TRACELINK UNIVERSITY

Home
Resources
TraceLink University

Inventory update confirmation canonical guidelines

An Inventory Update Confirmation is an electronic message sent in response to an Inventory Update Request to verify that the requested changes to stock status, usability, or location have been completed. It includes a reference to the original request, updated property details and product identifiers (NDC, GTIN, lot, or serial number), the new stock status or disposition (e.g., available, quarantined, destroyed), any location changes, the date and time the update was applied. An Inventory Update Confirmation verifies that important actions, such as quarantined products, security checks, and adjusting cold chain handling, have been executed.

Canonical guidelines

Canonical Element	Type Definition	Description	IDoc mapping (standard & customized IDoc)	X12 mapping	EDIFACT mapping
data	–	JSon object type/root element.	–	–	–
controlFileHeader	–	Canonical control file header to store data for control segments and envelopes of IDocs, and X12 and EDIFACT EDI messages.	–	–	–
canonicalInventoryUpdateConfirmationHeader	–	Inventory balance message header	Segment E1DRM01	Segment BIA	Segment BGM
processingFunctionTypeCode	string	Processing function, transaction set purpose code. Valid values include: CREATE CHANGE CANCEL ORIGINAL REPLACE STATUS INFORMATION Use Java call-out table CL_ProcessingFunctionTypeCode.txt	E1DRM01/ACTION Use Java call-out table CL_ProcessingFunctionTypeCode.txt	BIA01 Use Java call-out table CL_ProcessingFunctionTypeCode.txt	BGM030 Use Java call-out table CL_ProcessingFunctionTypeCode.txt
inventoryBalanceReportType	string	Report type for message. Valid value is SELLERINVENTORY.	–	BIA02 = SI	BGM010-010 = 263 (Inventory adjustment status report))
messageTypeIdentifierCode	string	Document/message name	–	–	BGM010-040
warehouseIdentifier	string	Warehouse identifier.	–	–	–
warehouseGroupIdentifier	string	Warehouse group identifier represents a trading group for multiple companies that use the same warehouse-space.	–	–	–
transactionReferenceIdentifierList (1)	array	Instance 1 transaction references for inventory update response mapping to X12 BIA and EDIFACT BGM segments.	Segment E1DRM01 Note: Map IDoc number to E1DRM01/TRANSNO	Segment BIA	Segment BGM
b2bTransactionIdentifierType	string	Referenced document type. Valid value is INVENTORYUPDATECONFIRMATION	b2bTransactionIdentifierType = INVENTORYUPDATECONFIRMATION	b2bTransactionIdentifierType = INVENTORYUPDATECONFIRMATION	b2bTransactionIdentifierType = INVENTORYUPDATECONFIRMATION
b2bTransactionIdentifierValue	string	Reference document identifier for the inventory update confirmation report.	TRANSNO If IDoc field not populated, map from EDI_DC40/DOCNUM	BIA03	BGM020-010
transactionDate	string	Date transaction created	–	BIA04 BIA05 (Time if present)	–

