lines"
  fields:
    - fieldName: BckPoNumber
      type: "System.String"
      defaultNull:
    - fieldName: InvoiceNumber
      type: "System.String"
      defaultNull:
    - fieldName: BckSKU
      type: "System.String"
      defaultNull:
    - fieldName: BckDescription
      type: "System.String"
      defaultNull:
    - fieldName: BckCustomerSku
      type: "System.String"
      defaultNull:
    - fieldName: BckDeductedPricePerQuantity
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: BckInvoicedQuantity
```

```
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: BckDeductedLineAmount
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: BckDeductedQuantity
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: BckGrossPricePerQuantityAmount
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: BckNetPricePerQuantityAmount
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: BckGrossPriceAmount
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: BckAppNumbers
    type: "System.String"
    defaultNull:
  - fieldName: InvBilledQuantity
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: InvGrossPriceAmount
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: InvNetPriceAmount
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: InvGrossPricePerQuantityAmount
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: InvNetPricePerQuantityAmount
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: InvItemDescription
    type: "System.String"
    defaultNull:
  - fieldName: InvInvoiceQtyEach
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: InvDenominator
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: InvUom
    type: "System.String"
    defaultNull:
  - fieldName: InvCustomerSKU
    type: "System.String"
    defaultNull:
  - fieldName: invBol
    type: "System.String"
    defaultNull:
  - fieldName: PodTotalShortage
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: PodTotalDamage
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: PodTotalOverage
    type: "System.Decimal"
    defaultNull: 0
  - fieldName: PckPoNumber
    type: "System.String"
```

```
      type: "System.String"
      defaultNull:
    - fieldName: PckSku
      type: "System.String"
      defaultNull:
    - fieldName: PckPackedQuantity
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: PckShipUom
      type: "System.String"
      defaultNull:
    - fieldName: PckShipmentNumber
      type: "System.String"
      defaultNull:
    - fieldName: PckExpiryDate
      type: "System.String"
      defaultNull:
    - fieldName: PckEachPerUom
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: PckCarrierRefNumber
      type: "System.String"
      defaultNull:
```

Parameter	Description
name	The name of the type of data, header, line-item, POD, or backup.
fields	The fields to be considered for validation.
fieldName	Name of the field.
type	Type of the field data.
defaultNull	Default value of the field.

system-validation-ruleset.yaml

```
kind: ruleset
metadata:
  name: deduction/v1/ruleset/system-validation-ruleset

spec:
  nodes:

    ##########################################################################
########
    # PHASE 1: general-line-validation
    # Applies to every LINE first. Since engine returns first match per line,
    # these hard validation failures must stay first in order.
    ##########################################################################
########
    - name: phase-1-general-line-validation
      if: '1==1'
      nodes:
        ##########################################################################
```

