TRACELINK UNIVERSITY

Inventory update request canonical guidelines

An Inventory Update Request is an electronic transaction that instructs changes to the status, usability, or location of stock within the supply chain. Sent by a supplier or inventory owner to a third-party logistics provider (3PL), it ensures product handling remains aligned with operational needs and regulatory requirements. Inventory Update Request is used to change stock disposition (such as from available to quarantined), direct the movement of goods within or between warehouses, or update handling instructions for specific cargo or handling units. By keeping inventory status and movement instructions synchronized across partners and systems, the Inventory Update Request helps prevent delays, supports cold chain and controlled substance management, and reduces the risk of errors. This real-time visibility enables better planning, quicker response to issues, and smoother coordination between manufacturers, 3PLs, and distributors.

Canonical guidelines

Canonical Element	Type Definition	Description	IDoc mapping	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.	–	–	–
canonicalInventoryUpdateRequestHeader	–	Inventory update request message header.	IDoc header = E1BP2017_GM_HEAD_01	Segment W15 (N9)	Segment BGM

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping	X12 mapping	EDIFACT mapping
processingFunctionTypeCode	string	Processing function, transaction set purpose code. Valid values include: • RESPONSE • CREATE • CHANGE • CANCEL • ORIGINAL • REPLACE • STATUS • INFORMATION • PROPOSED Use Java call-out table CL_ProcessingFunctionTypeCode.txt		W1504 Use Java call-out table CL_ProcessingFunctionTypeCode.txt	BGM030 Use Java call-out table CL_ProcessingFunctionTypeCode.txt
messageTypeIdentifierCode	string	Code identifying purpose of EDI message.	–	–	BGM010-040
externalERPUserName	string	External ERP user name if relevant. Used by SAP to post inventory update request.	PR_UNAME	Can be mapped to N902 where N901 = JD	N/A
warehouseIdentifier	string	Warehouse id.	–	–	–
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 request at header level for TL transaction identifier type INVENTORYUPDATEREQUEST mapping.	segment E1BP2017_GM_HEAD_01	Segment W15	Segment BGM
b2bTransactionIdentifierType	string	Referenced document type. Valid value is INVENTORYUPDATEREQUEST.	b2bTransactionIdentifierType = INVENTORYUPDATEREQUEST	W1505 = IA inventory b2bTransactionIdentifierType = INVENTORYUPDATEREQUEST	BGM010-010 = 78(inventory movement advice) b2bTransactionIdentifierType = INVENTORYUPDATEREQUEST
b2bTransactionIdentifierValue	string	Document identifier for transaction referenced by b2bTransactionIdentifierType.	INVENTORYUPDATEREQUEST mapping REF_DOC_NO	W1502 (required) b2bTransactionIdentifierType = INVENTORYUPDATEREQUEST	BGM020-010 where BGM010-10 = 78 b2bTransactionIdentifierType = INVENTORYUPDATEREQUEST
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.	–	–	–
transactionReferenceIdentifierList(2)	array	Instance 2 transaction references for inventory update request at header level for all other document types.	segment E1BP2017_GM_HEAD_01	Segment REF	Segment loop 1 RFF
b2bTransactionIdentifierType	string	Referenced document type. Valid values include: • DOCUMENT • PURCHASEORDER • PHYSICALINVENTORYRECORD • REPLENISHMENTPO Use Java call-out to table CL_B2BTransactionIdentifierType.txt	b2bTransactionIdentifierType = BOL	REF01 b2bTransactionIdentifierType = anything Use Java call-out to table CL_B2BTransactionIdentifierType.txt	RFF010-010 b2bTransactionIdentifierType = anything Use Java call-out to table CL_B2BTransactionIdentifierType.txt
b2bTransactionIdentifierValue	string	Document identifier for transaction referenced by b2bTransactionIdentifierType.	BILL_OF_LADING	N902 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	Header dates relevant for inventory update request.	–	Segment W15 (required date) Segment G62	–
dateType	string	Date type at header level for inventory update. Valid values include: • INVENTORYREPORT • INVENTORYUPDATE • GOODSRECEIPT • DOCUMENT Use Java call-out to lookup table CL_DateType.txt.	dateType = INVENTORYUPDATEREQUEST dateType = DOCUMENT	G6201: dateType = INVENTORYUPDATEREQUEST (if not already mapped) or anything else use java call-out to lookup table CL_DateType.txt	DTM010-010 = • 366 (INVENTORYUPDATEREQUEST) or • 137 (DOCUMENT) 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.	• PSTNG_DATE = dateType INVENTORYUPDATEREQUEST • DOC_DATE = dateType DOCUMENT	G6202 if dateType INVENTORYUPDATEREQUEST, W1501 (required)	DTM010-020