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping (standard & customized IDoc)	X12 mapping	EDIFACT mapping
epochTransactionDate	integer	Date transaction created in epoch integer date format. Each string date recorded will have an epoch date format generated by a java exit.	–	–	–
lineItemNumber	string	Line item number in customer purchase order. Line item level only.	–	–	–
transactionReferenceIdentifierList (2)	array	Instance 2 transaction references for inventory update request at header level for all other document types	–	Segment REF	Segment RFF
b2bTransactionIdentifierType	string	Referenced document type. Valid values include: • BOL • DOCUMENT • PURCHASEORDER Use Java call-out to lookup table CL_B2BTransactionIdentifierType.txt.	–	REF01 b2bTransactionIdentifierType = anything Use Java call-out to lookup table CL_B2BTransactionIdentifierType.txt	RFF010-010 b2bTransactionIdentifierType = anything Use Java call-out to lookup table CL_B2BTransactionIdentifierType.txt
b2bTransactionIdentifierValue	string	Document identifier for transaction referenced by b2bTransactionIdentifierType.	–	REF02 as defined by REF01	RFF010-020 as defined by RFF010-010
transactionDate	string	Date transaction created.	–	–	–
epochTransactionDate	integer	Date transaction created in epoch integer date format. Each string date recorded will have an epoch date format generated by a java exit.	–	–	–
lineItemNumber	string	Line item number in customer purchase order. Line item level only.	–	–	–
mpcDates	array	Dates relevant for the batch master.	Segment E1DRM01	Segment DTM	Segment DTM
dateType	string	Date type at header level for inventory balance in date format YYYYMMDD and time format (IDoc, X12) HHMMSS. Valid values include: • EFFECTIVE • REPORTSTART • REPORTEND • DOCUMENT Use Java call-out to lookup table CL_DateType.txt	–	DTM01 = 090 or 091 or anything else Use Java call-out to lookup table CL_DateType.txt	DTM010-010 = "7" (Effective date) or "137" (document/message date) or anything else Use Java call-out to lookup table CL_DateType.txt
dateValue	string	Date defined by referenced date type in date format YYYY-MM-DD. Dates will be populated with string of 0's if no date found for any of these date types. Maps will pass thru whatever it finds in IDoc date and time fields.	• E1DRM01/EFF_DATE • E1DRM01/EFF_TIME • E1DRM01/TRANDATE • E1DRM01/TRANTIME	DTM02 where DTM01 as defined by DTM01 BIA04 (if not populated by instance 1 transactionReferenceIdentifierList) BIA05 (if time present)	DTM010-020 where DTM010-010 = "7" (Effective date) or "137" (document/message date)
epochDateValue	integer	Date defined by referenced date type generated in epoch integer date format. Each string date recorded will have an epoch date format generated by a java exit.	–	–	–
timeZone	string	Time zone for referenced dates.	–	–	–
freeTextNote	array	–	–	–	FTX
textReferenceCode	string	Text reference or subject code. Will map whatever is found.	–	–	FTX010
textFunctionCode	string	Text function code. Will map whatever is found.	–	–	FTX020
language	string	Language of free text.	–	–	FTX050
textFormatCode	string	Format code for free text.	–	–	
freeText	string	Free test string.	–	–	FTX040-010
customerBusinessInformation	–	Customer party company identifiers and address information.	Segment E1ADRM0 where PARTNER_Q = "AG"	Segment loop group N1 N101 = "BT"	Segmenyt loop group NAD NAD010 = "BT"
businessName	string	Business name for partner or location.	E1ADRM0/NAME1	N102	NAD030-010
additionalBusinessName	string	Additional business name or overflow of business name.	E1ADRM0/NAME2	N201	NAD030-020
mpcCompanyLocationIdentifierList	array	Identifier list for business party.	–	–	–
isIdentifierPresentInEvent	boolean	Is identifier present in event boolean.	–	–	–
companyIdentifierType	string	Regulatory company identifier type for business partner.	–	N103	NAD020-030
companyIdentifierValue	string	Company identifier value for business partner.	E1ADRM0/PARTNER_ID	N104	NAD020-010
alternateCompanyLocationIdentifierValue	string	Alternate company location identifier value.	–	–	–
address	–	Party address information.	–	–	–
address1	string	Main street address.	E1ADRM0/STREET1	N301	NAD050-010
address2	string	Supplemental street address.	E1ADRM0/STREET2	N302	NAD050-020
city	string	City.	E1ADRM0/CITY1	N401	NAD060
district	string	District.	–	–	–
state	string	State or region code.	E1ADRM0/REGION	N402	NAD070-010
postalCode	string	Postal code.	E1ADRM0/POSTL_COD1	N403	NAD080
country	string	Country code.	E1ADRM0/COUNTRY1	N404	NAD090
telephone	string	Telephone number.	E1ADRM0/TELEPHONE1	–	–
fax	string	Fax number.	E1ADRM0/TELEFAX	–	–
url	string	Website URL.	–	–	–
buildingNumber	string	Building number.	–	–	–
plantNumber	string	Plant number.	–	–	–
floorNumber	string	Floor number in building.	–	–	–
lscAddressIdentifier	string	LSC address identifier.	–	–	–
village	string	Village name.	–	–	–
houseNumber	string	House number.	–	–	–
township	string	Township.	–	–	–
premisesCode	string	Premises code.	–	–	–

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping (standard & customized IDoc)	X12 mapping	EDIFACT mapping
companyLocationContactInformation	–	–	–	Segment N1/PER	Hard code CTA010 to "RP" responsible person
contactType	string	Contact type. Valid values: • PRIMARY • ORDERCONTACT • PRODUCTMANAGER	–	PER01 = • AJ (PRIMARY) or • OC (ORDERCONTACT) or • PM (PRODUCTMANAGER)	Hard code CTA010 to "RP" responsible person
contactName	string	Contact name	–	PER02	CTA020-020
contactTitle	string	Title of contact	–	–	–
contactPhone	string	Contact phone number.	–	PER04 where PER03 = TE or PER06 where PER05 = TE	COM010-010 where COM010-020 = "TE"
contactEmail	string	Contact email address.	–	PER04 where PER03 = EM or PER06 where PER05 = EM	COM010-010 where COM010-020 = "EM"
supplierBusinessInformation	–	Supplier party company identifiers and address information.	Segment E1ADRM0 where PARTNER_Q = "LF"	Segment loop group N1 N101 = "SU"	Segment loop group NAD NAD010 = "SU"
businessName	string	Business name for partner or location.	E1ADRM0/NAME1	N102	NAD030-010
additionalBusinessName	string	Additional business name or overflow of business name.	–	N201	NAD030-020
mpcCompanyLocationIdentifierList	array	Identifier list for business party.	–	–	–
isIdentifierPresentInEvent	boolean	Is identifier present in event boolean.	–	–	–
companyIdentifierType	string	Regulatory company identifier type for business partner.	–	N103	NAD020-030
companyIdentifierValue	string	Company identifier value for business partner.	E1ADRM0/PARTNER_ID	N104 CVS - REF01 hard code VR REF02	NAD020-010
alternateCompanyLocationIdentifierValue	string	Alternate company location identifier value.	–	–	–
address	–	Party address information.	–	Segment N3, N4	–
address1	string	Main street address.	E1ADRM0/STREET1	N301	NAD050-010
address2	string	Supplemental street address.	E1ADRM0/STREET2	N302	NAD050-020
city	string	City	E1ADRM0/CITY1	N401	NAD060
district	string	District	–	–	–
state	string	State or region code	E1ADRM0/REGION	N402	NAD070-010
postalCode	string	Postal code	E1ADRM0/POSTL_COD1	N403	NAD080
country	string	Country code	E1ADRM0/COUNTRY1	N404	NAD090
telephone	string	Telephone number	E1ADRM0/TELEPHONE1	–	–
fax	string	Fax number	E1ADRM0/TELEFAX	–	–
url	string	Website URL	–	–	–
buildingNumber	string	Building number	–	–	–
plantNumber	string	Plant number	–	–	–
floorNumber	string	Floor number in building	–	–	–
lscAddressIdentifier	string	LSC address identifier	–	–	–
village	string	Village name	–	–	–
houseNumber	string	House number	–	–	–
township	string	Township	–	–	–
premisesCode	string	Premises code	–	–	–
companyLocationContactInformation	–	–	–	Segment N1/PER	Hard code CTA010 to "RP" responsible person
contactType	string	Contact type. Valid values: • PRIMARY • ORDERCONTACT • PRODUCTMANAGER	–	PER01 = • AJ (PRIMARY) or • OC (ORDERCONTACT) or • PM (PRODUCTMANAGER)	Hard code CTA010 to "RP" responsible person
contactName	string	Contact name	–	PER02	CTA020-020
contactTitle	string	Title of contact	–	–	–
contactPhone	string	Contact phone number	–	PER04 where PER03 = TE or PER06 where PER05 = TE	COM010-010 where COM010-020 = "TE"
contactEmail	string	Contact email address	–	PER04 where PER03 = EM or PER06 where PER05 = EM	COM010-010 where COM010-020 = "EM"
reportingLocationBusinessInformation	–	Reporting location company identifiers and address information.	For IDoc, the reporting location could be the CMO, the manufacturer	Segment loop group N1 N101 = "RL"	Segment loop group NAD NAD010 = "GY"
businessName	string	Business name for partner or location.	–	N102	NAD030-010
additionalBusinessName	string	Additional business name or overflow of business name.	–	N201	NAD030-020
mpcCompanyLocationIdentifierList	array	Identifier list for business party.	–	–	–
isIdentifierPresentInEvent	boolean	Is identifier present in event boolean.	–	–	–
companyIdentifierType	string	Regulatory company identifier type for business partner.	–	N103	NAD020-030
companyIdentifierValue	string	Company identifier value for business partner.	–	N104	NAD020-010
alternateCompanyLocationIdentifierValue	string	Alternate company location identifier value	–	–	–
address	–	Party address information.	–	–	–
address1	string	Main street address	–	N301	NAD050-010
address2	string	Supplemental street address	–	N302	NAD050-020
city	string	City	–	N401	NAD060
district	string	District	–	–	–
state	string	State or region code	–	N402	NAD070-010
postalCode	string	Postal code	–	N403	NAD080
country	string	Country code	–	N404	NAD090
telephone	string	Telephone number	–	–	–
fax	string	Fax number	–	–	–
url	string	Website URL	–	–	–
buildingNumber	string	Building number	–	–	–
plantNumber	string	Plant number	–	–	–
floorNumber	string	Floor number in building	–	–	–
lscAddressIdentifier	string	LSC address identifier	–	–	–
village	string	Village name	–	–	–

