Material Requirements Planning Report Blueprint

What is this document | Accounting Seed is a flexible accounting platform where you can add, build and layer your own customizations on top of the platform. Accounting Seed Blueprints provide an example of a specific use case that can serve as a template to implement a feature for your organization, but also can be tweaked or modified in any way you need to achieve your exact requirements. Blueprints serve as a design template or starting spot for your own feature.

accountingseed

What is the feature?

A multi-level bill of materials (multi-level BOM), sometimes referred to as an indented BOM, is a bill of materials that lists the assemblies, sub-assemblies, components, and parts required to make a product in a parent-child, top-down method. It provides a display of all items that are in parent-children relationships. A multi-level BOM is essentially a nested list whose parts or items are listed in two or more levels of detail to illustrate multiple assemblies within a product's top level BOM. Multi-level BOMs are used to organize complex and highly configurable products to ensure that all assemblies, subassemblies, components, and parts required are captured.

This feature allows all levels of the multi-level BOM to be aggregated and the material quantities required to build a specific number of products. It also provides the quantities available for manufacturing of the BOM.

When is this feature set used?

This feature set is used to report on the material required to build a specific quantity of a multi-level BOM Product. This report facilitates planning for a specific product build by identifying the total quantity of each part required to build a specific quantity of the manufactured product, the current quantities available in inventory for all parts (direct and subassembly) required to build the manufactured product, and calculating the total material required to be purchased.

What terms or definitions do I need to understand?

BOM/Bill of Material	A list of the materials with quantities used to create a manufactured product.
M-BOM / Multi-Layer BOM / Nested BOM	A list of the materials within nested subassemblies and quantities used to create a manufactured product.
Inventory Balance / Inventory Quantity Available	The quantity of a product that's physically on-site which can be shipped to a customer or used to manufacture another product.
Purchase Order	A formal communication sent to a seller of goods or services to purchase a set quantity of items with specific prices.

Can you give me an overview of how this works?

To run the Material Requirement Planning Report, the user navigates to the Product that is to be manufactured and selects the "BOM Requirements" button. This will display the Total Quantity_Required Input Page. This intermediate visual force page will be used to set the total Quantity_Required of the manufactured product to be built.

The user will then click on the run report button to execute the Material Requirement Planning Report. An asynchronous process is executed that will run through the report logic. The logic will first identify all of the Product Parts required to build the final product and then calculate the specific quantity of each Product Part required to complete the build, the current quantity available in inventory for each Product Part and the total material required to be purchased.

The report logic creates a Material Requirements record for each unique Product Part that is part of the Multi-level BOM Product (both purchased and manufactured).

Once the process has completed, a custom report will be run and displayed to the user.

What object model changes are required?

Object model changes are additional data objects or fields that you would add to Accounting Seed to achieve the customization. We suggest the following changes:

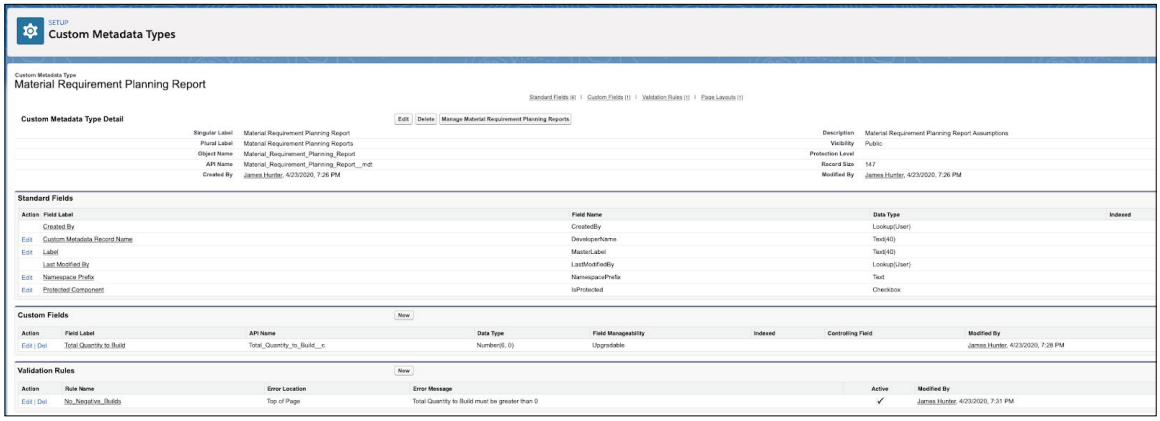
Custom Metadata Type

Custom Metadata Types are used to set Material Requirement Planning Report assumptions. Create a custom metadata type called "Material Requirement Data" and add the following custom field:

Field Label	Data Type
Total_Quantity_to_Build	Number (6,0)
Top Product	Text

In addition to the custom field, add a validation rule that ensures that the Total_Quantity_to_Build is greater than 0.

Here's a snippet of how this should look after configuration:



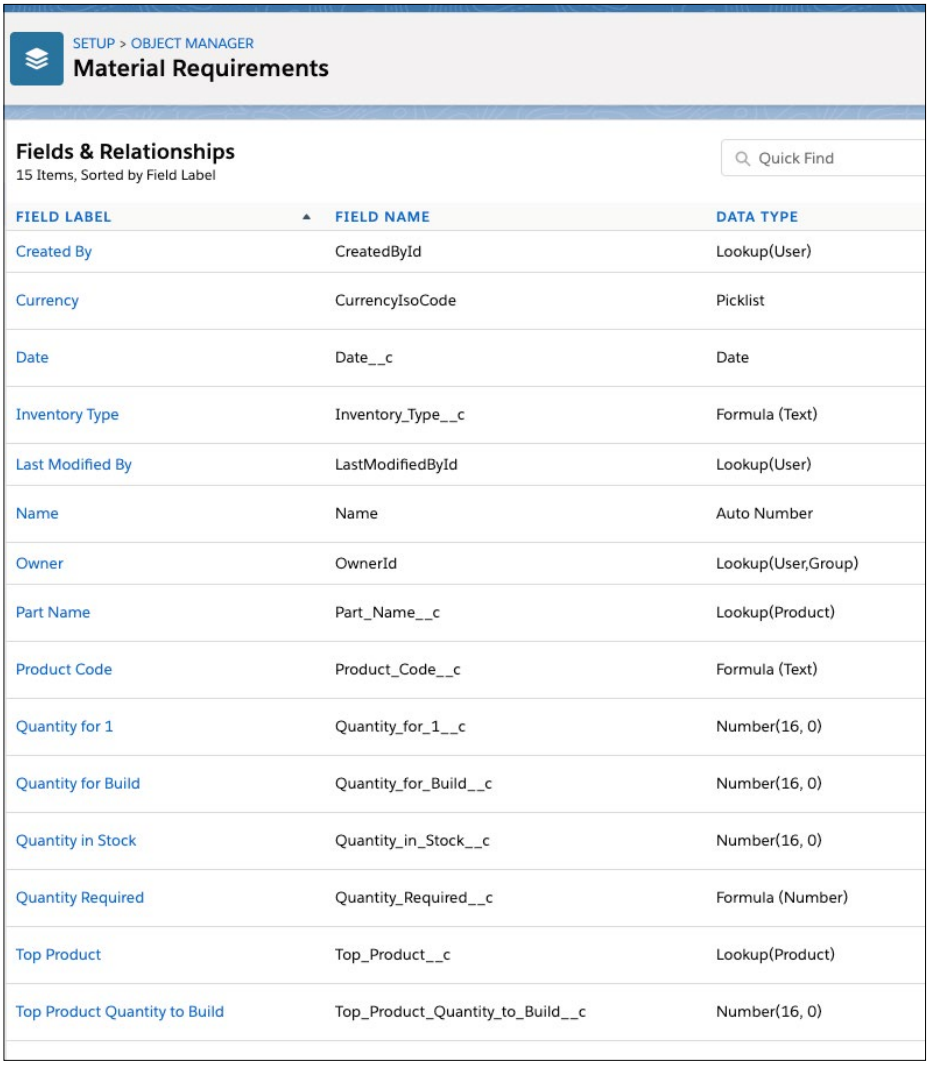
Custom Fields

Create the following custom fields on standard Accounting Seed Objects:

Object	Field Label	Data Type	Formula
Product Part	Product Code	Formula(Text)	AcctSeed__Product_Part__r.ProductCode

Custom Object

Create a Custom object called “Material Requirements” to capture Products Parts, Quantities needed to build, Inventory Quantities available with the following fields:



SETUP > OBJECT MANAGER

Material Requirements

Fields & Relationships

15 Items, Sorted by Field Label

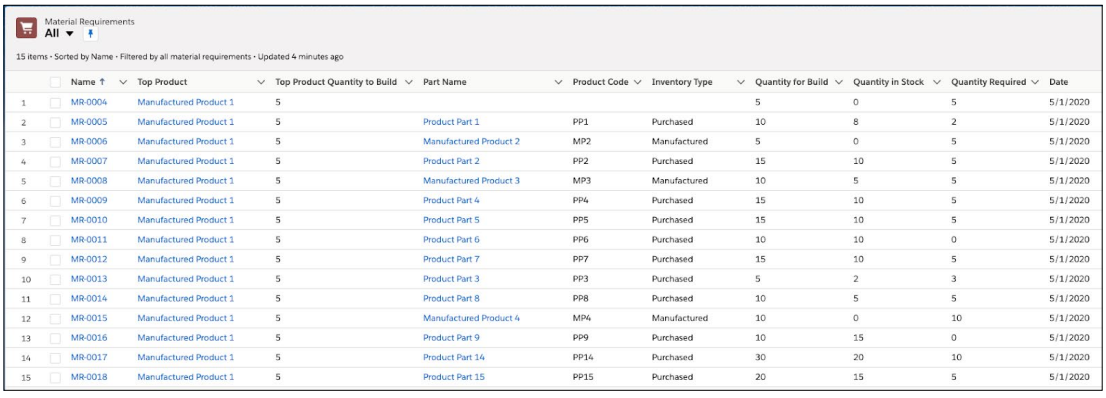
Quick Find

FIELD LABEL	FIELD NAME	DATA TYPE
Created By	CreatedById	Lookup(User)
Currency	CurrencyIsoCode	Picklist
Date	Date__c	Date
Inventory Type	Inventory_Type__c	Formula (Text)
Last Modified By	LastModifiedById	Lookup(User)
Name	Name	Auto Number
Owner	OwnerId	Lookup(User,Group)
Part Name	Part_Name__c	Lookup(Product)
Product Code	Product_Code__c	Formula (Text)
Quantity for 1	Quantity_for_1__c	Number(16, 0)
Quantity for Build	Quantity_for_Build__c	Number(16, 0)
Quantity in Stock	Quantity_in_Stock__c	Number(16, 0)
Quantity Required	Quantity_Required__c	Formula (Number)
Top Product	Top_Product__c	Lookup(Product)
Top Product Quantity to Build	Top_Product_Quantity_to_Build__c	Number(16, 0)

Custom Tab

Create a Custom Tab for the Material Requirements Object. Select the tab. Select and pin the All List View. Edit the list view adding all fields on the object to the view.

Below is a snippet of how this Tab will be displayed:



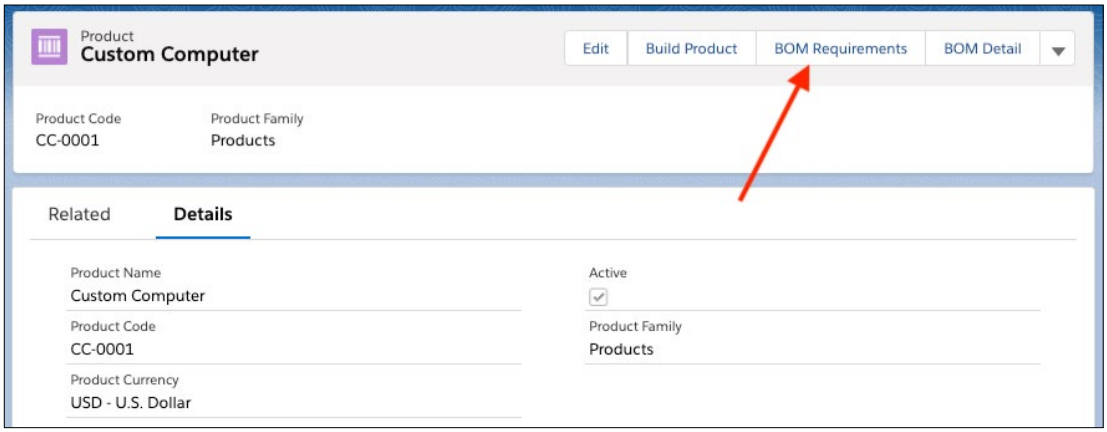
Material Requirements

All

15 items • Sorted by Name • Filtered by all material requirements • Updated 4 minutes ago

	Name	Top Product	Top Product Quantity to Build	Part Name	Product Code	Inventory Type	Quantity for Build	Quantity in Stock	Quantity Required	Date
1	MR-0004	Manufactured Product 1	5				5	0	5	5/1/2020
2	MR-0005	Manufactured Product 1	5	Product Part 1	PP1	Purchased	10	8	2	5/1/2020
3	MR-0006	Manufactured Product 1	5	Manufactured Product 2	MP2	Manufactured	5	0	5	5/1/2020
4	MR-0007	Manufactured Product 1	5	Product Part 2	PP2	Purchased	15	10	5	5/1/2020
5	MR-0008	Manufactured Product 1	5	Manufactured Product 3	MP3	Manufactured	10	5	5	5/1/2020
6	MR-0009	Manufactured Product 1	5	Product Part 4	PP4	Purchased	15	10	5	5/1/2020
7	MR-0010	Manufactured Product 1	5	Product Part 5	PP5	Purchased	15	10	5	5/1/2020
8	MR-0011	Manufactured Product 1	5	Product Part 6	PP6	Purchased	10	10	0	5/1/2020
9	MR-0012	Manufactured Product 1	5	Product Part 7	PP7	Purchased	15	10	5	5/1/2020
10	MR-0013	Manufactured Product 1	5	Product Part 3	PP3	Purchased	5	2	3	5/1/2020
11	MR-0014	Manufactured Product 1	5	Product Part 8	PP8	Purchased	10	5	5	5/1/2020
12	MR-0015	Manufactured Product 1	5	Manufactured Product 4	MP4	Manufactured	10	0	10	5/1/2020
13	MR-0016	Manufactured Product 1	5	Product Part 9	PP9	Purchased	10	15	0	5/1/2020
14	MR-0017	Manufactured Product 1	5	Product Part 14	PP14	Purchased	30	20	10	5/1/2020
15	MR-0018	Manufactured Product 1	5	Product Part 15	PP15	Purchased	20	15	5	5/1/2020

Custom Button
On the Product object create a custom button called “BOM Requirements”.



Custom Report Type
Create a custom report type called “Material Requirements” with the “Material Requirements” Object as the primary Object.

Custom Report
Create a custom report called “Material Requirement Report” utilizing the custom report type “Material Requirements” and configure as follows:

Fields to include on report:
- Top Product
- Part Name
- Product Code
- Inventory Type
- Quantity_for_Build
- Quantity_in_Stock
- Quantity_Required

Groupings:
- Group rows by Top Product

Below is a snippet of how this report will be displayed:

Report: Material Requirements
Material Requirements Report

Top Product	Part Name	Product Code	Inventory Type	Quantity for Build	Quantity in Stock	Quantity Required
Manufactured Product 1	Manufactured Product 1	MP01	Manufactured	5	0	5
	Manufactured Product 2	MP02	Manufactured	5	0	5
	Manufactured Product 3	MP03	Manufactured	10	5	5
	Manufactured Product 4	MP04	Manufactured	10	0	10
	Product Part 1	PP01	Purchased	10	8	2
	Product Part 2	PP02	Purchased	15	10	5
	Product Part 3	PP03	Purchased	5	2	3
	Product Part 4	PP04	Purchased	15	10	5
	Product Part 5	PP05	Purchased	15	10	5
	Product Part 6	PP06	Purchased	10	10	0
	Product Part 7	PP07	Purchased	15	10	5
	Product Part 8	PP08	Purchased	10	5	5
	Product Part 9	PP09	Purchased	10	15	0
	Product Part 14	PP14	Purchased	30	20	10
	Product Part 15	PP15	Purchased	20	15	5

What code is needed for this process?

Custom Visualforce Page
Create the following custom Visualforce page called “BOM Requirements Input” and configure as follows:

The Manufactured Product will be the product that the visualforce page is executed from. The Quantity_Required will be initially displayed with a default value of 1. The user will click on the Run button to execute the report script using the value. A validation will verify that the total Quantity_Required is greater than 0. The Manufactured Product and the Total Quantity_Required will be stored in the custom metadata type “Material Requirement Data” fields Top Product and Total_Quantity_to_Build respectively.

Material Requirements Script
Accounting Seed recommends an asynchronous script is launched when the user clicks the Run button.

If any Material Requirements records exist the script will delete all records. This will allow a very easy re-running of this process and will avoid duplicate records.

Step 1 - Create Product Part Records:
The Material Requirements script creates a Material Requirements record for each Product Part within the Multi-level BOM Product, starting with the Top Product itself.

Top Product:
The script will first set 4 variables.
1. It will set the Level_0_Product_Parent variable to the Top Product from metadata type “Material Requirement Data”.
2. It will set the Current_Part_Level to 0.
3. It will set the Level_0_Quantity_for_Build variable to equal the Total_Quantity_to_Build from metadata type “Material Requirement Data”.
4. It will set the Current_Quantity_to_Build variable to equal the Total_Quantity_to_Build from metadata type “Material Requirement Data”.

The Top Product will be added as a new Material Requirements record with the following fields populated:
- Date = Today’s Date.
- Top Product = Top Product from metadata type “Material Requirement Data”
- Top Product Quantity to Build = Total_Quantity_to_Build from metadata type “Material Requirement Data”
- Part Name = Top Product from metadata type “Material Requirement Data”
- Product Code = Product Code of the Top Product.
- Inventory Type = Inventory Type of the Top Product.
- Quantity_for_1 = 0
- Quantity_for_Build = Total_Quantity_to_Build from metadata type “Material Requirement Data”
- Quantity_in_Stock = 0

The script will then proceed to the Top Product’s first Product Part.

Purchased Product Part:
If the Product Part type equals “Purchased”, the script will first check to see if the Product Part already exists on the object.

If the Product Part already exists on the object, the Quantity_for_Build will be calculated per Step 2 below and added to the current Quantity.

If it does not, the Product Part will be added as a new Material Requirements record. It will populate the following fields:

- Date = Today’s Date.
- Top Product = Top Product from metadata type “Material Requirement Data”
- Top Product Quantity to Build = Total_Quantity_to_Build from metadata type “Material Requirement Data”
- Part Name = Current Product Part’s Name
- Quantity_for_1 = Current Product Part’s Quantity
- Quantity_for_Build = Quantity calculated per Step 2 below.
- Quantity_in_Stock = 0

It will then proceed to the Product Parent’s next Product Part.

Manufactured Product Part:
If the Product Part type equals “Manufactured”, the script will first check to see if the Product Part already exists on the object.

If the Product Part already exists on the object, the Quantity_for_Build will be calculated per Step 2 below and added to the current Quantity.

If it does not, the Product Part will be added as a new Material Requirements record populating the following fields:.