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping	X12 mapping	EDIFACT mapping
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.	–	–	–
customerBusinessInformation	–	Customer party company identifiers and address information.	–	Segment loop 0100 N1 N101 = "BY"	Segmenyt loop 2 NAD NAD010 = "BY"
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.	–	–	–
companyIdentifierType	string	Company identifier type for business partner.	–	N103 Use Java call-out to lookup table CL_CompanyIdentifierType.txt	NAD020-030 Use Java call-out to lookup table CL_CompanyIdentifierType.txt
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	–	–	–
houseNumber	string	House number	–	–	–
township	string	Township	–	–	–
premisesCode	string	Premises code	–	–	–
companyLocationContactInformation	–	–	–	Hard code PER01 to "AJ" primary contact	Segment loop 4 CTA/COM
contactType	string	Contact type. Valid values: • CUSTOMER • DELIVERY • ORDERCONTACT • PRIMARY • PRODUCTMANAGER • RESPONSIBLE • SUPPLIER Use Java call-out to table CL_ContactType.txt.	–	PER01 = • AJ (PRIMARY) or • OC (ORDERCONTACT) or • PM (PRODUCTMANAGER)	Hard code CTA010 to • RP (PRIMARY) or • OC (ORDERCONTACT) or • PM (PRODUCTMANAGER) or • BS (CUSTOMER) or • DL (DELIVERY) or • RP (RESPONSIBLE) or • SU (SUPPLIER) Use Java call-out to table CL_ContactType.txt
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 loop 0100 N1 N101 = "SU"	Segmenyt loop 2 NAD NAD010 = "SU"
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	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	–	–	–
houseNumber	string	House number	–	–	–