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping (standard & customized IDoc)	X12 mapping	EDIFACT mapping
houseNumber	string	House number	–	–	–
township	string	Township	–	–	–
premisesCode	string	Premises code	–	–	–
companyLocationContactInformation	–	–	–	Segment N1/PER	Hard code CTA010 to "RP" responsible person
contactType	string	Contact type. Valid values: • PRIMARY • ORDERCONTACT • PRODUCTMANAGER	–	PER01 = • AJ (PRIMARY) or • OC (ORDERCONTACT) or • PM (PRODUCTMANAGER)	Hard code CTA010 to "RP" responsible person
contactName	string	Contact name	–	PER02	CTA020-020
contactTitle	string	Title of contact	–		
contactPhone	string	Contact phone number	–	PER04 where PER03 = TE or PER06 where PER05 = TE	COM010-010 where COM010-020 = "TE"
contactEmail	string	Contact email address	–	PER04 where PER03 = EM or PER06 where PER05 = EM	COM010-010 where COM010-020 = "EM"
shipToLocationInformation	–	Ship to party company identifiers and address information.	E1ADRM0 where PARTNER_Q = "WE"	Segment loop group N1 N101 = "ST"	Segment loop group NAD NAD010 = "ST"
businessName	string	Business name for partner or location.	E1ADRM0/NAME1	N102	NAD030-010
additionalBusinessName	string	Additional business name or overflow of business name.	–	N201	NAD030-020
mpcCompanyLocationIdentifierList	array	Identifier list for business party.	–	–	–
isIdentifierPresentInEvent	boolean	Is identifier present in event boolean.	–	–	–
companyIdentifierType	string	Regulatory company identifier type for business partner.	–	N103	NAD020-030
companyIdentifierValue	string	Company identifier value for business partner.	E1ADRM0/PARTNER_ID	N104	NAD020-010
alternateCompanyLocationIdentifierValue	string	Alternate company location identifier value.	–	–	–
address	–	Party address information.	–	–	–
address1	string	Main street address.	E1ADRM0/STREET1	N301	NAD050-010
address2	string	Supplemental street address.	E1ADRM0/STREET2	N302	NAD050-020
city	string	City	E1ADRM0/CITY1	N401	NAD060
district	string	District	–	–	–
state	string	State or region code	E1ADRM0/REGION	N402	NAD070-010
postalCode	string	Postal code	E1ADRM0/POSTL_COD1	N403	NAD080
country	string	Country code	E1ADRM0/COUNTRY1	N404	NAD090
telephone	string	Telephone number	E1ADRM0/TELEPHONE1	–	–
fax	string	Fax number	E1ADRM0/TELEFAX	–	–
url	string	Website URL	–	–	–
buildingNumber	string	Building number	–	–	–
plantNumber	string	Plant number	–	–	–
floorNumber	string	Floor number in building	–	–	–
lscAddressIdentifier	string	LSC address identifier	–	–	–
village	string	Village name	–	–	–
houseNumber	string	House number	–	–	–
township	string	Township	–	–	–
premisesCode	string	Premises code	–	–	–
companyLocationContactInformation	–	–	–	Segment N1/PER	Segmenyt NAD/CTA/COM
contactType	string	Contact type. Valid values: • PRIMARY • ORDERCONTACT • PRODUCTMANAGER	–	PER01 = • AJ (PRIMARY) or • OC (ORDERCONTACT) or • PM (PRODUCTMANAGER)	Hard code CTA010 to "RP" responsible person
contactName	string	Contact name.	–	PER02	CTA020-020
contactTitle	string	Title of contact.	–	–	–
contactPhone	string	Contact phone number.	–	PER04 where PER03 = TE or PER06 where PER05 = TE	COM010-010 where COM010-020 = "TE"
contactEmail	string	Contact email address.	–	PER04 where PER03 = EM or PER06 where PER05 = EM	COM010-010 where COM010-020 = "EM"
canonicalInventoryUpdateConfirmationItemDetails	array	Inventory balance message line item details.	Segment E1DRM02	Segment loop group LIN	Segment loop group 9 LIN
lineItemNumber	string	Line item number for inventory balance report.	E1DRM02/POSEX	LIN01	LIN010
processingFunctionTypeCode	string	Processing function, transaction set purpose code. Valid values include: • CREATE • CHANGE • CANCEL • ORIGINAL • REPLACE • STATUS • INFORMATION Use Java call-out table CL_ProcessingFunctionTypeCode.txt	E1DRM02/ACTION Use Java call-out table CL_ProcessingFunctionTypeCode.txt		LIN020 Use Java call-out table CL_ProcessingFunctionTypeCode.txt
storageLocation	string	Location where stock is stored.	–	–	–
plantIdentifier	string	Plant identifier.	YMDRM02/WERKS	LIN/REF01 = PE (Plant identifier)	LIN/RFF01 = PE (Plant identifier)
productionMaterialState	string	Current state of product in production.	–	–	–
palletIdentifier	string	Pallet Identifier.	–	–	–
productCodesIdentifiers	array	Internal material identifier in the backend/ERP systems for the product. In the delivery this will be the supplier, although the customer and other identifiers may also be present.	Segment E1DRM02/E1DRM04/E1EDP19	Segment loop group LIN	Segment loop group LIN and PIA for additional identifiers

