Contents

[1. Create Revised Version 1](#_Toc1729870986)

[2. Import Sales Quote 3](#_Toc1334381349)

[3. Shipping Instructions 4](#_Toc642588189)

[4. Picking/Shipping 5](#_Toc1813756075)

[5. Sales Order Template 6](#_Toc1884221463)

[6. Split Sales Quote 7](#_Toc163316346)

[7. Item parameters 8](#_Toc900073535)

# Create Revised Version

Need to have possibility to create copy of Sales Quote with incremented suffix -R1, -R2, etc. For that add new action on Sales Quote card (page 41) which runs custom function that based on standard copy document functionality with predefined parameters:

* Include Header = True
* Recalculate Lines = False



Graphical user interface, text, application, email

Description automatically generated

Add message to user indicating Revised version xxxx has been created.

**\*\*\*Optional subtask\*\*\***

Create automated test codeunit for this task. Add several test cases:

* Creating first copy of base document
* Creating one more copy of base document
* Creating copy of last copy.

Don’t forget to check result “No.” with expected in each case.

# Import Sales Quote

Need to have possibility to import sales lines from excel sheet to already created Sales Quote document. For that add new action on Sales Quote card (page 41). It should run processing only report that read excel file(template -“Quote Sheet Example.xlsx”) and insert new sales lines for current document. Request page should give possibility to choose file and sheet name. Besides, it should have checkbox “Create new Item”.

le. If yes – just use all values for line from table, otherwise check Boolean field from request page. If it is true then create and use new item with values from excel, if false – just skip record.

At the end show successful message with count of inserted record.

Table

Description automatically generated

**\*\*\*Optional subtask\*\*\***

Create automated test codeunit for this task. Remember, that we can’t use any additional files in test codeunits. So, you should manually fill excel buffer. Check result values in “Sales Line” table with expected values from excel or Item table. Also don’t forget to check result count of inserted records with expected counts.

# Shipping Instructions

First in this task you need add next new fields on “Ship-to Address” table and page:

* “Delivery Location” (Type - Enum)
  1. Regular Commercial
  2. Farm
  3. Airport
  4. Prison
  5. Hospital
  6. School
* “Receiving Phone” (Type - Text[20])
* “Date Needed By” (Type - Date)
* “Guaranteed” (Type - Boolean)

Values from these new custom fields should flow to **same** fields on Customer and

“Sales Header” tables after validation “Ship-to Code”. Pages “Customer Card”, “Sales Quote”, “Sales Order” should be also modified.

Then from Sales Order they should be filled on sales shipment after posting (table + page).

Also, values should flow from Sales Order to Inventory Pick and then on Posted Inventory Pick (related tables and pages should be also modified)

**\*\*\*Optional subtasks\*\*\***

1. Create TDD using draw.io application. Using specific shapes, tables and diagrams show modifications for standard tables/pages and new custom functionality.
2. Create automated test codeunit. Flow values of new fields from “Ship-to Address” to “Posted Inventory Pick”. Check expected values with result on each step.

# Picking/Shipping

In this task you need to add “Shipment Method Code” (properties like in standard on Sales Header) and “Shipment Date” (make visible if exists in standard) on Warehouse Shipment and Warehouse Pick (tables, pages). Values should firstly flow from Sales Order.

Also, you need to add automatically creation of Warehouse Shipment after Release Sales Order.

Sort records on Warehouse Picks list page by “Shipment Date” in descending order.

**\*\*\*Optional subtasks\*\*\***

1. Create TDD using draw.io application. Using specific shapes, tables and diagrams show modifications for standard tables/pages and new custom functionality.
2. Create automated test codeunit. Create documents and check expected values of these fields with actual.

# Sales Order Template

Currently in standard there are templates for creating Customer/Vendor/Item with predefined standard fields.

In this task you should in same way add template functionality for creating Sales Orders:

* Possibility to choose available template from list in process of sales order creation
* Possibility to apply template
* Possibility to save as template
* Possibility to view

Choose some standard fields from sales header for configuring predefined values.

Graphical user interface, application

Description automatically generatedGraphical user interface

Description automatically generated

**\*\*\*Optional subtasks\*\*\***

1. Create TDD using draw.io application. Using specific shapes, tables and diagrams show modifications for standard tables/pages and new custom functionality.
2. Create automated test codeunit. Create documents and check expected values of predefined fields in template with actual.

# Split Sales Quote

Currently in standard we get one document when create sales order from sales quote. In this task you should customize converting process in next way – functionality should separate quote into several orders depending on “Location Code”.

For example, we have 2 lines with “Location Code” – RED and one line with BLUE. So, after pressing “Make Order” we should get 2 documents. First with RED lines and second with BLUE.