```
#####
      # G1 - PO consistency
      ##########################################################################
#####
    #  - name: g1-po-consistency
    #    if: '
    #      it.Line["InvPONumber"] != it.Line["BckPoNumber"]
    #      or it.Line["BckPoNumber"] != it.PurchaseOrder["PoNumber"]'
    #    then:
    #      validationStatus: "Invalid"
    #      invalidReason: "PO number mismatch"
    #      invalidQuantity: ':it.Line["BckDeductedQuantity"]'
    #      invalidAmount: ':Convert.ToDecimal(it.Line["BckDeductedQuantity"]) * Convert.ToDecimal(it.Line["BckDedu
ctedPricePerQuantity"])'
    #      validQuantity: 0
    #      validAmount: 0
#
    #  ##########################################################################
######
    #  # G2 - Invoice number consistency
    #  ##########################################################################
######
    #  - name: g2-invoice-number-consistency
    #    if: 'it.Header["InvoiceNumber"] != it.Line["InvoiceNumber"]'
    #    then:
    #      validationStatus: "Invalid"
    #      invalidReason: "Invoice number mismatch"
    #      invalidQuantity: ':it.Line["BckDeductedQuantity"]'
    #      invalidAmount: ':Convert.ToDecimal(it.Line["BckDeductedQuantity"]) * Convert.ToDecimal(it.Line["BckDedu
ctedPricePerQuantity"])'
    #      validQuantity: 0
    #      validAmount: 0
#
    #  ##########################################################################
######
    #  # G3 - BOL consistency
    #  ##########################################################################
######
    #  - name: g3-bol-consistency
    #    if: '
    #      (it.Line["invBol"] != null and it.Line["invBol"].ToString().Trim() != "")
    #      and
    #      (it.POD["BolNumberList"] != null)
    #      and
    #      not it.POD["BolNumberList"].ToString().Trim().ToLower()
    #        .Contains(it.Line["invBol"].ToString().Trim().ToLower())'
    #    then:
    #      validationStatus: "Invalid"
    #      invalidReason: "BOL mismatch"
    #      invalidQuantity: ':it.Line["BckDeductedQuantity"]'
    #      invalidAmount: ':Convert.ToDecimal(it.Line["BckDeductedQuantity"]) * Convert.ToDecimal(it.Line["BckDedu
ctedPricePerQuantity"])'
    #      validQuantity: 0
    #      validAmount: 0
#
#
    ##########################################################################
#####
      # G4 -  Backup->invoice SKU consistency
      ##########################################################################
#####
      - name: g4-customer-sku-consistency
        if: '
          it.Line["InvItemDescription"] == null
```

```
        and it.Line["InvPONumber"] == null'
      then:
        validationStatus: "Invalid"
        invalidReason: "Deducted SKU is not invoiced"
        invalidQuantity: ':it.Line["BckDeductedQuantity"]'
        invalidAmount: ':it.Line["BckDeductedLineAmount"]'
        validQuantity: 0
        validAmount: 0
#
    ##########################################################################
#####
    # G5 - POD signature / stamp
    ##########################################################################
#####
    - name: g5-pod-signature
      if: '
        it.POD["IsCustomerSigned"] != 1
        and it.POD["PodIsHandwrittenSource"] == 0
        and it.POD["CustomerSignatureText"] == ""
        and (
          (it.Line["PodTotalShortage"] == 0)
          and (it.Line["PodTotalDamage"] == 0)
          and (it.Line["PodTotalOverage"] == 0)
        )'
      then:
          validationStatus: "Invalid"
          invalidReason: "POD is not signed ShlomoT3"
          invalidQuantity: ':it.Line["BckDeductedQuantity"]'
          invalidAmount: ':Convert.ToDecimal(it.Line["BckDeductedQuantity"]) * Convert.ToDecimal(it.Line["BckDedu
ctedPricePerQuantity"])'
          validQuantity: 0
          validAmount: 0

  ##########################################################################
########
  # PHASE 2: generic-pod-without-sku-identifier
  # This section is only for generic POD support (no SKU identifier in POD).
  # Order is full-match -> partial-match -> unsupported/fallback.
  ##########################################################################
########
  - name: phase-2-generic-pod-without-sku-identifier
    if: 'it.POD["PodTotalDiscrepancyCasesWithIdentifier"] == 0'
    nodes:
      ##########################################################################
#####
      # P2.1 - No discrepancy at POD level -> invalid deduction
      ##########################################################################
#####
      - name: p2-no-discrepancy-invalid
        if: 'Convert.ToDecimal(it.POD["PodTotalDiscrepancyCases"]) == 0'
        then:
          validationStatus: "Invalid"
          invalidReason: "POD does not support shortages"
          invalidQuantity: ':it.Line["BckDeductedQuantity"]'
          invalidAmount: ':it.Line["BckDeductedLineAmount"]'
          validQuantity: 0
          validAmount: 0

      ##########################################################################
#####
      # P2.2 - Generic POD discrepancy fully supports backup total
      ##########################################################################
#####
      - name: p2-generic-full-support-valid
```