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping (standard & customized IDoc)	X12 mapping	EDIFACT mapping
productCodeType	string	Code to qualify product identifier type: • GTIN-14 • NTIN • CN-RESCODE • VENDORPARTNUM • UPC • SKU • CUSTOMER • SUPPLIER • MANUFACTURER Use java call-out to lookup table CL_ProductCodeType	QUALF = • 001 or • 002 or • 003 or • 004 Use java call-out to lookup table CL_ProductCodeType	• LIN02 or • LIN04 or • LIN06 or • LIN08 or • LIN10 or • LIN12 Use java call-out to lookup table CL_ProductCodeType	LIN030-020 Additional product types: PIA010 = 1 (additional identification) PIA020-020 PIA030-020 PIA040-020 PIA050-020 PIA060-020 Use java call-out to lookup table CL_ProductCodeType
productCodeValue	string	Product code identifier value.	E1EDP19/IDTNR Create 1 instance E1EDP19 for each qualifier present	LIN03 or LIN05 or LIN07 or LIN09 or LIN11 or LIN13	LIN030-010 Additional product codes: PIA020-010 PIA030-010 PIA040-010 PIA050-010 PIA060-010
productItemInformation	array	Product item code information. Only using a subset of all available attributes.	Segment E1DRM02/E1DRM04/E1EDP19	Segment PID	Segment IMD
productLanguageCode	string	Language for product description.	–	–	IMD030-040, IMD030-050
productName	string	Name of product.	E1EDP19/KTEXT	PID05	IMD030-040, IMD030-060
genericName	string	Generic name for product.	–	–	–
productDescription	string	Description of the product defined by the current product identifier.	–	–	–
lotNumber	string	Lot or batch identifier.	E1EDP19/IDTNR where QUALF = "010"	• LIN03 or • LIN05 or • LIN07 or • LIN09 or • LIN11 or • LIN13 where LIN02 ... LIN012 = B8 (batch) or LT (lot)	• LIN03 or • LIN05 or • LIN07 or • LIN09 or • LIN11 or • LIN13 where LIN02 ... LIN012 = NB (batch) or BB (lot)
customerLotNumber	string	Customer lot number. Added for Pierre Fabre.	–	–	–
subLotNumber	string	Sublot number.	–	–	–
lotStockType	string	Indicates if the batch is under QA, in which case the expected availability date should match the QM lot planned end date. Pass through mapping.	YMDRM02/Z_STOCK_TYPE	–	–
restrictedUseStatus	string	Lot or batch in restricted use status flag. Pass thru mapping. Valid values: • X - lot in restricted use status null, missing, • Empty tag - Lot NOT in restricted use status (default)	YMDRM02/Z_BATCH_STATUS	–	–
expirationDate	string	Expiration date for lot.	YMDRM02/Z_EXPIRATION_DATE	DTM02 where DTM01 = 036	DTM010-020 where DTM01-010 = 36
epochExpirationDate	integer	Expiration date for lot generated in epoch integer date format. Each string date recorded will have an epoch date format generated by a java exit.	–	–	–
expiryBucket	integer	Number of days within which drugs are going to be expired.	–	DTM06=Days for Drugs Expiry	–
itemStatus	string	The requested status.	–	–	–
itemCategory	string	Groups similar articles together (e.g. product group). It has no effect on functionality, but can be used to link articles for queries or reports if necessary.	–	–	–
itemType	string	Determines the type of articles declared on this order. Item Type links similar articles allowing processes or information for specific groups of articles.	–	–	–
isStoreSensitiveGoods	boolean	This flag indicates, whether the article needs to be stored in an area for sensitive goods. This is considered in the put-away process. Valid values: • 0 - No • 1 - Yes	–	–	–
isTemperatureControlled	boolean	This flag indicates, whether the article needs to be stored in a temperature controlled area. This is considered in the put-away process. Valid values: • 0 - No • 1 - Yes	–	–	–

