1.2. Stock order by

Picking, Multi-Picking, Zone Picking

Picking Flow
Multi-Picking Flow
Zone Picking Flow

The order to take the stock is based on settings in the pick list controller. If there is a custom 'Stock order by' option, the sorting will be the value of that setting.

DEFAULT

When the setting has the value *DEFAULT*. If the setting ‘Must the user pick full pallet from bulk location’ is checked, the sorting is as follows:

- BBD
- BatchNumber
- BatchNumber2
- Priority pick location
- Full pallet
- Non-pick location
- Has LUID
- Location sequence
- LUID

Note: If the ‘Must the user pick full pallet from bulk location’ and ‘Can the user pick bulk quantity from bulk location’ checkboxes are enabled then these options forces to first take larger quantity stock from a bulk location before using pick locations.

The Bulk Pick Quantity can be modified on 'Item Master Data'.

Otherwise the sorting is done as:

- BBD
- BatchNumber
- BatchNumber2
- Priority pick location
- Pick location
- Has LUID
- Full pallet
- Location sequence
- LUID

After getting this sorting, the system will loop through all the lines it finds. If during this loop, a full pallet is found on a pick location, it is stored in a separate list, and is NOT used on the pick list line. If there is still quantity to allocate after assigning the locations to the pick list line, the stock for full

pallets on a pick location is used.

BIGGEST PALLET FIRST

When setting the value as *BIGGEST PALLET FIRST*, the system picks from the pallet where the remaining quantity will be lowest.

First the system sorts the stock as follows:

- highest quantity
- oldest LUID

After the sorting, the system will loop through all the lines.

- If the quantity on the logistic unit is higher than the quantity to pick, the system stores it in a memory list which is sorted by: lowest quantity/ oldest LUID.
- If the quantity on the logistic unit is lower or equals to the quantity to pick, the system assigns it to the picklist.

When there are still stock to pick after the system looped through every line, it will use stock from the memory list.

Examples
We have the following stock situation for the example:
Product A has 5 pallets in stock, the default quantity per pallet is 10 pcs. But we also have 1 ‘older’ larger pallets that has 12 pcs and also an ‘open’ pallet with only 4 pcs.

Product A	SSCC 001	12 pcs
Product A	SSCC 002	10 pcs
Product A	SSCC 003	10 pcs
Product A	SSCC 004	10 pcs
Product A	SSCC 005	4 pcs

Scenario 1: Ordered quantity = quantity on given SSCC
Example 1: We have a sales order for 4 pcs → system allocates SSCC 005
Example 2: We have a sales order for 10 pcs → system allocates SSCC 002 (oldest SSCC that matches the quantity)
Example 3: We have a sales order for 12 pcs → system allocates SSCC 001

Scenario 2: Smallest SSCC < Ordered quantity < quantity on given SSCC
Example 1: We have a sales order for 5 pcs → system allocates 4 pcs from SSCC 005 and 1 pcs from SSCC 001 (because this is the oldest).

Scenario 3: Smallest SSCC > Ordered quantity
Example 1: We have a sales order for 3 pcs → system allocates 3 pcs from SSCC 005 (biggest SSCC that is less than the quantity to pick)

Scenario 4: Ordered quantity > Biggest SSCC
Example 1: We have a sales order for 14 pcs → system allocates 12 pcs from SSCC 001 (biggest SSCC that is less than remaining quantity) and allocate 2 pcs from SSCC 005 (smallest remaining quantity)

Scenario 5: Ordered quantity > Biggest SSCC, but lowest quantity does not fulfill the needs
For this scenario the stock is:

Product A	SSCC 001	12 pcs
Product A	SSCC 002	10 pcs
Product A	SSCC 003	10 pcs
Product A	SSCC 004	10 pcs
Product A	SSCC 005	4 pcs
Product A	SSCC 006	1 pcs

Example 1: We have a sales order for 14 pcs → system allocates 12 pcs from SSCC 001 (biggest SSCC that is less than remaining quantity). Next SSCC 006 is used, because the quantity available is more than the remaining quantity.