```
      name: p2-generic-full-support-valid
      if: 'Convert.ToDecimal(it.POD["PodTotalDiscrepancyCases"]) == Convert.ToDecimal(it.Header["BackupCaseQu
antityTotal"])'
      then:
        validationStatus: "Valid"
        validQuantity: ':it.Line["BckDeductedQuantity"]'
        validAmount: ':it.Line["BckDeductedLineAmount"]'
        invalidQuantity: 0
        invalidAmount: 0
        invalidReason: ""

    ##########################################################################
#####
    # P2.3 - Generic POD partial support when only one deducted SKU exists
    ##########################################################################
#####
    - name: p2-generic-partial-single-sku
      if: >
        Convert.ToDecimal(it.POD["PodTotalDiscrepancyCases"]) < Convert.ToDecimal(it.Header["BackupCaseQuant
ityTotal"])
        && Convert.ToDecimal(it.Header["BckSkuCount"]) == 1
      then:
        validationStatus: "Partial Valid"
        validQuantity: ':Convert.ToDecimal(it.POD["PodTotalDiscrepancyCases"])'
        validAmount: ':Convert.ToDecimal(it.POD["PodTotalDiscrepancyCases"]) * Convert.ToDecimal(it.Line["BckDe
ductedPricePerQuantity"])'
        invalidQuantity: ':Convert.ToDecimal(it.Header["BckTotalCases"]) - Convert.ToDecimal(it.POD["PodTotalDiscr
epancyCases"])'
        invalidAmount: ':(Convert.ToDecimal(it.Header["BckTotalCases"]) - Convert.ToDecimal(it.POD["PodTotalDiscr
epancyCases"])) * Convert.ToDecimal(it.Line["BckDeductedPricePerQuantity"])'
        invalidReason: "POD support partial shortages"

    ##########################################################################
#####
    # P2.4 - Generic POD partial support with multiple SKUs is not allocable
    ##########################################################################
#####
    - name: p2-generic-partial-multi-sku-unsupported
      if: >
        Convert.ToDecimal(it.POD["PodTotalDiscrepancyCases"]) > 0
        && Convert.ToDecimal(it.POD["PodTotalDiscrepancyCases"]) < Convert.ToDecimal(it.Header["BackupCaseQu
antityTotal"])
        && Convert.ToDecimal(it.Header["BckSkuCount"]) > 1
      then:
        validationStatus: "Invalid"
        invalidReason: "Generic POD discrepancy cannot be allocated across multiple SKUs"
        invalidQuantity: ':it.Line["BckDeductedQuantity"]'
        invalidAmount: ':it.Line["BckDeductedLineAmount"]'
        validQuantity: 0
        validAmount: 0

  ##########################################################################
########
  # PHASE 3: sku-identified-pod-matching
  # Applies when POD contains discrepancy with identifier (SKU-level proof).
  # Here SKU-level evidence has priority over generic POD totals.
  ##########################################################################
########
  - name: phase-3-sku-identified-pod-matching
    if: 'it.POD["PodTotalDiscrepancyCasesWithIdentifier"] > 0'
    nodes:
      ##########################################################################
#####
      # P3.1 - No shortage/damage/overage support on this line
      ##########################################################################
```

```
    ##########################################################################
#####
    - name: p3-no-osd-support-on-line
      if: '
        it.Line["PodTotalShortage"] == 0
        and it.Line["PodTotalDamage"] == 0
        and it.Line["PodTotalOverage"] == 0'
      then:
        validationStatus: "Invalid"
        invalidReason: "POD does not support Shortage"
        invalidQuantity: ':it.Line["BckDeductedQuantity"]'
        invalidAmount: ':it.Line["BckDeductedLineAmount"]'
        validQuantity: 0
        validAmount: 0