- Date = Today’s Date.
- Top Product = Top Product from metadata type “Material Requirement Data”
- Top Product Quantity to Build = Total_Quantity_to_Build from metadata type “Material Requirement Data”
- Product Part = Current Product Part Name
- Product Code = Current Product Part’s Product Code
- Inventory Type = Current Product Part’s Inventory Type.
- Quantity_for_1 = Current Product Part’s Quantity
- Quantity_for_Build = Quantity calculated per Step 2 below.
- Quantity_in_Stock = 0

If the product part has children then the following additional logic should be applied. Instead of proceeding to the Product Parent’s next Product Part, the script will first save the Current_Product_Parent data.
The script will save the following data from Current_Product_Parent data.
1. The Current_Part_Level will also be increased by 1.
2. The Product Parent will be saved to the Level_x_Product_Parent variable, where x equals the Current_Part_Level. For example, the Current_Part_Level =1. The manufactured Product Part will be saved to the Level_1_Product_Parent variable.
3. The Level_x_Quantity_for_Build variable, where x equals the Current_Part_Level will be set to the Quantity calculated per Step 2 below. For example, the Current_Part_Level =1. The Level_1_Quantity_for_Build variable will be set to the Quantity calculated per Step 2 below.

Once the Script has saved the Current_Product_Parent data, the Script will:
1. Set the Current_Product_Parent to the manufactured Product Part
2. Set the Current_Quantity_to_Build to the Quantity calculated per Step 2 below

The Script will then proceed to the new Product Parent’s first Product Part.

When the Current_Product_Parent’s last Product Part has been processed, the script will set:
1. The Current_Part_Level will then be decreased by 1.
2. The Current_Product_Parent will be set to the Level_x_Product_Parent where x equals the Current_Part_Level.
3. The Current_Quantity_to_Build will be set to the Level_x_Quantity_for_Build variable, where x equals the Current_Part_Level

The script will then return to the Product Parent record of the Current_Product_Parent, and proceed to the next Product Part after the manufactured Product Part that was just completed.

This method of changing Product Parents and recording their Product Parts will continue as many levels deep as required to fully identify all of Product Parts of the multi-level BOM product.

The final results of this process will be a comprehensive listing of all of the Product Parts, both manufactured and purchased, required to build the Top Product.

Step 2 - Calculate Quantity_for_Build:
The Quantity_for_Build is the total quantity of each manufactured product and each purchased Product Part required to build the Top Product's Total Quantity_Required. The Quantity_for_Build as noted above is a field type Number(16,0). Depending on how you manage and allocate your inventory, decimal places can be added.

The Quantity_for_Build is calculated for each product per Step 1.

The basic calculation for the Product Part Quantities is the Current_Quantity_to_Build multiplied by the Quantity that is on the Product Part record (the Quantity required to manufacture one Parent Product). For example, if the Current_Quantity_to_Build = 5 units and the Product Part Quantity = 5, the Product Part's Quantity_for_Build = 25 (5 x 5).

Step 3 - Calculate Quantity_in_Stock:
The Script will calculate the current quantity available for each product on each Material Requirements record. The Script will step through the Material requirements records. For each record, the script will run a SOQL query on the Inventory Quantity Available (IQA) Object, selecting all IQAs whose Product matches the Product Part on the Material Requirements and summarizing the Quantity Available of the selected records. The script will then set the Quantity_in_Stock field of the Material Requirements record to this summarized Quantity Available.

Step 4 - Adjust Quantity_for_Build:
At this point, the script will reduce the Quantity_for_Build of the Product Parts for manufactured products already built and available in Inventory. The process will be similar to step 1 as it steps through each of the Product Parts of that manufactured product.

The script will first check to see if any adjustments are required. Only manufactured Products are checked. If all of the manufactured Products Parts have a Quantity Available = 0, then no adjustments are required and this step is complete.

If there are manufactured Products with a Quantity Available > 0, the Script will step through each Material Requirements record starting with the first.

If the Product Part type is Purchased, the script will proceed to the next record.