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping	X12 mapping	EDIFACT mapping
township	string	Township	–	–	–
premisesCode	string	Premises code	–	–	–
companyLocationContactInformation	–	–	–	Hard code PER01 to "AJ" primary contact	Hard code CTA010 to "RP" responsible person
contactType	string	Contact type. Valid values: • CUSTOMER • DELIVERY • ORDERCONTACT • PRIMARY • PRODUCTMANAGER • RESPONSIBLE • SUPPLIER Use Java call-out to table CL_ContactType.txt	–	PER01 = • AJ (PRIMARY) or • OC (ORDERCONTACT) or • PM (PRODUCTMANAGER) or • DC (CUSTOMER) or • DC (DELIVERY) or • RP (RESPONSIBLE) or • SU (SUPPLIER) Use Java call-out to table CL_ContactType.txt	Hard code CTA010 to • RP (PRIMARY) or • OC (ORDERCONTACT) or • PM (PRODUCTMANAGER) or • BS (CUSTOMER) or • DL (DELIVERY) or • RP (RESPONSIBLE) or • SU (SUPPLIER) Use Java call-out to table CL_ContactType.txt
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.	–	Segment loop 0100 N1 N101 = "ST"	Segmenyt loop 2 NAD NAD010 = "ST"
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.	–	–	–
companyIdentifierType	string	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	–	–	–
houseNumber	string	House number	–	–	–
township	string	Township	–	–	–
premisesCode	string	Premises code	–	–	–
companyLocationContactInformation	–	–	–	Hard code PER01 to "AJ" primary contact	Hard code CTA010 to "RP" responsible person
contactType	string	Contact type. Valid values: • CUSTOMER • DELIVERY • ORDERCONTACT • PRIMARY • PRODUCTMANAGER • RESPONSIBLE • SUPPLIER Use Java call-out to table CL_ContactType.txt	–	PER01 = • AJ (PRIMARY) or • OC (ORDERCONTACT) or • PM (PRODUCTMANAGER)	Hard code CTA010 to • RP (PRIMARY) or • OC (ORDERCONTACT) or • PM (PRODUCTMANAGER) or • BS (CUSTOMER) or • DL (DELIVERY) or • RP (RESPONSIBLE) or • SU (SUPPLIER) Use Java call-out to table CL_ContactType.txt
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"
freeTextNote	array	Ship to party company identifiers and address information.	IDoc header = E1BP2017_GM_HEAD_01	Segment NTE	Segment FTX
textReferenceCode	string	Text reference or subject code. Will map whatever is found.	–	NTE01	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.	HEADER_TXT	NTE02	FTX040-010
customFields	array	Name value pairs for mapping pass through elements that will not be processed in TL system but may be passed outbound. Header level.	–	–	–
name	string	Name of mapped field.	–	–	–