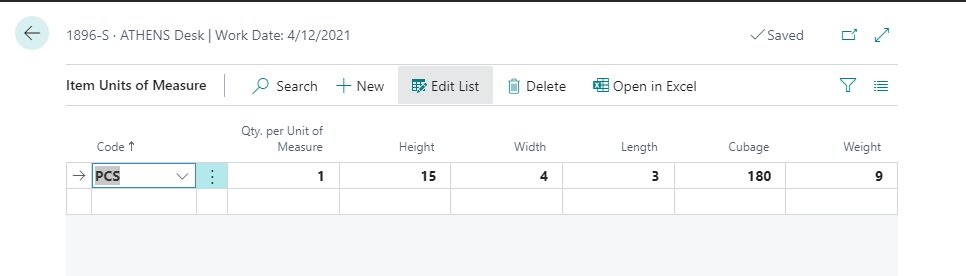
**\*\*\*Optional subtasks\*\*\***

1. Create TDD using draw.io application. Using specific shapes, tables and diagrams show modifications for standard tables/pages and new custom functionality.
2. Create automated test codeunit. Create documents and check expected values of location code and count of created orders. with actual.

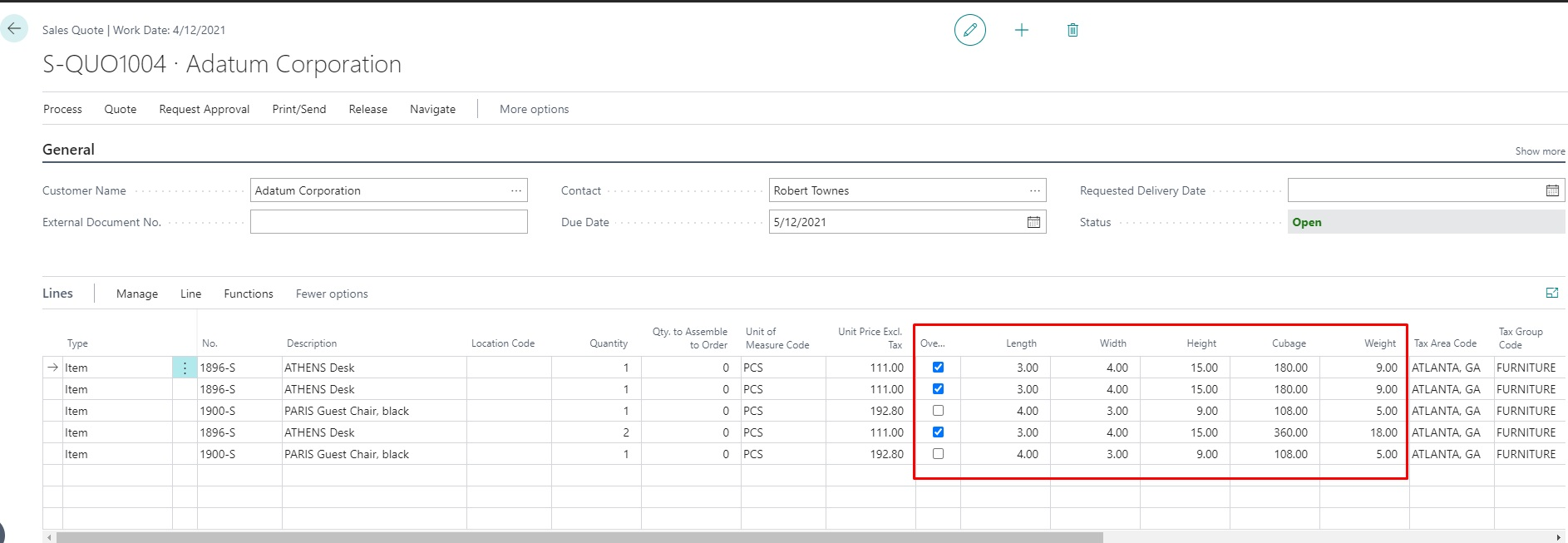
# Item parameters

In this task you will need to add on sales quote lines same fields as standard in “Item Unit of Measure”:

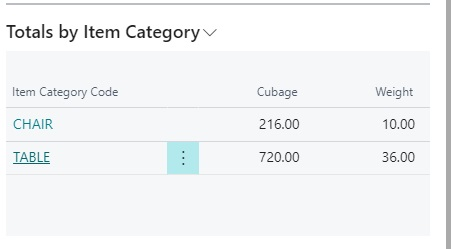
* + Length
  + Width
  + Height
  + Cubage
  + Weight



These values should be updated and recalculated after validating “No.” and “Unit of Measure Code.” based on current quantity. Also, there should be the possibility to modify them manually.



Besides, a new factbox should be added and show totals of Cubage and Weight by “Item Category”.



**\*\*\*Optional subtasks\*\*\***

1. Create TDD using draw.io application. Using specific shapes, tables and diagrams show modifications for standard tables/pages and new custom functionality.
2. Create automated test codeunit. Create document, setups and check expected result of calculations with actual.