FILLRATE
100

FILLRATE100 USER'S GUIDE

Ref: gf-2020-001-003

CONTENTS

Introduction 2

Initial configuration 2

Frequent updates 3

Daily operation 4

Step 1 – Buffer Management 4

Step 2 – Upload SKU-Locations status 6

Step 3 – Generate replenishment orders 7

CASE: 8

Production and Purchase Orders 10

Dashboard analysis 11

Configuration of Parameters 12

Special SKU-Location Parameters 12

Size of Red zone in buffers 12

Parameters 13

ANNEX 15

Considerations for a productive plant 15

Production line as a system 15

Production cost 18

Recommendations to manufacturers and wholesalers 18

How to calculate initial buffers 19

Seasonality and Promotions 20

FILLRATE
100

Introduction

FILLRATE100 is web application that works as SaaS (software as a service), for a subscription fee.

The main goal of FILLRATE100 is to help making only one daily decision, answering correctly the next question:

How much should I replenish of each SKU in each Location?

FILLRATE100 answers this question based on the Theory Of Constraints, developed by Dr. Goldratt.

The difference between FILLRATE100 and other methods to make this decision is to consider actual consumption instead of a forecast.

To do that, we need to choose a frequency for each SKU-Location. It must be fixed, and the shortest possible. See the chapter to decide frequency to replenish.

Initial configuration

Each SKU at each Location must be unique. It is mandatory to set the following data for each one:

- SKU: the code in your ERP or system.
- Location: the warehouse where we want to hold inventory for this SKU.
- Buffer: the amount of inventory that we decided to maintain rotating (consuming and replenishing).
- External id: It is a unique identifier for this SKU-Location. Recommended to use exactly this combination; SKU+Location (see samples).
- Replenishment time: the longest between supply time and order lead time. This is mandatory because it tells FILLRATE100 when to suggest changes to buffers. Default is 7 days if you don't supply any number.

Optional fields:

- Description: it is useful to have a description.
- Origin: it is not mandatory, however it is useful to have it since it allows to filter data and export sets of instructions per source.
- Day to replenish: this field indicates the day to replenish a certain SKU at a certain location. FILLRATE100 can work without this information, however, it is much easier for the user to filter per this field and only choose those SKU-Locations that must be replenished that day. It is especially useful to replenish from a productive plant or an external supplier (See the chapter to decide frequency to replenish.)

- Unit cost: it is useful to show the investment in inventory in graphs and tables, since it does not make sense to show a total adding meters with units with liters. If your inventory only has one unit of measurement, it might not be so important for you.

Frequent updates

Usually, once a day is enough for most supply chains. The recommendation is to update status data at least once a day and just before deciding replenishment orders.

The data needed for the update is:

- External id: exactly the same as in the configuration.
- On Hand: the amount of stock that is available for consumption for the SKU at the location.
- Transit: the amount of stock that, although it is not yet available, it is already commited to arrive to the location.

Optionally, you can include SKU, location and description, however External id is all what FILLRATE100 needs to identify the SKU-Location.

Daily operation

Every time you create a new SKU at a location, or you need to modify some data, you need to upload the configuration spreadsheet.

Note that you don't need to include all SKU-Locations or all fields. If you want to create five new SKU-Locations, just include those in the spreadsheet.

If you only want to modify buffer size, or replenishment time, just import External id and the desired columns.

You can also import all the data every time. Only be careful with buffers, because importing will over write old data without warning.

Once you ensured you have all SKU-Locations in FILLRATE100, it is time to follow the procedure:

Step 1 – Buffer Management

Buffers are adjusted according to the pace of consumption:

- When several days within replenishment time, buffer status was red (on hand below 1/3 of the buffer), FILLRATE100 will suggest to increase buffer by 1/3.
- When several days within replenishment time, buffer status was green (on hand over 2/3 of the buffer), FILLRATE100 will suggest to decrease buffer by 1/3.

A couple of notes here:

- When buffer status is black, FILLRATE100 will not suggest increase because there is not enough information to explain why there is a shortage. You can always manually adjust the buffer in case you think the buffer is way too low, either importing or in Current status, editing the respective SKU-Location.
- When buffer status is light blue, that is over 1, FILLRATE100 will not suggest a decrease because there will be no replenishment anyway. If you want to stop replenishment of one specific SKU-Location, you can manually set the buffer to zero, and instruct manually an order to retrieve the remainder on hand from that location to its origin or to another location.

FILLRATE100 runs automatically every six hours a process looking for suggestions.