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping	X12 mapping	EDIFACT mapping
value	string	Value in named field.	–	–	–
canonicalInventoryUpdateRequestItemDetails	array	Canonical inventory item details.	IDoc line item loop = Segment E1BP2017_GM_ITEM_CREATE	Segment N9 within Segment loop group 0300 W19	Segment LOC within Segment loop group 9 LIN
plantIdentifier	string	Customer plant identifier for sales forecast.	PLANT	N902 where N901 = PE	LOC020-010 where LOC010 = 19
storageLocation	string	Location in plant where stock is stored.	STGE_LOC	N902 where N901 = WS (Warrehouse Storage Location)	LOC020-010 where LOC010 = 14
productCodesIdentifiers	array	Product code identifiers list.	–	Segment loop group 0300 W19	Segment loop group 9 LIN/PIA
productCodeType	string	Code to qualify product identifier type: `CUSTOMER` `SUPPLIER` . Use Java call-out to lookup table CL_ProductCodeType.txt.	productCodeType = CUSTOMER Use Java call-out to lookup table CL_ProductCodeType.txt	• productCodeType = CUSTOMER = W1905 = BP • productCodeType = SUPPLIER = W1905 = VN Use Java call-out to lookup table CL_ProductCodeType.txt	• productCodeType = CUSTOMER = LIN030-020 = BP • productCodeType = SUPPLIER = PIA020-020 = VN Use Java call-out to lookup table CL_ProductCodeType.txt
productCodeValue	string	Product code identifier value.	MATERIAL	• CUSTOMER mapping W1906 where W1905 = BP • SUPPLIER mapping W1906 where W1905 = VN	• CUSTOMER mapping LIN030-010 where LIN030-020 = BP • SUPPLIER mapping PIA020-010 where PIA020-020 = VN
productItemInformation	array	Product item descriptions and information.	–	Segment G69	Segment IMD Set IMD010 = F (Free form)
productLanguageCode	string	Language for product description.	–	–	IMD030-060
productName	string	Name of product.	ITEM_TEXT	G6901	IMD030-040
genericName	string	Generic name for product.	–	–	–
productDescription	string	Description of the product defined by the current product identifier.	–	–	IMD030-050
lotNumber	string	Lot or batch identifier (customer).	–	–	PIA020-010 where PIA010 = 1 (hard code) and PIA020-020 = BB
customerLotNumber	string	Customer lot number.	BATCH	N902 where N901 = LT	–
supplierLotNumber	string	Supplier lot or batch identifier. For Merck's custom XML this represents the expected batch or lot number from the released process (work) order.	VENDRBATCH	W1909	–
expirationDate	string	Expiration date for lot	EXPIRYDATE	G6202 where G6201 = 36 (Expiration date)	DTM010-020where DTM010-010 = 36 (Expiration date)
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.	–	–	–
manufacturingDate	string	Manufacturing date for lot.	PROD_DATE	G6202 where G6201 = BL (Manufacturing date)	DTM010-020where DTM010-010 = 94 (Manufacturing date)
epochManfacturingDate	integer	Manufacturing date for lot generated in epoch integer date format. Each string date recorded will have an epoch date format generated by a java exit.	–	–	–
customerIdentifier	string	Customer identifier.	CUSTOMER	N104 where N101 = BY (from header)	NAD020-010 where NAD010 = BY (from header)
supplierIdentifier	string	Supplier identifier.	VENDOR	N104 where N101 = SU (from header)	NAD020-010 where NAD010 = SU (from header)
goodsRecipientIdentifier	string	Goods recipient/ship to party location identifier.	GR_RCPT	N104 where N101 = ST (from header or line item)	NAD020-010 where NAD010 = ST (from header or line item)
productionMaterialState	string	Material type. Valid values: • `API` - Active pharmaceutical ingredient • `WIP` - Work in process • `COMPONENT` - Component material • `FINISHEDGOODS` • `RAWMATERIAL`	–	–	–
componentIdentifier	string	Component product identifier. Used in custom maps when components are being consumed after production.	–	–	–
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	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. Linked to the stockRotationBasis field.	–	–	–
stockRotationBasis	string	Determines the basis, with stockRotationMethod, on which stock will be selected and allocated in an outbound process.	–	–	–
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.	–	–	–
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 whould be entered in ArticleValue and then value10 would be entered here.	–	–	–
costValueOfStock	string	The cost value of the stock	–	–	–
isQuantityReturnToStock	boolean	Specifies for qty return stock (Y= qty return to stock, N= no qty return to stock). Valid values: • True - Y = Quantity return to stock • False - N = No quantity return to stock	–	–	–
customerPlantIdentifier	string	Client Plant code (for Recsup).	–	–	–
packagingIdentifier	string	External packaging container identifier for parent container.	–	–	–
carrierTrackingIdentifier	string	Carrier Tracking Number.	–	–	–
containerIdentifier	string	Identifier of the container.	–	–	–
containerType	string	The Type of container.	–	–	–
containerProductCode	string	GTIN of the container.	–	–	–
warehouseManagementSystemInformation	–	–	–	–	–
statusCode	string	Alloga's local WMS status code.	–	–	–
reasonCode	string	Alloga's localWMS reason code.	–	–	–
transactionCode	string	Alloga's local WMS Transaction code.	–	–	–
storageLocation	string	Alloga's local WMS Storage Location Code.	–	–	–
totalItemPriceValue	–	–	–	–	–
priceValue	number	Total price value for items ordered in message.	–	–	–
currencyCode	string	Currency code for price value.	–	–	–
mpcDates	array	Dates relevant for inventory update request line item	–	–	Segment LIN/DTM
dateType	string	Date type at header level for inventory update. Valid value is INVENTORYUPDATE.	dateType = INVENTORYUPDATE	dateType = INVENTORYUPDATE	dateType = INVENTORYUPDATE = DTM010-010 = 366
dateValue	string	Date defined by referenced date type in date format YYYY-MM-DD.	INVENTORYUPDATE mapping REF_DATE	INVENTORYUPDATE mapping G6202 where G6201=AB "AB" qualifier describes "Warehouse Inventory Stock"	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.	–	–	–
transactionReferenceIdentifierList	–	Transaction references for purchase order number used in the custom IDoc segment	–	–	Segment loop group 10 RFF
b2bTransactionIdentifierType	string	Referenced document type. Valid values include: • PURCHASEORDER • SUPPLIERORDER • DOCUMENT • WORKORDER Use Java call-out to lookup table CL_B2BTransactionIdentifierType.txt	QUALF = • 001 (PURCHASEORDER) or • 002 (SUPPLIERORDER) or • 011 (DOCUMENT) or • 012 (DELIVERY) 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