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping (standard & customized IDoc)	X12 mapping	EDIFACT mapping
isCaptureSerialNumbers	boolean	This flag indicates whether an article requires serial number capturing. This flag is considered in the RDT-based outbound processes. Valid values: • 0 - No • 1 - Yes	–	–	–
stockRotationMethod	string	Determines the sequence in which the stock is allocated in outbound processes. This field is linked to the following "StockRotationBasis" field.	–	–	–
stockRotationBasis	string	In conjunction with the previous "StockRotationMethod" field, this determines the basis upon which the stock will be selected and allocated in the outbound process.	–	–	–
pricingUnitNumerator	string	Specifies how many units of the articles the article value applies to. For Example, if articles are only valued in packs of 10 then the value for 10 articles would be entered in ArticleValue and then value10 would be entered here.	–	–	–
commodityExportTariffCode	string	Export commodity code of the article. Used to derive a customs tariff when the goods are exported.	–	–	–
commodityImportTariffCode	string	Import commodity code of the article. Used to derive a customs tariff when the goods are imported.	–	–	–
recipeCode	string	Free-text optional field. It often appears on shipment documentation.	–	–	–
smallNormalFlag	string	This value indicates the type of RDT-workbank-tasks, required for picking/packing. 2 needs to be set if SAP-tasks are to be generated. Valid values: • NORMAL - 1 • SMALL - 2	–	–	–
serialNumberCountBySKU	number	Number of serial stickers needed per SKU box.	–	–	–
warehouseLocation	string	Warehouse location.	–	–	–
warehouseZone	string	Warehouse work zone.	–	–	–
additionalWarehouseZoneInformation	string	More zoning data.	–	–	–
inventoryStatusUpdateCode	string	Stock Lock code for the transaction state. Codes in Appendix.	–	–	–
receiptDate	string	Date stock received.	–	–	–
itemQuantity	array	Total inventory quantity at line item level.	Segment E1DRM02	–	–
quantityType	string	Type of quantity recorded. Valid value: • ITEM • ONHAND • PICKED	quantityType = ITEM	–	–
quantityValue	number	Quantity of product reported in inventory.	STQTY	–	–
unitOfMeasure	string	Unit or Basis for Measurement Code. Quantity UOM's include: • CA - Case • CP - Crate • CT - Carton • DS - Display • DZ - Dozen • EA - Each • PC - Piece • PF - Pallet • PK - Package • T3 - Thousand Pieces • TH - Thousand • TY - Tray • UN - Unit	MEASUNIT	–	–
additionalProductReferencesFlag	–	Flags that determine if additional product references and dates are mandatory.	–	–	–
productReferenceFlag	string	Determines whether an Article Reference is mandatory when a goods item is created, altered or received.	–	–	–
productDateFlag	string	Determines whether Article Date 1 is mandatory when a goods item is created, altered or received.	–	–	–
additionalProductReferencesFlag2	–	Flags that determine if additional product references and dates are mandatory.	–	–	–
productReferenceFlag	string	Determines whether an Article Reference is mandatory when a goods item is created, altered or received.	–	–	–