Now there is still 1 piece remaining. There is no more stock to be used, so now the system will loop through the remaining stock, but smallest quantity first. This means that 1 piece of SSCC 005 will be used.

Ad hoc picking

For ad hoc picking, the stock is sorted by:

- BBD
- Batch with smallest free stock
- Pick locations
- Location with most LUID's
- Non-full pallets
- Smallest quantities per inventory line
- Location sequence

'Stock order by' query

The following tables are used in the query:

- "OITM"
- "PMX_OSSL"
- "PMX_OSWA"
- "PMX_LUID"
- "PMX_LUID" AS "MasterLUIDTable"
- "PMX_ITRI"

Subqueries with the used columns:

- "OldestSerialPerLuid"
  ("LUID", "SerialNumber")
- "InventoryDetail"
  ("InventoryQuantity", "InventoryQuantityUom2", "ItemCode", "QualityStatus", "PmxWhsCode", "LocationCode", "InternalKey", "LUID", "MasterLUID", "ItriKey")
  This subquery lists all the stock and the details needed for a 'Detail' level locking linked to

them.

- “InventoryLUID”
  (“InventoryQuantity”, “InventoryQuantityUom2”, “ItemCode”, “QualityStatus”, “PmxWhsCode”, “LUID”, “MasterLUID”, “ItriKey”)
  This subquery lists all the stock and the details needed for a ‘LUID’ level locking linked to them.
- “InventoryBatch”
  (“InventoryQuantity”, “InventoryQuantityUom2”, “ItemCode”, “QualityStatus”, “PmxWhsCode”, “ItriKey”)
  This subquery lists all the stock and the details needed for a ‘Batch’ level locking linked to them.
- “InventoryItem”
  (“InventoryQuantity”, “InventoryQuantityUom2”, „ItemCode“, “QualityStatus”, “PmxWhsCode”)
  This subquery lists all the stock and the details needed for an ‘Item’ level locking linked to them.
- “LockedItem”
  (“LockedQuantity”, “LockedQuantityUom2”, „ItemCode”, „QualityStatus“, “PmxWhsCode”)
  This subquery lists all the stock locked on ’Item/Quantity’ level with the details needed for the locking.
- “LockedBatch”
  (“LockedQuantity”, “LockedQuantityUom2”, “ItemCode”, “QualityStatus”, “PmxWhsCode”, “BatchNumber”, “BatchNumber2”, “BBD”, “ItriKey”)
  This subquery lists all the stock locked on ’Batch’ level with the details needed for the locking.
- “LockedLUID”
  (“LockedQuantity”, “LockedQuantityUom2”, “ItemCode”, “QualityStatus”, “PmxWhsCode”, “LUID”, “BatchNumber”, “BatchNumber2”, “BBD”, “ItriKey”)
  This subquery lists all the stock locked on ’LUID’ level with the details needed for the locking.
- “LockedDetail”
  (“LockedQuantity”, “LockedQuantityUom2”, “ItemCode”, “QualityStatus”, “PmxWhsCode”, “LocationCode”, “LUID”, “BatchNumber”, “BatchNumber2”, “BBD”, “ItriKey”)
  This subquery lists all the stock locked on ’Detail’ level with the details needed for the locking.

When allocating stock for the picklist, the system will count the available stock based on the following logic:
First the system counts the difference between the inventory quantity and the locked quantity for each level.

{“InventoryItem”.”InventoryQuantity“ - “LockedItem”.”LockedQuantity“}
{“InventoryBatch”.”InventoryQuantity“ - “LockedBatch”.”LockedQuantity“}
{“InventoryLUID”.”InventoryQuantity“ - “LockedLUID”.”LockedQuantity“}
{“InventoryDetail”.”InventoryQuantity“ - “LockedDetail”.”LockedQuantity“}

The lowest calculated value will be taken as the available quantity.