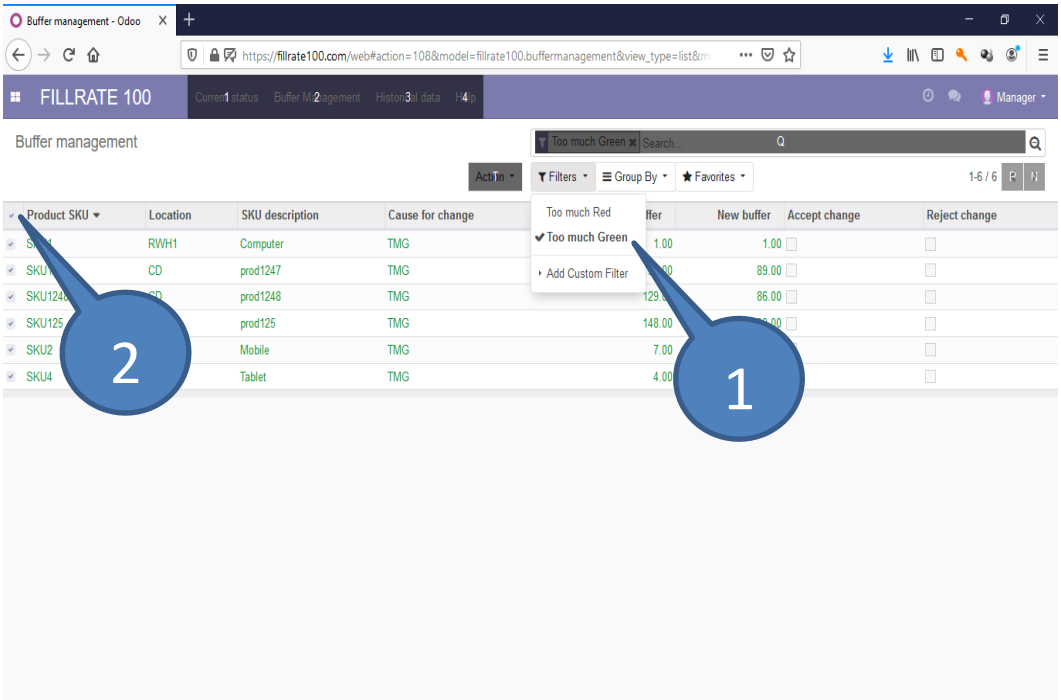
Regarding acceptance or not of suggestions, FILLRATE100 has two possibilities:

- Suggestions are NOT accepted automatically, which is the default. In this case, you have to go to Buffer Management and see whether you want to accept the suggestions that FILLRATE100 found.
- Suggestions are accepted automatically. In this case, there is nothing to do, because you will never find any suggestions pending of acceptance.

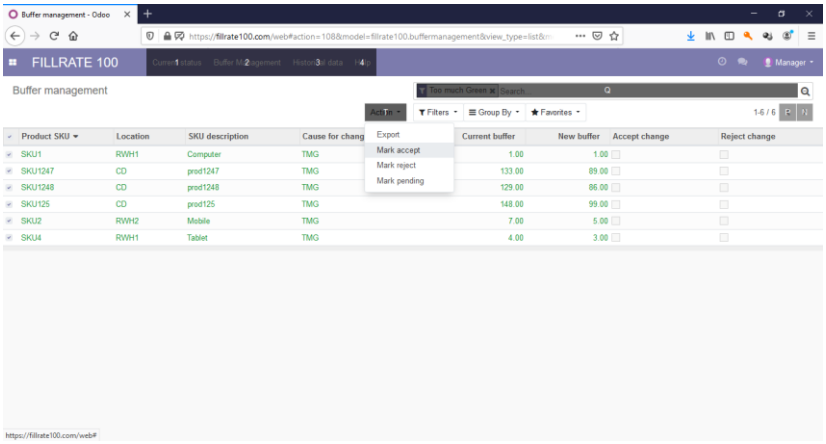
Only if you have set your system in not automatic mode, you have to first do the buffer management task. If it is automatic, skip this step and go to the next directly.

The way to Accept or Reject suggestions:

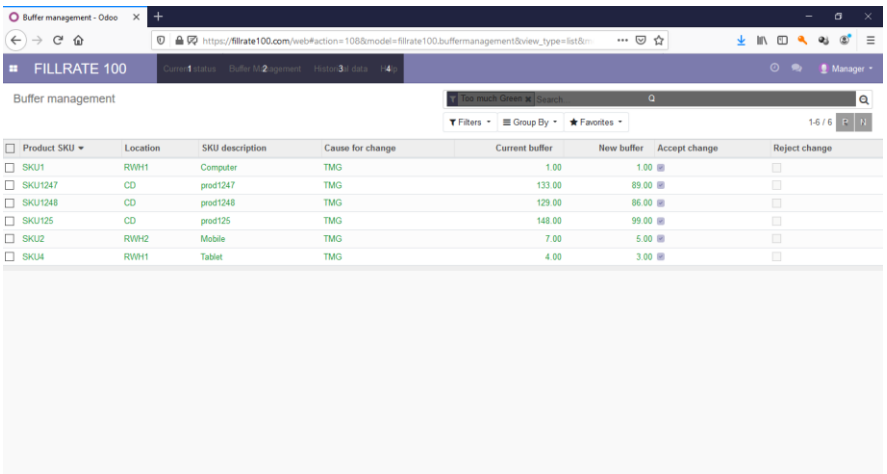
If you want to accept some and reject others, you filter first (you can add more criteria). And second, all the filtered ones are marked.



And then you choose what you want to do from the 'Action' menu.



FILLRATE 100



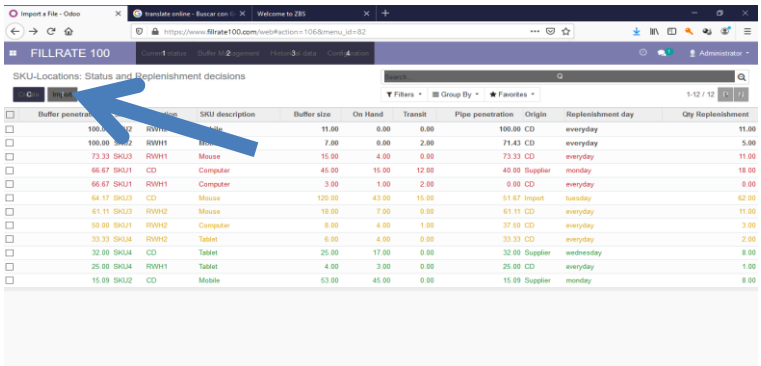
That's it. Every hour, FILLRATE100 executes the marked decisions.