    ##########################################################################
#####
    # P3.2 - Partial shortage support on this line
    ##########################################################################
#####
    - name: p3-partial-shortage
      if: '
        it.Line["BckInvoicedQuantity"] != 0
        and Convert.ToDecimal(it.Line["PodTotalShortage"])
            < (it.Line.ContainsKey("backupCaseQuantity")
              ? Convert.ToDecimal(it.Line["backupCaseQuantity"])
              : Convert.ToDecimal(it.Line["BckDeductedQuantity"]))'
      then:
        validationStatus: "Partial Valid"
        validQuantity: ':Math.Min(Convert.ToDecimal(it.Line["PodTotalShortage"]), Convert.ToDecimal(it.Line["BckDe
ductedQuantity"]))'
        validAmount: ':Math.Min(Convert.ToDecimal(it.Line["PodTotalShortage"]), Convert.ToDecimal(it.Line["BckDed
uctedQuantity"])) * Convert.ToDecimal(it.Line["BckDeductedLineAmount"])'
        invalidQuantity: ':Convert.ToDecimal(it.Line["BckDeductedQuantity"]) - Math.Min(Convert.ToDecimal(it.Line["
PodTotalShortage"]), Convert.ToDecimal(it.Line["BckDeductedQuantity"]))'
        invalidAmount: ':(Convert.ToDecimal(it.Line["BckDeductedQuantity"]) - Math.Min(Convert.ToDecimal(it.Line["
PodTotalShortage"]), Convert.ToDecimal(it.Line["BckDeductedQuantity"]))) * Convert.ToDecimal(it.Line["BckDeducte
dLineAmount"])'
        invalidReason: "Partial Valid"

    ##########################################################################
#####
    # P3.3 - Full shortage support on this line
    ##########################################################################
#####
    - name: p3-full-shortage
      if: '
        it.Line["BckInvoicedQuantity"] != 0
        and Convert.ToDecimal(it.Line["PodTotalShortage"])
            >= (it.Line.ContainsKey("backupCaseQuantity")
              ? Convert.ToDecimal(it.Line["backupCaseQuantity"])
              : Convert.ToDecimal(it.Line["BckDeductedQuantity"]))'
      then:
        validationStatus: "Valid"
        validQuantity: ':it.Line["BckDeductedQuantity"]'
        validAmount: ':it.Line["BckDeductedLineAmount"]'
        invalidQuantity: 0
        invalidAmount: 0
        invalidReason: ""

    ##########################################################################
#####
    # P3.4 - Optional amount-match shortcut (kept disabled)
    ##########################################################################
" " " " "
```

```
#####
      #- name: p3-amount-match-valid
      #  if: 'Convert.ToDecimal(it.Line["BckDeductedLineAmount"]) == Convert.ToDecimal(it.POD["PodTotalShortage
"])'
      #  then:
      #    validationStatus: "Valid"
      #    invalidReason: ""
      #    validQuantity: ':it.Line["BckDeductedQuantity"]'
      #    invalidQuantity: 0
```

Parameter	Description
If then	The condition based on which the rules will be applied.
ValidationStatus	The validation status of the case based on rules.
ValidQuantity	The deduction quantity in case of valid deduction.
ValidAmount	The deduction amount in case of valid deduction.
InvalidQuantity	The deduction quantity in case of invalid deduction.
InvalidAmount	The deduction amount in case of invalid deduction.
InvalidReason	The deduction reason in case of invalid deduction.
ProductItemId	The unique Id of the product for which deduction is processed.

For example, in the above template:

```
- if: 'it.Line["BilledQtyInvoice"] != 0 and
         it.Line["BackupPricePerQty"] == it.Line["InvoiceNetPricePerQty"] and
         it.Line["Shortage"] == it.Line["DeductedQty"]'
    then:
      validationStatus: "Valid"
      validQuantity: ':it.Line["DeductedQty"]'
      validAmount: ':it.Line["DeductAmt"]'
      invalidQuantity: 0
      invalidAmount: 0
      invalidReason: ""
      productitemId: ':it.Line["ProductItemId"]'
```

If the Billed quantity in invoice is not 0 and the price per quantity in the backup file is equal to the net price per quantity in the invoice and the shortage is equal to the deducted quantity then the deduction is considered valid and the case validation status is set to Valid in the portal.

Version Beta

Overview

In Genpact Deductions Recovery, a case is considered valid or invalid or partially valid for deductions based on some pre-defined rule. The *document-validation-type.yaml* file defines the respective fields from the header, line-level and the POD data, which are considered for validation.

The *document-validation.yaml* file configures the rules for validation of the extracted header, line-item and POD data. Based on fulfillment of these pre-defined rules the deduction case is marked as Valid, Invalid or Partially valid and processed accordingly.

The *caseform-validation-statuses.yaml* file defines the available validation status for a case in the portal and the system.
The *caseform-invalid-reasons.yaml* file configures the deduction invalid reasons that can be chosen and will appear in the case details in the portal.

Sample YAML settings

caseform-validation-statuses.yaml