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping (standard & customized IDoc)	X12 mapping	EDIFACT mapping
productDateFlag	string	Determines whether Article Date 1 is mandatory when a goods item is created, altered or received.	–	–	–
additionalProductReferencesFlag3	–	Flags that determine if additional product references and dates are mandatory.	–	–	–
productReferenceFlag	string	Determines whether an Article Reference is mandatory when a goods item is created, altered or received.	–	–	–
productDateFlag	string	Determines whether Article Date 1 is mandatory when a goods item is created, altered or received.	–	–	–
totalItemPriceValue	–	Total price value for items reported in inventory balance.	–	–	–
priceValue	number	Price amount value.	–	–	–
currencyCode	string	Currency code for price value.	–	–	–
storageLocationQuantity	array	Storage location quantity.	Segment E1DRM02/E1DRM04	Segment LIN/QTY	Segment LIN/INV/QTY/LOC
storageLocation	string	Storage location.	STOCK_LOC	–	LOC020-010 where LOC01-010 = 14
quantityType	string	Type of quantity recorded. Valid values include: • ONHAND • ONHOLD • ONORDER • INSPECTION • ALLOCATED • PICKED • TOTAL Use Java call-out to table CL_QuantityType.txt	quantityType = ONHAND E1DRM04/QUALF = "079" Use Java call-out to table CL_QuantityType.txt	• quantityType = ONHAND (QTY01 = "17") OR • quantityType = ONHOLD (QTY01 = "QH") OR • quantityType = ONORDER (QTY01 = "63") OR • quantityType = ALLOCATED (QTY01 = "H8") OR • quantityType = TOTAL (QTY01 = "26") Use Java call-out to table CL_QuantityType.txt	• quantityType = ONHAND (QTY010 = "17") OR • quantityType = ONHOLD (QTY010 = "27") OR • quantityType = ONORDER (QTY010 = "21") OR • quantityType = INSPECTION (QTY010 = "89") OR • quantityType = ALLOCATED (QTY01 = "170") Use Java call-out to table CL_QuantityType.txt
quantityValue	number	Quantity of product reported in inventory.	E1DRM04/QTY where quantityType = ONHAND	QTY02	QTY020
unitOfMeasure	string	Unit or Basis for Measurement Code. Quantity UOM's include: • CA - Case • CP - Crate • CT - Carton • DS - Display • DZ - Dozen • EA - Each • PC - Piece • PF - Pallet • PK - Package • T3 - Thousand Pieces • TH - Thousand • TY - Tray • UN - Unit Use Java call-out to table UOM_TableLookup2.txt	E1DRM04/UOM where quantityType = ONHAND Use Java call-out to table UOM_TableLookup2.txt	QTTY03-01 Use Java call-out to table UOM_TableLookup2.txt	QTTY030 Use Java call-out to table UOM_TableLookup2.txt
inventoryType	string	Code specifying a type of inventory. Relevant to EDIFACT.	–	–	INV020 Note: check value of isIncrease to determine if value is added to or subtracted from inventory. Use Java call-out to lookup table CL_InventoryType.txt
isIncrease	boolean	Whether the transaction added or subtracted inventory levels, null if stock levels remain static. Either a '+' for adding stock or a '-' to subtract stock. Null if there is no change in the qty. • True - Add stock • False - Subtract stock.	–	–	INV010 If isIncrease = true, INV010 = 2 (movement into inventory) Else, INV010 = 1 (movement out of inventory)
inventoryStatusUpdateCode	string	Code identifying the inventory update request or movement type Use Java call-out to lookup table CL_StatusUpdateCode.	–	–	INV030 Note: check value of isIncrease to determine if value is added to or subtracted from inventory. Use Java call-out to lookup table CL_StatusUpdateCode
shippingReceivingPoint	string	Identifier code for the depot or site where the inventory is held, normally the warehouse to which the goods will be received.	–	–	–
isBonded	boolean	Specifies whether stock declared on order line is bonded or not. Valid values: • 0 - No • 1 - Yes	–	–	–
isTaxed	boolean	Specifies whether stock declared on order line is taxed or not. Valid values: • 0 - No • 1 - Yes	–	–	–
conversionFactor	string	The UOM conversion factor to apply to the client UOM.	–	–	–
inventoryQuantities	array	Inventory quantities.	–	–	Segment loop group 12 INV/QTY

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping (standard & customized IDoc)	X12 mapping	EDIFACT mapping
quantityType	string	Type of quantity recorded. Valid values include: • INVENTORYADJUSTMENT • INVENTORYDISCREPANCY • INVENTORYMOVEMENT • INVENTORYWITHDRAWAL • ONHAND • ONHOLD • ONORDER	–	–	QTY010-010 Use Java call-out to table CL_QuantityType.txt
quantityValue	string	Additional product reference or attribute date value in date format YYYY-MM-DD.	–	–	QTY010-020 segment group 12 Need a quantity type in QTY010-010: 501 = qty credited 502 = qty debited INV010 identifies movement into (1) or out of (2) inventory
unitOfMeasure	string	Unit or Basis for Measurement Code. Quantity UOM's include: • CA - Case • CP - Crate • CT - Carton • DS - Display • DZ - Dozen • EA - Each • PC - Piece • PF - Pallet • PK - Package • T3 - Thousand Pieces • TH - Thousand • TY - Tray • UN - Unit Use Java call-out to table UOM_TableLookup2.txt	–	–	QTY010-030 Use Java call-out to table UOM_TableLookup2.txt
additionalProductReferences	–	Additional product references instance 1.	–	–	–
additionalProductReference	string	Additional product reference or attribute. Pass thru mapping.	–	–	–
productDate	string	Additional product reference or attribute date value in date format YYYY-MM-DD.	–	–	–
epochProductDate	long	Additional product reference or attribute date value in epoch date format.	–	–	–
additionalProductReferences2	–	Additional product references instance 2.	–	–	–
additionalProductReference	string	Additional product reference or attribute. Pass thru mapping.	–	–	–
productDate	string	Additional product reference or attribute date value in date format YYYY-MM-DD.	–	–	–
epochProductDate	long	Additional product reference or attribute date value in epoch date format.	–	–	–
additionalProductReferences3	–	Additional product references instance 3.	–	–	–
additionalProductReference	string	Additional product reference or attribute. Pass thru mapping.	–	–	–
productDate	string	Additional product reference or attribute date value in date format YYYY-MM-DD.	–	–	–
epochProductDate	long	Additional product reference or attribute date value in epoch date format.	–	–	–
productItemDetails	array	Details about the product, container, weight, volume, value, quantity, and other key information at a subline level.	–	–	–
itemCode	string	Unique identification of goods item.	–	–	–
consignmentIdentifier	string	Identification of the consignment to which the stock belongs.	–	–	–
productCodeValue	string	Product code value.	–	–	–
caseIdentifier	string	Case identifier number.	–	–	–
masterCaseIdentifier	string	Master case identifier.	–	–	–
packageType	boolean	Package type used for this PackKey Level, eg, BOX. Appendix includes full list of qualifiers.	–	–	–
conditionStatus	boolean	Status of the product item.	–	–	–
isBonded	string	Specifies whether stock declared on order line is bonded or not. Valid values: • `True` - 1 = Product is bonded • `False` - 0 = Product is not bonded (default)	–	–	–