Step 2 – Upload SKU-Locations status

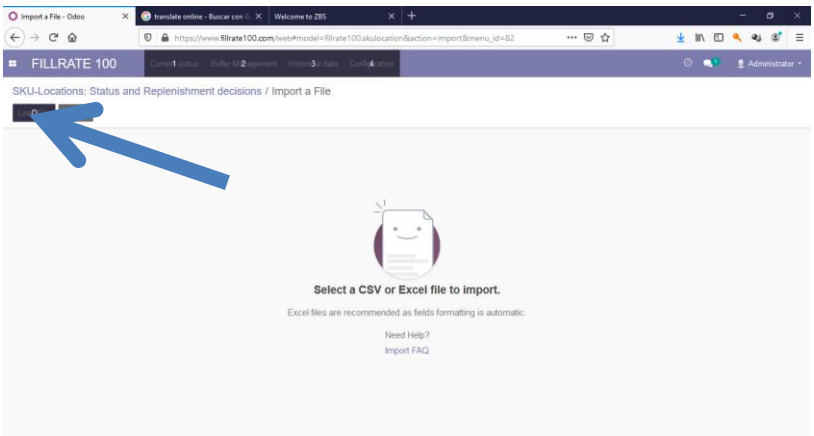
To get the most updated need for replenishment, FILLRATE100 requires the updated on hand and transit for each SKU-Location.

Before using FILLRATE100, you needed to have this information so the assumption is that you can get it easily from your system. Whether it is an ERP or just a worksheet, you can develop an automatic process to extract these two numbers (on hand and transit) for each SKU at each location every time you want.

Prepare the worksheet with SKU-Location, current 'on hand' and current 'transit'. Then, import this worksheet into current status in FillRate100.



FILLRATE
100



Choose you worksheet with the updated data and import.

Step 3 – Generate replenishment orders

Now we have all data up to date and FILLRATE100 already calculated the required quantities for those SKU-Locations that need replenishment.

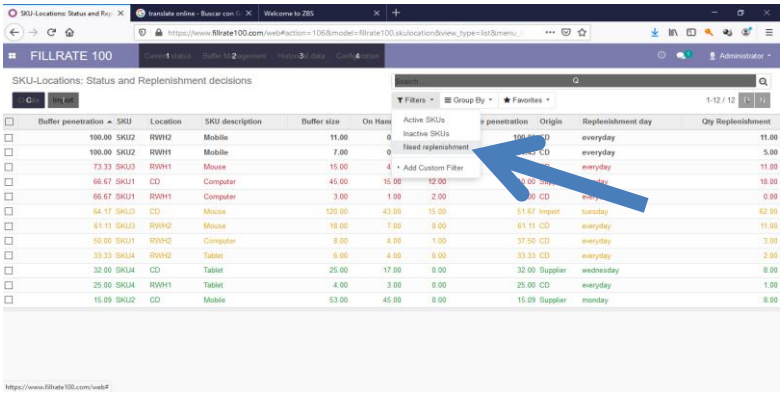
Here you can choose to filter your data with different criteria and then export the appropriate records to different formats.

The exported records can be used as you please:

- Send different worksheets to different people in your organization.
- Put the generated file in a folder so an automatic process feeds it to your ERP.
- Other possibilities you have in your own procedures to execute replenishment order to the various locations.

Note that you can mark or unmark visible records, leaving the hidden records untouched. This way it is easy to export partially only the set of records you need.

The first filter to apply is 'Need replenishment', leaving out of this query all SKU-Locations that don't need any replenishment order.



One of the filters that should be used every time is by ‘Day to replenish’.

CASE:

Let us suppose we have in our stock a laptop that we purchase to a wholesaler. We have a DC (distribution center) where we receive all inventories from external suppliers. And we own three stores where we want to offer this laptop for sale.

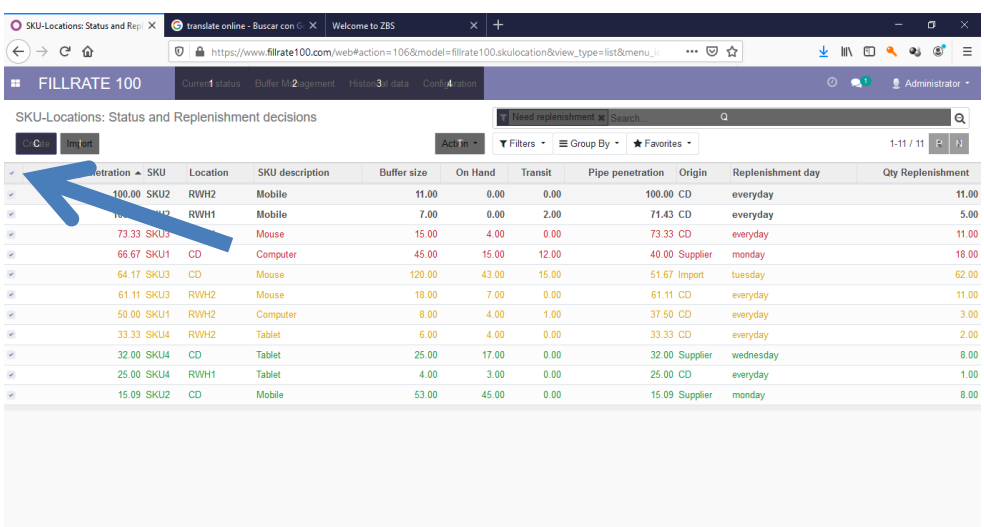
- Locations: DC, Store1, Store2, Store3.
- SKU: LAP-01
- Stock-Locations: LAP-01DC, LAP-01Store1, LAP-01Store2, LAP-01Store3.