```
kind: document
metadata:
  name: deduction/v1/documents/caseform-validation-statuses
spec:
    values:
      - "valid"
      - "invalid"
      - "partial"
      - "empty"
```

Parameter	Description
values	The values for the validation status. • Valid • Invalid • Partial • Empty

caseform-invalid-reasons.yaml

```
kind: document
metadata:
  name: deduction/v1/documents/caseform-invalid-reasons
spec:
    values:
        - "asaf test"
        - "test"
        - "POD does not support shortages"
        - "POD support partial shortages"
        - "Deducted SKU is not invoiced"
        - "Customer deducted at higher price"
        - "UOM issue"
```

Parameter	Description
values	The reasons for an invalid status.

document-validation-types.yaml

```
kind: document
metadata:
  name: deduction/v1/documents/document-validation-types
spec:
  # --------------------------
  # HEADER
  # --------------------------
  - name: "Header"
    fields:
      - fieldName: InvoiceNumber
        type: "System.String"
        defaultNull:
      - fieldName: ReasonCode
        type: "System.String"
        defaultNull:
      - fieldName: DeductionStatus
        type: "System.String"
        defaultNull:
      - fieldName: CustomerID
        type: "System.String"
        defaultNull:
      - fieldName: Amount
        type: "System.Decimal"
        defaultNull: 0
      - fieldName: InvoiceBillQuantity
        type: "System.Decimal"
        defaultNull: 0
      - fieldName: PO_OrderNumber
        type: "System.String"
        defaultNull:
      - fieldName: Bol
        type: "System.String"
        defaultNull:
      - fieldName: Carrier
        type: "System.String"
        defaultNull:
      - fieldName: CarrierReferenceNumber
        type: "System.String"
        defaultNull:
      - fieldName: PODNumberOfPackages
        type: "System.Decimal"
        defaultNull: 0
      - fieldName: Document_provider
        type: "System.String"
        defaultNull:
      - fieldName: Carrier_sign
        type: "System.Boolean"
        defaultNull: false
      - fieldName: Customer_sign
        type: "System.Boolean"
        defaultNull: false
      - fieldName: CustomerName
        type: "System.String"
        defaultNull:
```

```
    - fieldName: SubjectToCount
      type: "System.Boolean"
      defaultNull: false


# --------------------------
# POD
# --------------------------
- name: "POD"
  fields:
    - fieldName: document_provider
      type: "System.String"
      defaultNull:
    - fieldName: customer_sign
      type: "System.Boolean"
      defaultNull: false
    - fieldName: customer_sign_text
      type: "System.String"
      defaultNull:
    - fieldName: carrier_sign
      type: "System.Boolean"
      defaultNull: false
    - fieldName: carrier_sign_text
      type: "System.String"
      defaultNull:
    - fieldName: subject_to_count
      type: "System.Boolean"
      defaultNull: false
    - fieldName: freight_charge_terms
      type: "System.String"
      defaultNull:
    - fieldName: total_packages_quantity
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: total_lbs_quantity
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: Bol
      type: "System.String"
      defaultNull:
    - fieldName: TotalShortage
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: TotalDamage
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: TotalOverage
      type: "System.Decimal"
      defaultNull: 0


# --------------------------
# PURCHASE ORDER
# --------------------------
- name: "PurchaseOrder"
  fields:
    - fieldName: order_number
      type: "System.String"
      defaultNull:
    - fieldName: TotalLbsQty
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: TotalPackageQty
      type: "System.Decimal"
```