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping	X12 mapping	EDIFACT mapping
b2bTransactionIdentifierValue	string	Transaction identifier value corresponding to transaction type identified by b2bTransactionIdentifierType.	• PURCHASEORDER mapping PO_ • NUMBER WORKORDER mapping ORDERID • SUPPLIERORDER mapping SALES_ORD • RESERVATION mappiing RESERV_NO	• PURCHASEORDER mapping N902 where N901 = PO • WORKORDER mapping N902 where N901 = WO • DELIVERY mapping N902 where N901 = DO • RESERVATION mapping N902 where N901 = RSN	• PURCHASEORDER mapping RFF020 where RFF010-010 = OP • WORKORDER mapping RFF020 where RFF010-010 = AOV • DELIVERY mapping RFF020 where RFF010-010 = AAK (Despatch Advice) • RESERVATION mapping RFF020 where RFF010-010 = AEO
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	• PURCHASEORDER mapping PO_ITEM • WORKORDER mapping ORDER_ITNO • SUPPLIERORDER mapping S_ORD_ITEM • RESERVATION mappiing RES_ITEM	• PURCHASEORDER mapping N903 where N901 = PO • WORKORDER mapping N903 where N901 = WO • DELIVERY mapping N903 where N901 = DO • RESERVATION mapping N903 where N901 = RSN	• PURCHASEORDER mapping RFF030 where RFF010-010 = OP • WORKORDER mapping RFF030 where RFF010-010 = AOV • DELIVERY mapping RFF030 where RFF010-010 = AAK (Despatch Advice) • RESERVATION mapping RFF030 where RFF010-010 = AEO
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. Valid values: • NO - 0 • YES - 1 • INBOUNDONLY - 2 • OUTBOUNDONLY - 3 • ITEMTYPE - 4 (Take from Item Type)	–	–	–
productDateFlag	string	Determines whether Article Date 1 is mandatory when a goods item is created, altered or received. Valid values: • NO - 0 • YES - 1 • INBOUNDONLY - 2 • OUTBOUNDONLY - 3 • ITEMTYPE - 4 (Take from Item Type)	–	–	–
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. Valid values: • NO - 0 • YES - 1 • INBOUNDONLY - 2 • OUTBOUNDONLY - 3 • ITEMTYPE - 4 (Take from Item Type)	–	–	–
productDateFlag	string	Determines whether Article Date 1 is mandatory when a goods item is created, altered or received. Valid values: • NO - 0 • YES - 1 • INBOUNDONLY - 2 • OUTBOUNDONLY - 3 • ITEMTYPE - 4 (Take from Item Type)	–	–	–
additionalProductReferencesFlag3	–	Flags that determine if additional product references and dates are mandatory.	–	–	–