Each SKU-Location has a buffer, a replenishment time, a day to replenish.

We can replenish any day we want from DC to any store, because we have full control of that. However, the wholesaler wouldn’t be specially happy if we send him a purchase order every day for a few units, so we choose a day of the week to place purchase orders, for example Tuesday. Day to replenish is then set like this:

STOCK-Location	Day to replenish
LAP-01DC	Tuesday
LAP-01Store1	Every day
LAP-01Store2	Every day
LAP-01Store3	Every day

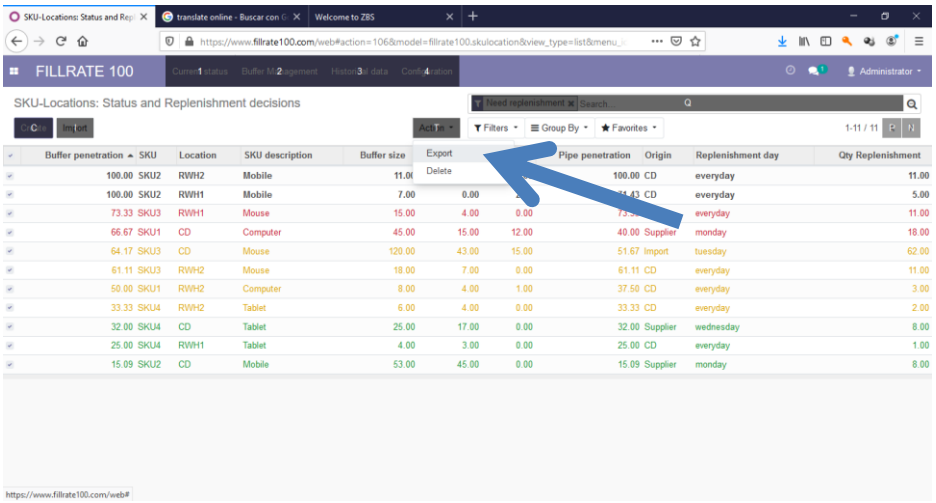
Now we can group by Origin, and separate internal sources from external sources. Then we filter by Day to Replenish, and choose only those that corresponds to the current date. And now we can mark the records that we want to export.

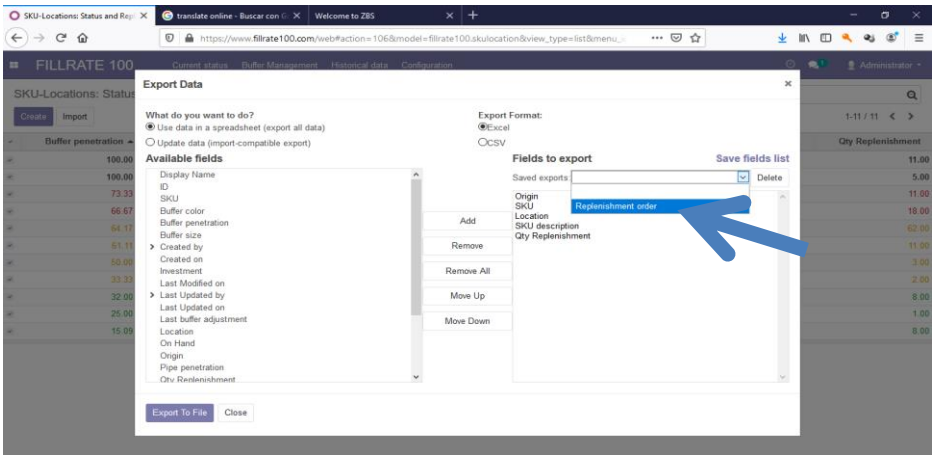


Pipe penetration	SKU	Location	SKU description	Buffer size	On Hand	Transit	Pipe penetration	Origin	Replenishment day	Qty Replenishment
100.00	SKU2	RWH2	Mobile	11.00	0.00	0.00	100.00	CD	everyday	11.00
[illegible]	[illegible]	RWH1	Mobile	7.00	0.00	2.00	71.43	CD	everyday	5.00
73.33	SKU3	[illegible]	Mouse	15.00	4.00	0.00	73.33	CD	everyday	11.00
66.67	SKU1	CD	Computer	45.00	15.00	12.00	40.00	Supplier	monday	18.00
64.17	SKU3	CD	Mouse	120.00	43.00	15.00	51.67	Import	tuesday	62.00
61.11	SKU3	RWH2	Mouse	18.00	7.00	0.00	61.11	CD	everyday	11.00
50.00	SKU1	RWH2	Computer	8.00	4.00	1.00	37.50	CD	everyday	3.00
33.33	SKU4	RWH2	Tablet	6.00	4.00	0.00	33.33	CD	everyday	2.00
32.00	SKU4	CD	Tablet	25.00	17.00	0.00	32.00	Supplier	wednesday	8.00
25.00	SKU4	RWH1	Tablet	4.00	3.00	0.00	25.00	CD	everyday	1.00
15.09	SKU2	CD	Mobile	53.00	45.00	0.00	15.09	Supplier	monday	8.00

FILLRATE 100

Now export the desired set of columns. Note that you can have different templates to export data.