```
      type: "System.Decimal"
      defaultNull: 0
# --------------------------
# LINES
# --------------------------
- name: "Lines"
  fields:
    - fieldName: InvoiceId
      type: "System.String"
      defaultNull:
    - fieldName: InvoiceNumber
      type: "System.String"
      defaultNull:
    - fieldName: SKU
      type: "System.String"
      defaultNull:
    - fieldName: ProductItemId
      type: "System.String"
      defaultNull:
    - fieldName: ItemName
      type: "System.String"
      defaultNull:
    - fieldName: CustomerSKU
      type: "System.String"
      defaultNull:
    - fieldName: BackupPricePerQty
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: BilledQtyBackup
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: DeductAmt
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: DeductedQty
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: BilledQtyInvoice
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: GrossPrice
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: InvoiceGrossPricePerQty
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: InvoiceNetPricePerQty
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: NetPrice
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: InvoiceItemDescription
      type: "System.String"
      defaultNull:
    - fieldName: InvoiceQtyEach
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: Shortage
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: Damaged
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: Overage
```

```
    - fieldName: Overage
      type: "System.Decimal"
      defaultNull: 0
    - fieldName: PackListQuantity
      type: "System.Decimal"
      defaultNull: 0
```

Parameter	Description
name	The name of the type of data, header, line-item or POD.
fields	The fields to be considered for validation.

document-validation.yaml

```
kind: ruleSet
metadata:
  name: deduction/v1/ruleset/document-validation
spec:

  ##########################################################################
##################################
  # !!!!!!! RULE EXECUTION PRIORITY !!!!!!
  # Validation rules are evaluated in sequence, from top to bottom.
  # All matching rules will run (not just the first one).
  # However, rules defined earlier take precedence when setting or overriding values.
  # Therefore, place higher-priority or more specific rules at the TOP of each node section.
  # Additionally Any then section must include the line --productitemId: ':it.Line["ProductItemId"]'--
  ##########################################################################
##################################

  nodes:

  ##########################################################################
Clorox ##############################################
  - if: '1 == 1' #'it.Header["CustomerName"] == "Clorox"'
    nodes:

    ##########################################################################
# HEADER ##############################################
    - if: 'it.Header["InvoiceBillQuantity"]!=0 and  it.Header["InvoiceBillQuantity"] == it.POD["total_packages_quantity"
]'
      then:
        validationStatus: "Invalid"
        invalidReason: "POD # packages match in invoice"
        productitemId: ':it.Line["ProductItemId"]'
        invalidQuantity: ':it.Line["DeductedQty"]'
        invalidAmount: ':it.Line["DeductAmt"]'

    ##########################################################################
# INVALID ##############################################

    # 1. Invalid: SKU is not invoiced
    - if: 'it.Line["BilledQtyInvoice"] == 0'
      then:
        validationStatus: "Invalid"
        invalidReason: "SKU is not invoiced"
```

```
      productitemId: ':it.Line["ProductItemId"]'
      invalidQuantity: ':it.Line["DeductedQty"]'
      invalidAmount: ':it.Line["DeductAmt"]'

  # 2. Invalid: Customer deducted at higher price
  - if: 'it.Line["InvoiceNetPricePerQty"] != 0 and ToDecimal(it.Line["BackupPricePerQty"]) > ToDecimal(it.Line["Invoi
ceNetPricePerQty"])'
    then:
      validationStatus: "Invalid"
      invalidReason: "Customer deducted at higher price"
      productitemId: ':it.Line["ProductItemId"]'
      invalidQuantity: ':it.Line["DeductedQty"]'
      invalidAmount: ':it.Line["DeductAmt"]'

  # 3. Invalid: POD does not support shortages
  - if: 'it.Line["Shortage"] == 0'
    then:
      validationStatus: "Invalid"
      invalidReason: "POD does not support shortages"
      productitemId: ':it.Line["ProductItemId"]'
      invalidQuantity: ':it.Line["DeductedQty"]'
      invalidAmount: ':it.Line["DeductAmt"]'