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping	X12 mapping	EDIFACT mapping
productReferenceFlag	string	Determines whether an Article Reference is mandatory when a goods item is created, altered or received. Valid values: • NO - 0 • YES - 1 • INBOUNDONLY - 2 • OUTBOUNDONLY - 3 • ITEMTYPE - 4 (Take from Item Type)	–	–	–
productDateFlag	string	Determines whether Article Date 1 is mandatory when a goods item is created, altered or received. Valid values: • NO - 0 • YES - 1 • INBOUNDONLY - 2 • OUTBOUNDONLY - 3 • ITEMTYPE - 4 (Take from Item Type)	–	–	–
productItemDetails	–	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	string	Package type used for this PackKey Level, eg, BOX. Appendix includes full list of qualifiers.	–	–	–
conditionStatus	string	Status of the product item.	–	–	–
isBonded	boolean	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)	–	–	–
isTaxed	boolean	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	–	–	–
actualCaseWeight	–	Actual case weight	–	–	–
weightType	string	Type of weight recorded. Valid value is CASE.	–	–	–
weightValue	string	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	–	Quantity of item.	–	–	–
quantityType	string	Type of quantity recorded. Valid value is ITEM.	–	–	–
quantityValue	string	Quantity of item.	–	–	–
unitOfMeasure	string	Unit of measure for 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.	–	–	–
stockTransactionBeforeUpdate	–	Information about the stock before the transaction is executed.	–	–	–
shippingReceivingPoint	string	Identifier code for the depot or site where the inventory is held, normally the warehouse to which the goods will be received.	–	–	–

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping	X12 mapping	EDIFACT mapping
warehouseZone	string	Warehouse zone identifier for the from transaction state. Codes in Appendix: broken link, no codes.	–	–	–
warehouseLocation	string	Warehouse location for the from transaction state. Codes in Appendix.	–	–	–
caseIdentifier	string	Case identifier number.	–	–	–
inventoryStatusUpdateCode	string	Stock status code for the from transaction state. Codes in Appendix. Current inventory status before the update. This will not be used to update inventory.	–	–	–
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	–	–	–
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.	–	–	–
itemQuantity	–	Quantity of ORDER.	–	–	–
quantityType	string	Type of quantity recorded. Valid value is ORDER.	–	–	–
quantityValue	string	Quantity of Order.	–	–	–
unitOfMeasure	string	Unit of measure for quantity.	–	–	–
actualTareWeight	–	Actual Tare weight.	–	–	–
weightType	string	Type of weight recorded. Valid value is TARE.	–	–	–
weightValue	string	Weight of the shipment as the product item details level.	–	–	–
unitOfMeasure	string	Weight unit of measure from master data for product.	–	–	–
inventoryUpdateQuantity	–	Quantity of item scheduled in the forecast.	–	Segment W19	Segment loop group 12 INV/QTY
quantityType	string	Type of quantity recorded for inventory update. Valid values include: • INCREASE • DECREASE Use Java call-out to table CL_QuantityType.txt	–	–	–
quantityValue	number	Quantity of product reported in inventory	ENTRY_QNT	W1902 If value include - (minus) sign, inventory decreased Else inventory increased	–

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping	X12 mapping	EDIFACT mapping
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	ENTRY_UOM	W1903	–
inventoryQuantities	–	Quantity of item	–	–	Segment loop group 12 INV/QTY
quantityType	string	Type of quantity recorded. Valid value is ITEM.	–	–	QTY010-010 = 501 (INCREASE) or 502 (DECREASE) Else Use Java call-out to table CL_QuantityType.txt
quantityValue	string	Quantity of item.	–	–	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 of measure for quantity.	–	–	QTY010-030 Use Java call-out to table UOM_TableLookup2.txt
inventoryStatusUpdateCode	string	Code identifying the inventory update request or movement type Use Java call-out to lookup table CL_StatusUpdateCode.	MOVE_TYPE Use Java call-out to lookup table CL_StatusUpdateCode	W1901 W1902 (positive or negative amount) W1916 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
inventoryStatusUpdateTransactionType	string	Transaction type or document referenced for inventory update request.	MVT_IND	–	–
inventoryStatusUpdateReasonCode	string	inventory update request reason code.	MOVE_REAS	–	–
stockTypeIndicator	string	Stock type indicator. Pass thru value for SAP, if populated.	STCK_TYPE	–	–
specialStockIndicator	string	Stock type indicator. Pass thru value for SAP, if populated.	SPEC_STOCK	–	–
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
stockTransactionIdentifier	string	Unique identifier for the inventory update transaction.	–	–	–
stockTransactionType	–	Two character code to identify the type of transaction, ie, whether a change, increase or decrease. Codes in appendix appear to be incomplete.	–	–	–
shippingReceivingPoint	string	Identifier code for the depot or site where the inventory is held, normally the warehouse to which the goods will be received.	–	–	–
warehouseZone	string	Warehouse zone identifier for the from transaction state. Codes in Appendix: broken link, no codes.	–	–	–
warehouseLocation	string	Warehouse location for the from transaction state. Codes in Appendix.	–	–	–
caseIdentifier	string	Case identifier number.	–	–	–
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	–	–	–
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. Valid values • True - add stock • False - subtract stock	–	–	INV010 If isIncrease = true, INV010 = 2 (movement into inventory) Else, INV010 = 1 (movement out of inventory)