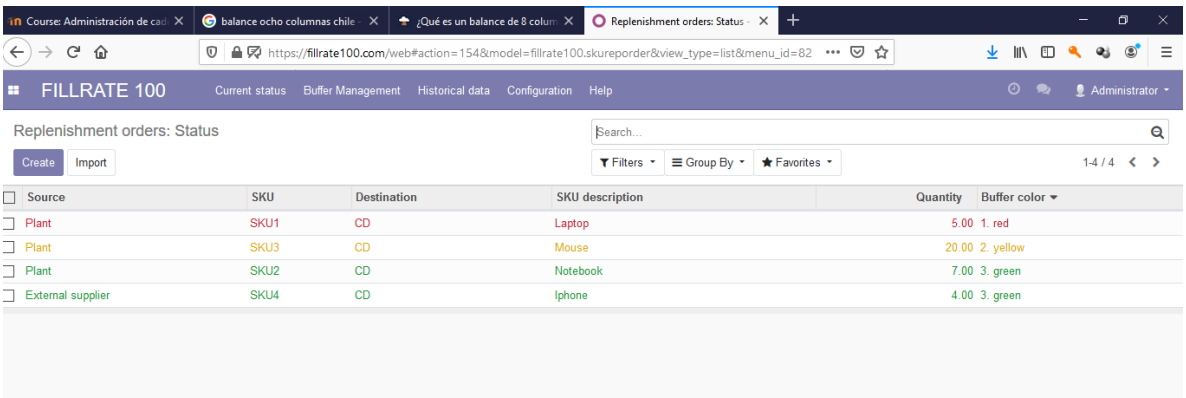
Production and Purchase Orders

Optionally, production orders and purchase orders can be loaded into FILLRATE100.

The purpose of loading these orders is to communicate to everyone the real priority for each one of them.

The user types Manager and Admin can create, import, modify and delete orders.

The user type User can see both status of SKUs and Orders. Anyone logged in, even with a smartphone, can see the colors for each SKU-Location and each production order or purchase order.



The screen is similar to others: data can be filtered, grouped, sorted, exported.

The purpose of this information is to focus both, production and purchasing departments, on the correct orders to process.

Especially in production, this feature of FILLRATE100 is useful for supervisors and operators to make the right decisions having at hand the colors. These colors are based on availability.

The color priority system consists in two principles:

- The order is, from maximum priority first:
  - 0. black
  - 1. red
  - 2. yellow
  - 3. green
- When two or more orders have the same color, make them in the order that maximizes productivity.

FILLRATE
100

Dashboard analysis

FILLRATE100 accumulates everyday three pieces of information, aggregated by Location:

- Availability rate
  - It is the percentage of active SKUs with positive on hand.
- On Hand
  - Number of units in total, adding all the stock of all SKUs at that location.
  - It does not make sense in a case where different SKU have different units of measurement.
- Investment
  - The number that results from multiplying the number of units by unit cost any given day is the investment in that SKU that day.
  - FILLRATE100 calculates the total investment in total stock, both on hand and transit, destined to that location, that day.
  - Should unit cost change, history remains as it was recorded.

FILLRATE100 offers different views of your dashboard, filtered by date or location. See tutorials to see some of the possibilities.