  ##########################################################################
# PARTIAL ##############################################

  - if: 'it.Line["BilledQtyInvoice"] != 0 and
        it.Line["BackupPricePerQty"] == it.Line["InvoiceNetPricePerQty"] and
        it.Line["Shortage"] > 0 and
        ToDecimal(it.Line["Shortage"]) < ToDecimal(it.Line["DeductedQty"])'
    then:
      validationStatus: "Partial Valid"
      validQuantity: ':Math.Min(ToDecimal(it.Line["Shortage"]), ToDecimal(it.Line["DeductedQty"]))'
      validAmount: ':Math.Min(ToDecimal(it.Line["Shortage"]), ToDecimal(it.Line["DeductedQty"])) * ToDecimal(it.Lin
e["BackupPricePerQty"])'
      invalidQuantity: ':ToDecimal(it.Line["DeductedQty"]) - Math.Min(ToDecimal(it.Line["Shortage"]), ToDecimal(it.Li
ne["DeductedQty"]))'
      invalidAmount: ':(ToDecimal(it.Line["DeductedQty"]) - Math.Min(ToDecimal(it.Line["Shortage"]), ToDecimal(it.Li
ne["DeductedQty"]))) * ToDecimal(it.Line["BackupPricePerQty"])'
      invalidReason: "Partial Valid"
      productitemId: ':it.Line["ProductItemId"]'

  ##########################################################################
# VALID ################################################

  - if: 'it.Line["BilledQtyInvoice"] != 0 and
        it.Line["BackupPricePerQty"] == it.Line["InvoiceNetPricePerQty"] and
        it.Line["Shortage"] == it.Line["DeductedQty"]'
    then:
      validationStatus: "Valid"
      validQuantity: ':it.Line["DeductedQty"]'
      validAmount: ':it.Line["DeductAmt"]'
      invalidQuantity: 0
      invalidAmount: 0
      invalidReason: ""
      productitemId: ':it.Line["ProductItemId"]'

  ##########################################################################
# DEFAULT NO HIT ###############################################

  - if: 'true'
    then:
      validationStatus: "DEFAULT NO HIT"
      validQuantity: 0
```

```
      validAmount: 0
      invalidQuantity: 0
      invalidAmount: 0
      invalidReason: "DEFAULT NO HIT"
      productitemId: ':it.Line["ProductItemId"]'

 ##########################################################################
Amazon ##############################################
 - if: 'it.Header["CustomerName"] == "Amazon"'
   nodes:
    - if: '1 == 1'
      then:
        validationStatus: "Amazon"
        productitemId: ':it.Line["ProductItemId"]'
```

Parameter	Description
if then	The condition based on which the rules will be applied.
ValidationStatus	The validation status of the case based on rules.
ValidQuantity	The deduction quantity in case of valid deduction.
ValidAmount	The deduction amount in case of valid deduction.
InvalidQuantity	The deduction quantity in case of invalid deduction.
InvalidAmount	The deduction amount in case of invalid deduction.
InvalidReason	The deduction reason in case of invalid deduction.
ProductItemId	The unique Id of the product for which deduction is processed.

For example, in the above template:

```
- if: 'it.Line["BilledQtyInvoice"] != 0 and
         it.Line["BackupPricePerQty"] == it.Line["InvoiceNetPricePerQty"] and
         it.Line["Shortage"] == it.Line["DeductedQty"]'
    then:
      validationStatus: "Valid"
      validQuantity: ':it.Line["DeductedQty"]'
      validAmount: ':it.Line["DeductAmt"]'
      invalidQuantity: 0
      invalidAmount: 0
      invalidReason: ""
      productitemId: ':it.Line["ProductItemId"]'
```

If the Billed quantity in invoice is not 0 and the price per quantity in the backup file is equal to the net price per quantity in the invoice and the shortage is equal to the deducted quantity then the deduction is considered valid and the case validation status is set to Valid in the portal.