© TraceLink Inc. 2009 - 2025 All Rights Reserved

Canonical Element	Type Definition	Description	IDoc mapping	X12 mapping	EDIFACT mapping
plantIdentifier	string	Customer plant identifier for sales forecast	PLANT	N902 where N901 = PE	LOC020-010 where LOC010 = 19
storageLocation	string	Location in plant where stock is stored	STGE_LOC	N902 where N901 = WS (Warrehouse Storage Location)	LOC020-010 where LOC010 = 14
transactionReferenceIdentifierList	array	Transaction references for inventory update quantity			Segment loop group 15 RFF
b2bTransactionIdentifierType	string	Referenced document type. Valid values include: • PURCHASEORDER • SUPPLIERORDER • DOCUMENT • WORKORDER Use Java call-out to lookup table CL_B2BTransactionIdentifierType.txt	QUALF = • 001 (PURCHASEORDER) or • 002 (SUPPLIERORDER) or • 011 (DOCUMENT) or • 012 (DELIVERY) 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	Transaction identifier value corresponding to transaction type identified by b2bTransactionIdentifierType.	• PURCHASEORDER mapping PO_NUMBER • WORKORDER mapping ORDERID • SUPPLIERORDER mapping SALES_ORD • RESERVATION mappiing RESERV_NO	• PURCHASEORDER mapping N902 where N901 = PO • WORKORDER mapping N902 where N901 = WO • DELIVERY mapping N902 where N901 = DO • RESERVATION mapping N902 where N901 = RSN	• PURCHASEORDER mapping RFF020 where RFF010-010 = OP • WORKORDER mapping RFF020 where RFF010-010 = AOV • DELIVERY mapping RFF020 where RFF010-010 = AAK (Despatch Advice) • RESERVATION mapping RFF020 where RFF010-010 = AEO
transactionDate	string	Date transaction created.	–	–	DTM010-020 where DTM010-010 = 366 (INVENTORYUPDATE)
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.	• PURCHASEORDER mapping PO_ITEM • WORKORDER mapping ORDER_ITNO • SUPPLIERORDER mapping S_ORD_ITEM • RESERVATION mappiing RES_ITEM	• PURCHASEORDER mapping N903 where N901 = PO • WORKORDER mapping N903 where N901 = WO • DELIVERY mapping N903 where N901 = DO • RESERVATION mapping N903 where N901 = RSN	• PURCHASEORDER mapping RFF030 where RFF010-010 = OP • WORKORDER mapping RFF030 where RFF010-010 = AOV • DELIVERY mapping RFF030 where RFF010-010 = AAK (Despatch Advice) • RESERVATION mapping RFF030 where RFF010-010 = AEO
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.	–	–	–
freeTextNote	array	Ship to party company identifiers and address information.	–	–	Segment LIN/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.	–	–	FTTX050
textFormatCode	string	Format code for free text.	–	–	N/A
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 but may be passed outbound. Line item level.	–	–	–
name	string	Name of mapped field.	–	–	–
value	string	Value in named field.	–	–	–

© TraceLink Inc. 2009 - 2025 All Rights Reserved