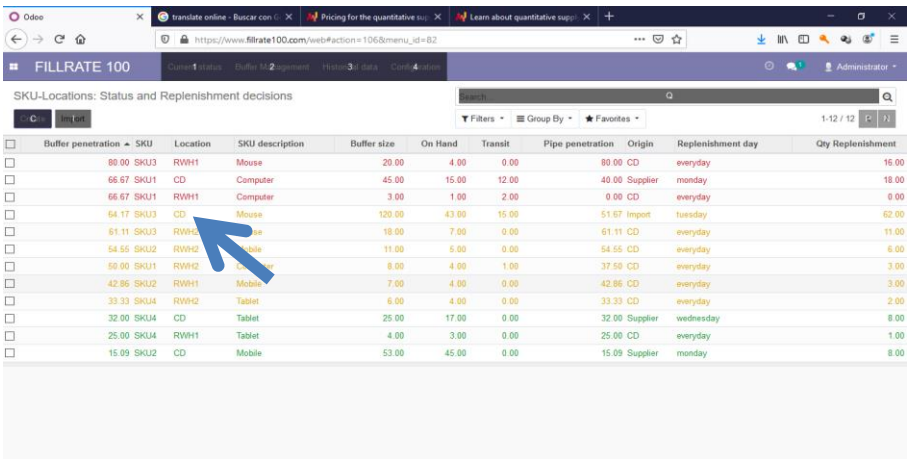
Configuration of Parameters

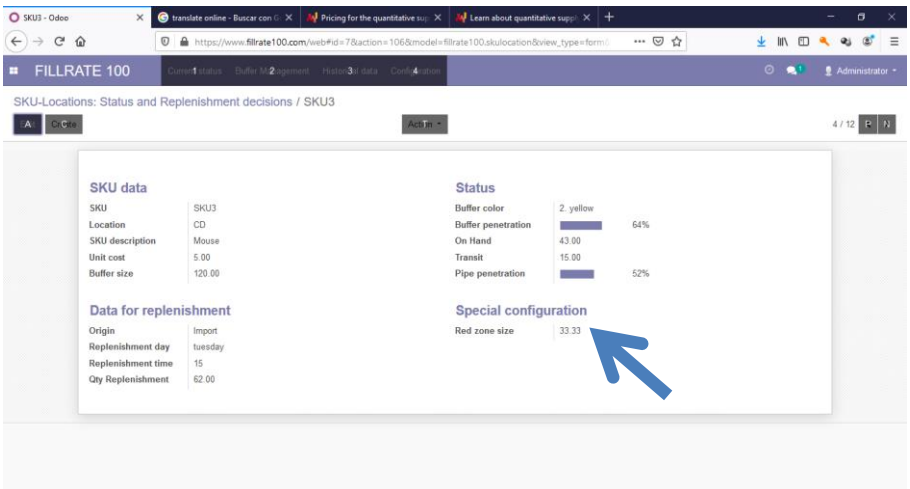
There are three levels of users in FILLRATE100: User, Manager and Admin.

User cannot change anything. Manager can modify special parameters for SKU-Location. Admin can modify everything.

Special SKU-Location Parameters

There are special cases that require special parameters for FILLRATE100 to suggest the correct decisions. These can be modified by both types of users, and they will be available in the main view, but only one by one.





There is only one special parameter for now. Please, do not change the default unless you understand what you are doing.

Size of Red zone in buffers

Default is 1/3 of the buffer. You may want to set this parameter to a higher value or lower value, depending on some conditions.

Higher value

When it is critical to have availability, and the cost of not having dwarves the cost of holding, it may be better to increase inventories.

The way to do this is to set the red zone to be more than 1/3. For example, 40 or 50, will trigger buffer increases when on hand is below those limits. The result is to have a bigger buffer, providing extra safety for availability.

Lower value

When supply time is much longer than order lead time, most of the stock will be in transit.

The color to set buffer status is based only on the on hand, because that is the available stock for consumption, however we shouldn't ignore the investment in all the transit.

As you can see, proportion of on hand/(on hand + transit) can be too much unbalanced when supply time is much longer than order lead time.

On the other hand, on hand accumulates all the slack that we want to have to cope with variability, and that is exactly what we wanted.

But we don't want to allow buffers to be much bigger than necessary, so it might be a good idea to set the red zone in less than 1/3. For example, 10 or 20 will not trigger buffer increases so frequently.

Parameters

Only Admin type users can configure the following parameters for the functionality of FILLRATE100.

- Threshold of Reds
  - It is percentage expressed as a whole number. Default is 60.
  - It is the percentage of occurrences of color red for a SKU-Location within the last replenishment time.
- Threshold of Greens
  - It is percentage expressed as a whole number. Default is 85.
  - It is the percentage of occurrences of color green for a SKU-Location within the last replenishment time.
- Automatic Buffer Changes
  - It is just marked for true, or unmarked for false. Default is false.
  - If it is marked, all buffer changes suggested by FILLRATE100 will be automatically executed.
- Days to keep color history
  - Color history is recorded every 12 hours. This will gather a great deal of history in the database and we need cleaning to avoid excessive accumulation of data.

    - Default is 360 days.
    - Cleaning is performed once per day.
- Days to keep adjustments history
    - Adjustments history is recorded every time that a change was accepted. This will gather a great deal of history in the database and we need cleaning to avoid excessive accumulation of data.
    - Default is 360 days.
    - Cleaning is performed once per day.
- Days to keep KPIs history
    - Dashboard history is recorded every 24 hours. This will gather a great deal of history in the database and we need cleaning to avoid excessive accumulation of data.
    - Default is 360 days.
    - Cleaning is performed once per day.

If designated administrator do not create a record to change this values, FILLRATE100 operates with default.

ANNEX

Considerations for a productive plant

If you are a manufacturer of consumer goods, you probably have a plant warehouse. We can call this warehouse the distribution center.

Usual considerations to decide daily production program are:

- Availability of SKU in the distribution center.
- Minimizing production cost.

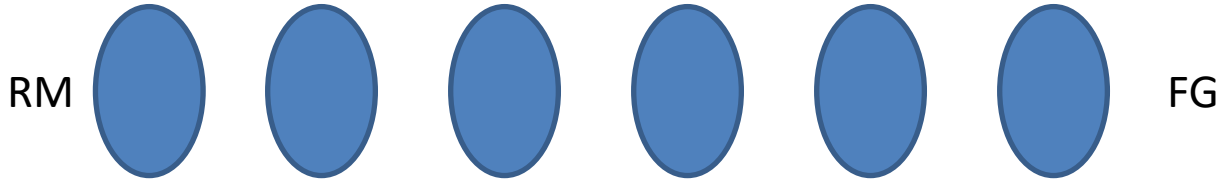
There is no question about the first consideration. After all, we get revenue from selling products that we do have in stock, therefore availability is a primary consideration to have a good fill rate in all our sale orders.