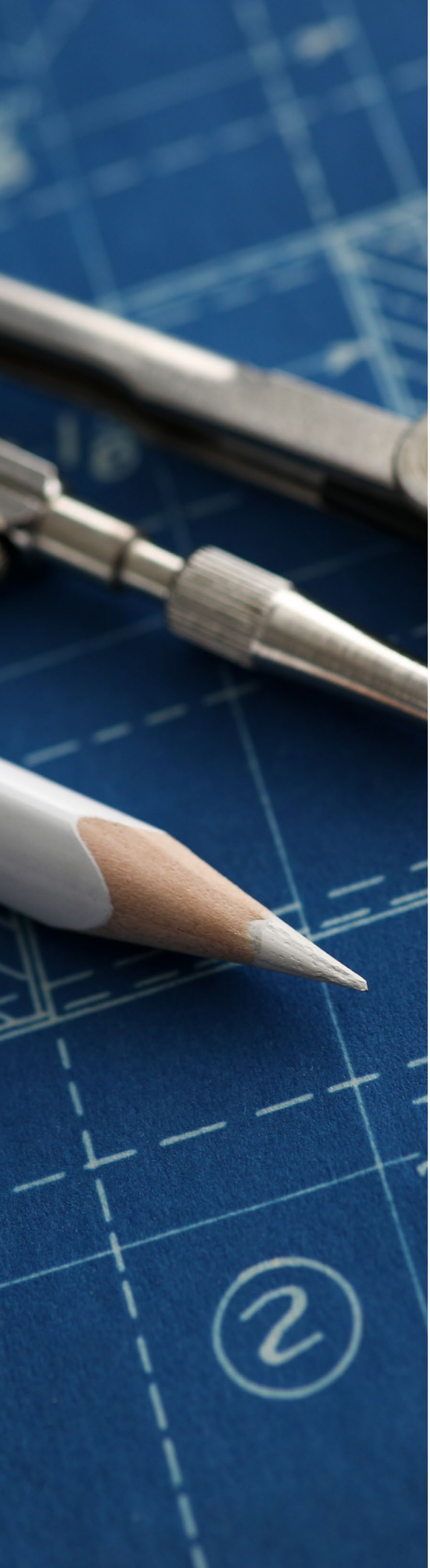
If the Product Part type is manufactured and the Quantity Available = 0, the script will proceed to the next record.

If the Product Part type is manufactured and the Quantity Available > 0, the script will then proceed to that Product's first Product Part and calculate and apply the adjustment amount, per the Apply Adjustment section below. The script will proceed through all of the manufactured Product's Product Parts performing the same calculation. After the last Product Part, the scription will return to the Material Requirements object and process the next record using the criteria above.

Apply Adjustment:
The adjustment amount calculation is Manufactured Product's Quantity Available multiplied by the Product Part's Quantity (the Quantity required to manufacture one Parent Product). For example, if the Quantity Available = 5 units and the Product Part Quantity = 5, the adjustment amount = 25 (5 x 5). The script will then find the matching Product Part record and subtract the adjustment amount from the Quantity_for_Build. If the resulting amount is less than 0, the Quantity_for_Build is set to 0.

Step 5 - Calculate Quantity_Required:
The Quantity_Required is a formula field. The formula is Quantity_for_Build * Quantity_in_Stock.

Step 6 - Execute and Display Report:
The Script will execute and display the Material Requirement Data custom report.



What does a successful outcome look like?

With the successful run of the script, Material Requirements records are created for all the Product Parts of the Top Product and the Top Product. These will be available by opening the Material Requirements tab. An example of what this would look like is below.

The Material Requirements Report is displayed to the user. An example of the report is above.

What process or policy changes should I consider with implementing this solution?

1. Additional Automation can be added to create a Purchase Order (or Orders) directly from the Material Requirements object.
2. Additional filters and fields could be added to segment the Quantity_in_Stock by Warehouse and Location to identify transfer requirements or to exclude specific Warehouses or Locations from the calculations.

About Accounting Seed

Accounting Seed is a modern, robust accounting platform powered by the Salesforce platform. We're committed to breaking down silos and building connections in order to take your business to the next level. Schedule a free demo here or contact us today to begin our conversation.

accountingseed

410.995.8406
accountingseed.com