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping (standard & customized IDoc)	X12 mapping	EDIFACT mapping
isTaxed	string	Specifies whether stock declared on order line is taxed or not. Valid values: • `True` - 1 = Product is taxed • `False` - 0 = Product is not taxed (default)	–	–	–
warehouseIdentifier	string	Warehouse / warehouse complex identifier. Identification of warehousing client that is used throughout the system.	–	–	–
countryOfOrigin	string	Country from which shipment was shipped.	–	–	–
lotNumber	string	Lot number at product item details level.	–	–	–
actualCaseWeight	–	Actual weight of case at the product details level.	–	–	–
weightType	string	Type of weight recorded. Valid value is CASE.	–	–	–
weightValue	number	Weight of the case as the product item details level.	–	–	–
unitOfMeasure	string	Weight unit of measure in master data for product.	–	–	–
actualCaseVolume	–	Actual case volume at the product details level.	–	–	–
volumeType	string	Type of weight recorded. Valid value is CASE.	–	–	–
volumeValue	number	Volume for item.	–	–	–
unitOfMeasure	string	Volume unit of measure.	–	–	–
itemQuantity	–	Order quantity at the product item details level.	–	–	–
quantityType	string	Type of quantity recorded. Valid value is `ITEM`.	–	–	–
quantityValue	number	Quantity of product in delivery.	–	–	–
unitOfMeasure	string	Unit of measure for delivery quantity.	–	–	–
totalItemPriceValue	–	Segment recording total value of goods in message.	–	–	–
priceValue	number	Total price value for items ordered in message.	–	–	–
currencyCode	string	Currency code for price value.	–	–	–
packagingDimensions	–	Segment recording total value of goods in message.	–	–	–
packagingLength	number	Packaging length.	–	–	–
packagingWidth	number	Packaging width.	–	–	–
packagingHeight	number	Packaging height.	–	–	–
unitOfMeasure	string	Volume unit of measure.	–	–	–
mpcDates	–	Arrival Date and Time of received goods (inbound) in warehouse.	–	–	–
dateType	string	Date type. Valid value is `ARRIVAL`.	–	–	–
dateValue	string	Date defined by referenced date type in datetime format YYYY-MM-DDTHH:MM:SS:nnZ.	–	–	–
timeZone	string	Time zone for referenced dates.	–	–	–
unitPricing	array	Unit price amounts for item.	–	Segment LIN/CTP	Segment LIN/INV/PRI
pricingType	string	Pricing type for price sales catalog. Valid values include: • `CATALOG` - Catalog price • `UNIT` - Unit price • `CHANGED` - Price change • `CONTRACT` - Contract price • `DISCOUNT` - Discounted price • `PROMOTION` - Promotion price Use Java call-out to lookup table CL_PricingType.txt.	–	CTP02 = • UCP (unit price) or • CAT or • CHG or • CON or • DPR or • PRP Use Java call-out to lookup table CL_PricingType.txt.txt	PRI010-010 = "AAF" (catalog price) PRI010-030 = • CA (Catalog) or • CT (Contract) or • PE (Each, unit)
totalStockValue	integer	Total pricing amount for that line item.	–	CTP08 (Total Monetary Amount)	–
currencyCode	string	Currency for pricing amount.	–	CUR02, refer enumerations for CUR01	–
transactionReferenceIdentifierList	–	Transaction references for purchase order number used in the custom IDoc segment.	–	Segment LIN/QTY/REF	Segment LIN/INV/QTY/LOC/RFF

b2bTransactionIdentifierType	string	Referenced document type.
		Valid values include:
		•
		`PURCHASEORDER`
		•
		`SUPPLIERORDER`
		•
		`DOCUMENT`
		•
		`WORKORDER`
		Use Java call-out to lookup table CL_B2BTransactionIdentifierType.txt.
			–	N901 =
				•
				PO (PURCHASEORDER) or
				•
				WO (WORKORDER) or
				•
				DO (DELIVERY) or
				•
				RSN (RESERVATION)
				Use Java call-out to lookup table CL_B2BTransactionIdentifierType.txt
					RFF010-010 =
					•
					OP (PURCHASEORDER) or
					•
					ADV (WORKORDER) or
					•
					AAK (DELIVERY) or
					•
					AEO (RESERVATION)
					Use Java call-out to lookup table CL_B2BTransactionIdentifierType.txt

b2bTransactionIdentifierValue	string	Purchase order number for lotAttributes.	–	–	–
transactionDate	string	Date transaction created.	–	–	–
epochTransactionDate	integer	Date transaction created in epoch integer date format. Each string date recorded will have an epoch date format generated by a java exit.	–	–	–
mpcDates	array	Dates relevant for the batch master.	Segment E1DRM02	Segment LIN/QTY/DTM	Segment LIN/INV/QTY/LOC/DTM