Regarding production cost, let us review some extended and prevalent misconceptions.

But first let us understand what is a production line.

Production line as a system

A production line is usually a chain of dependent processes, as depicted here:



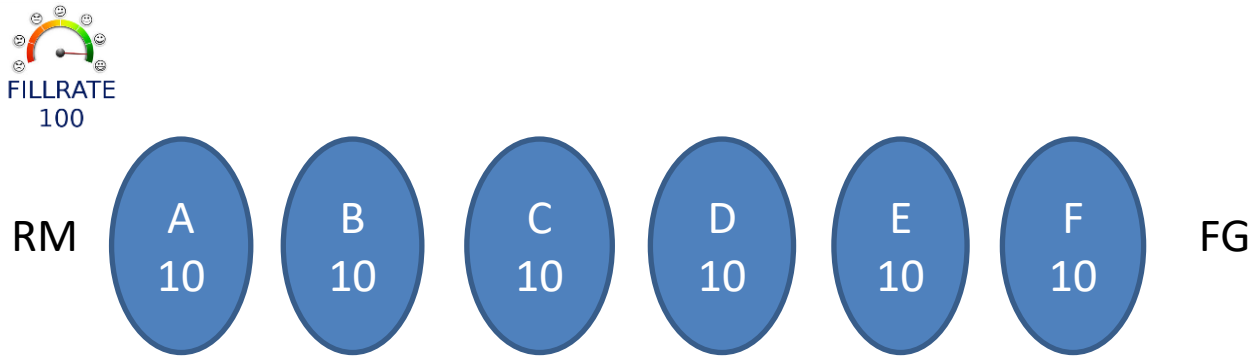
Raw material is fed into the first process and after several steps it is converted into a finished product.

Balanced lines destroy productivity of the system

First we need to understand that we cannot have a balanced line to produce the average production of each process.

There some good games to prove in practice this assertion. My favorite is dice game as described in the novel The Goal, chapter 14.

Here I am going to offer a mathematical reasoning to understand what happens when we have variability and dependency. Let us put some letters and numbers to our process:



The numbers represent the capacity of units per hour that each resource has. But we know that capacity has variability, so 10 units/hour is an average.

Roughly speaking, we can say that half of the time a resource runs at less than 10 units/hours and half of the time runs at more than 10 units/hour.

We don't know when is each half of the time. All the resources from A to F have cycles of more than 10 and less than 10.

Let us start with A, B and C.

If A works at 12 units/hour and B works at 8 units/hour, B will deliver to C only 8 units per hour.

If consider that F is delivering units of finished products, what F does is the whole system productivity. As we have seen, F will work as fast as the slowest in the chain.

To have 10 units/hour in the whole chain, we need that all six resources work at 10 units/hour or more simultaneously. We know that at any given hour, the average will be 1o units/hour or more with probability of ½. What is the probability that all six work at 10 units/hour at the same time?

Probability(six resources @10 or more altogether) = $(1/2)^6 = 1/64$

Approximately 5 or 6 days in a year we can have the average production of the whole line similar to the average of each resource when capacity is balanced throughout the system.

What determines total productivity?

We know now that we need systems with unbalanced capacities.

This is, by the way, the natural state of all systems and most likely our production lines.

We need to know what determines then the productivity of a system that sells consumer goods.

First, we know we don't need to balance internal capacities, so in our example, A can be at 20, B at 15, C at 30, D at 10, E at 25, F at 20, and we will have a steady flow of 10 on average.

But our system is productive when we sell products. This is the first realization in contrast with other views. We could look at how much the plant produces, but it is not what the plant produces, but what the company sells that counts.

Let us look at the whole system:

Demand is all that depletes the warehouse, ending in sales to end consumers.

Productivity will be determined by the constraint of the whole system in terms of what blocks more sales to be made.

What is the constraint here? A better question is, where do we want to be the constraint?

Plant as the constraint

When the plant is the constraint, there is not enough capacity to produce all what the market demands. We can have all the capacity in our warehouse, but the plant will produce slower that demand pulls the goods.

Therefore, we will have many shortages in the warehouse, and eventually demand will accommodate to this capacity: consumers will buy somewhere else.

The warehouse as the constraint

This means that the plant is capable of producing all what market demands but there is not enough space to store the inventories before they are shipped to customers, so we ask the plant to stop sending products. We can have too much inventory for some items and shortages for others.

This situation can be the same when the space or cash constraint is in other nodes of the supply chain. We still don't sell as much as possible. And, again, consumers shop anywhere else.

Demand as the constraint

This means that the company is selling all that market demands.

It also means that plant and warehouse have more capacity than required by demand.

In this situation, producing more than needed before the next replenishment only leads to higher inventory, but not to higher sales. On the contrary, producing more than demand builds excess inventory that will end up as blockage for new sales. Liquidating excess inventory produces less profit and cannibalizes sales of better margin products. And it is possible that goods are perishable or they have a short shelf life. In that case, writing off obsolete stock is a waste of raw materials and energy invested in making them.

Conclusions

To maximize revenue, we need more internal capacity than demand, in our plant, in our warehouses and transportation, and all resources that take the product to the end consumer.

However we don't need all resources utilized all the time to produce what the market demands, therefore the maximum productivity is achieved when there is idle capacity in all our resources.

Production cost

The previous conclusion leads to discuss how to calculate production cost.