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping (standard & customized IDoc)	X12 mapping	EDIFACT mapping
dateType	string	Date type at header level for inventory balance in date format YYYYMMDD and time format (IDoc, X12) HHMMSS. Valid values: • AVAILABILITY • EFFECTIVE Use Java call-out to lookup table CL_DateType.txt	–	DTM01 = • 007 (effective) or • 036 (expiration) or • 018 (availability) or • 405 (manufacturing) Use Java call-out to lookup table CL_DateType.txt	DTM010-010 = 36 (expiration) or 44 (availability) Use Java call-out to lookup table CL_DateType.txt
dateValue	string	Date defined by referenced date type in date format YYYY-MM-DD. Dates will be populated with string of 0's if no date found for any of these date types. Maps will pass thru whatever it finds in IDoc date and time fields.	EFFECTIVE: E1DRM02/EFF_DATE AVAILABILITY: YMDRM02/Z_AVAIL_DATE	DTM02	DTM010-020
epochDateValue	integer	Date defined by referenced date type generated in epoch integer date format. Each string date recorded will have an epoch date format generated by a java exit.	–	–	–
timeZone	string	Time zone for referenced dates.	–	–	–
destinationLocationQuantity	array	Destination location quantity. Maps to X12 SDQ segment. Records quantities by location identifiers. Multiple locations-quantity pairs recorded on each line.	–	Note: CVS uses SDQ in their 846 as optional, linking CVS store numbers to quantities for each store.	–
unitOfMeasure	string	Unit or Basis for Measurement Code. Quantity UOM's include: • CA - Case • CP - Crate • CT - Carton • DS - Display • DZ - Dozen • EA - Each • PC - Piece • PF - Pallet • PK - Package • T3 - Thousand Pieces • TH - Thousand • TY - Tray • UN - Unit Use Java call-out to lookup table CL_DateType.txt	–	SDQ01	–
companyIdentifierType	string	Company identifier type for locations identified within group.	–	SDQ02	–
locationQuantitySequence01	–	Location quantiity group 1 with attributes to match SDQ location/quantity pairs.	–	–	–
locationIdentifier	string	Location identifier.	–	SDQ03	–
destinationQuantity	integer	Quantity of product reported for location.	–	SDQ04	–
locationQuantitySequence02	–	Location quantiity group 1 with attributes to match SDQ location/quantity pairs.	–	–	–
locationIdentifier	string	Location identifier.	–	SDQ05	–
destinationQuantity	integer	Quantity of product reported for location.	–	SDQ06	–
locationQuantitySequence03	–	Location quantiity group 1 with attributes to match SDQ location/quantity pairs.	–	–	–
locationIdentifier	string	Location identifier.	–	SDQ07	–
destinationQuantity	integer	Quantity of product reported for location.	–	SDQ08	–
locationQuantitySequence04	–	Location quantiity group 1 with attributes to match SDQ location/quantity pairs.	–	–	–
locationIdentifier	string	Location identifier.	–	SDQ09	–
destinationQuantity	integer	Quantity of product reported for location.	–	SDQ10	–
locationQuantitySequence05	–	Location quantiity group 1 with attributes to match SDQ location/quantity pairs.	–	–	–
locationIdentifier	string	Location identifier.	–	SDQ11	–
destinationQuantity	integer	Quantity of product reported for location.	–	SDQ12	–
locationQuantitySequence06	–	Location quantiity group 1 with attributes to match SDQ location/quantity pairs.	–	–	–
locationIdentifier	string	Location identifier.	–	SDQ13	–
destinationQuantity	integer	Quantity of product reported for location.	–	SDQ14	–
locationQuantitySequence07	–	Location quantiity group 1 with attributes to match SDQ location/quantity pairs.	–	–	–
locationIdentifier	string	Location identifier.	–	SDQ15	–
destinationQuantity	integer	Quantity of product reported for location.	–	SDQ16	–
locationQuantitySequence08	–	Location quantiity group 1 with attributes to match SDQ location/quantity pairs.	–	–	–
locationIdentifier	string	Location identifier.	–	SDQ17	–

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping (standard & customized IDoc)	X12 mapping	EDIFACT mapping
destinationQuantity	integer	Quantity of product reported for location.	–	SDQ18	–
locationQuantitySequence09		Location quantiity group 1 with attributes to match SDQ location/quantity pairs.	–	–	–
locationIdentifier	string	Location identifier.	–	SDQ19	–
destinationQuantity	integer	Quantity of product reported for location.	–	SDQ20	–
locationQuantitySequence10	–	Location quantiity group 1 with attributes to match SDQ location/quantity pairs.	–	–	–
locationIdentifier	string	Location identifier.	–	SDQ21	–
destinationQuantity	integer	Quantity of product reported for location.	–	SDQ22	–
freeTextNote	array		–	–	FTX
textReferenceCode	string	Text reference or subject code. Will map whatever is found.	–	–	FTX010
textFunctionCode	string	Text function code. Will map whatever is found.	–	–	FTX020
language	string	Language of free text.	–	–	FTX050
textFormatCode	string	Format code for free text.	–	–	
freeText	string	Free test string.	–	–	FTX040-010
customFields	array	Name value pairs for mapping pass through elements that will not be processed in TL system at different level.	–	–	–
name	string	Name of mapped field.	–	–	–
value	string	Value in named field.	–	–	–
canonicalInventoryUpdateConfirmationSummary	array	Inventory balance confirmation summary.	–	Segment CTT	–
summaryType	string	Type of summary. Valid values include: • NUMBEROFITEMS • LINEITEMTOTAL • TOTALAMOUNT	–	Summary Types in PO: • NUMBEROFITEMS • LINEITEMTOTAL • TOTALAMOUNT	–
summaryValue	string	Value in summary.	–	CTT01 for NUMBEROFITEMS AMT02 for LINEITEMTOTAL when AMT01 = 1 AMT02 for TOTALAMOUNT when AMT01 = TT	–
summaryValueUnit	string	Unit of measure or currency or other unit for summary value.	–	–	–

© TraceLink Inc. 2009 - 2025 All Rights Reserved