We've been taught that an idle resource is always a waste, which is the opposite of what we already know is true.

What are the consequences of believing that "an idle resource is always a waste"?:

- Setups cost money: if we think that a minute lost in production is a minute lost in sales, then we will try to minimize setups as much as possible. In many companies, this is achieved by the MOQ – Minimum Order Quantity. MOQ is the size of the minimum batch to run the process for a specific SKU. But setups rarely cost extra money. Only in cases where material is wasted in the setup process, this is true. In general, setups don't add to real cost. It is true that too many setups can turn the plant into a constraint, and this we must avoid.
- Production cost is calculated as total expense divided by total produced units. We know now that excess inventories cost more. What counts here is total expense divided by total sold units. Therefore, we should stop calculating those unit costs and start considering only raw material cost (this is the recommendation in FILLRATE100 for unit cost per SKU).
- Transportation cost: similar to setup, if we send all the products that we need in a half full truck load, the freight cost is not higher. The tendency of saving these costs also leads to minimum batches which are too big.
- Purchasing cost: another fallacy. We don't spend more money in our procurement if we place more orders. Another thing is that suppliers will accept too frequent orders.

There many other practices in supply chains, from plants to retailers, that are guided by this belief of "an idle resource is always a waste". All those practices are blocking supply chains to really improve.

Recommendations to manufacturers and wholesalers

Supplying from a plant or an external supplier has similarities:

- In both cases, too many orders may cause more chaos than benefits.
- There are some setup times attach to each order to consider, so we better have a reasonable number of orders per SKU.

The recommendation is to set replenishment time to at least weekly.

Manufacturers:

- Separate SKUs in different groups according to setups in the most critical resources.
- Assign groups to days in the week. All the SKUs in a group will have the same Day to Replenish. Thus, when triggering replenishment orders to the plant, work orders will arrive in the best way to minimize setups.
- Do not set minimum batches. There is no need for that. The weekly frequency will take care of saving setups.
- In production, look at the color of the SKUs that are in orders to decide priorities. This is the order: black should skip queues. Then: red, yellow and green.

Buying from External Suppliers:

- Separate SKUs in different groups according to suppliers.
- Assign groups to days in the week. All the SKUs in a group will have the same Day to Replenish. Thus, when triggering replenishment orders, purchase orders will be sent only once in a week to the same supplier.
- Do not set minimum batches. There is no need for that. The weekly frequency will take care of not sending multiple orders in a week to the same supplier. As for minimum quantities, either you negotiate discounts based on monthly purchases or accept them. In the latter case you will have some items in light blue for a while.
- You can look at the color of the SKUs that are in purchase orders to focus your follow up on the black and red ones.

How to calculate initial buffers

When companies start using FILLRATE100, one prevailing question is, how do we calculate initial buffers?

Buffer size is a target amount of stock that we want to keep in our pipeline to satisfy all the reasonable consumption within a replenishment time:

- A buffer must be defined for each SKU at each location where we want to have that SKU.
- Pipeline here is the available stock on hand plus all transit minus consumption.
- We will replenish only to complete the buffer in the pipeline with the predetermined frequency, to cover for replenishment time.
- Reasonable means the highly probable event that average consumption within the replenishment time will not abruptly change. These drastic changes are most likely to have a special cause for them. Most special causes are visible and you can manually adjust buffers. Two of those special causes are treated in another chapter of this section: seasonality and promotions. Note that a drastic change is more than double the demand in a period of two replenishment times. Within these boundaries, buffer management reacts perfectly and keep the correct buffers.

The data needed to calculate the buffers at a certain location are:

- List of SKUs
- History of consumption for the last 3 to 6 months.
- You have decided the frequency and have calculated the resulting replenishment time.

Let us see an example with 20 days of consumption for an SKU with 3 days as replenishment time:

History (days)	Consumption	Moving Sum of last 3 days
1	1	
2	6	
3	4	11
4	8	18
5	6	18
6	6	20
7	0	12
8	1	7
9	0	1
10	4	5
11	6	10
12	1	11
13	8	15
14	9	18
15	5	22
16	9	23
17	8	22
18	10	27
19	7	25
20	7	24

In the 3rd column we see a maximum of 27, saying that in twenty days, we had a maximum of consumption of 27 units in consecutive days. We set 27 to our buffer.

Seasonality and Promotions

Known events of a peak of demand must be handled with more than buffer management.

The main idea is that buffers drive the system, so we change the size of the buffers and we will trigger all the actions to cope with the peak, that can't be handled only with available capacity and regular buffers.

Let us say that we have a promotion for a family of products that are 10% of the total sales. And we expect to sell 3x as the regular sale. So we need for that family a temporal increase of the buffers at the appropriate locations of three times, to have built the inventory by the date of the event.

In this case, the total increment will demand from the system an additional capacity of 20%:

	Regular	Promotion
Sales of promotion	10%	30%
Sales non promoted	90%	90%
Total sales	100%	120%

In this case you could plan an increase in one step, and probably the system will absorb it.

When the promotion is off, a replenishment time before, reduce buffers to previous state, to avoid excess inventory.

Seasonality is a promotion that affects most of the SKUs. In this case, an expected double in sales will require an incremental increase of the buffers in more steps, so your system can absorb it.

Remember to reestablish the buffers to the previous state when seasonality is off. It is recommended to reduce buffers at least one replenishment time before the event.