

Special Issues

Linking & Linking Issues

Linking similar Inventory Items is a very useful feature of the Cimarron Restaurant Solutions System. There are many reasons we implemented this system. The most important have to do with Menu Ingredients Management and “Taking Inventory”.

Scenario #1

Lets assume that you purchase Carrots from three different vendors, depending upon which vendor has the best quality or availability from week to week. Further assume that you have a recipe that calls for 1 pound of carrots as an ingredient. When you created that recipe you had to pick a very specific carrot from inventory, “Carrots#1”. Your recipe was originally created using 1 pound of “Carrots#1” from Vendor A at a cost of \$1/pound. So your recipe now has a cost of \$1. Next week, however, the price of carrots doubles and only Vendor B has any in stock, so you buy “Carrots#2” from Vendor B. Our system is built to update the cost of every recipe based upon it’s ingredients costing in real time. However, this recipe did not update since the ingredient listed in the recipe, “Carrots#1”, did not change pricing. The system has no way of updating the cost of a recipe that does not contain the exact item just purchased, in this case “Carrots#2” from Vendor B. Your recipe should actually cost twice as much as last week, but the carrots we bought this week from Vendor B are not listed as an ingredient in the recipe, so the Ingredient Cost did not change. And next week when we get a really good deal from Vendor C for “Carrot#3” at \$1.50/pound, the recipe will remain unchanged for the same reason; the ingredient line refers to “Carrots#1” from Vendor A, not “Carrots#2” nor “Carrots#3” from vendors B and C.

To address this problem, we developed a “LINKING” system whereby we link “Carrots#1”, “Carrots#2”, & “Carrots#3” so that the system treats all three items as one “virtual” item. Now, when you buy ANY of these linked Carrots, the system will see that your Recipe refers to a linked item and will substitute that newly purchased item as the current line item in the recipe. So when you buy the \$2 carrots from Vendor B, the system will go to the recipe, change Carrots#1 (at \$1/pound) to Carrots#2 (at \$2/pound) and re-

calculate the cost of that recipe. The same thing will occur again when you purchase Carrots#3 from Vendor C next week; Carrots#3 will become the item referred to in the recipe at the current cost of Carrots#3 (\$1.50/pound). Now the true cost of every menu is updated based upon real time invoice charges for Ingredient Items from day to day – as long as similar items are linked properly.



We extended this “recipe” logic further to apply to count forms as well. Even though you are buying carrots from 3 different vendors, you don’t want to see 3 different products on your count forms. You just want to see one, preferably the last one you purchased. So linking also “swaps” linked products on and off of the count forms. This will not affect your inventory value reports, since all the carrots will show up with all purchases and beginning value lumped together. But it can affect your inventory valuation at count time if you do not use FIFO. If you use price last paid and you happened to be counting at a recent high price purchase, you will skew your inventory value. If you do not have similar products linked and you only count one of them you’ll do the same thing. On the other hand, if you have these products linked AND you have FIFO on, it just doesn’t matter which one you count since the system is going to use the invoices for all the links to make the valuation. This gives you get one additional advantage because you can consciously decide to link products but turn swapping OFF on some. That means these products never swap onto inventory forms, yet their purchase values are counted properly if you have FIFO on. For instance, you could decide to always count vodka by the case and never see the single Vodka bottle purchases (Show on forms OFF, Swapping OFF). You don’t want that single expensive bottle showing on count forms, even if you purchased it last. With the two linked and FIFO on, all Vodka will be “counted” properly at inventory time.

The important thing to understand about this system is the fact that the system is just a dumb machine, and you are responsible for making sure it can swap one ingredient out for another and still calculate the cost of the recipe properly. Think about it: if you buy item #1 in pounds and item #2 in kilograms, the system will not calculate the cost properly. Or if you setup your pack units so that the system thinks that the case price can substitute for the pound price you will have a problem. Suddenly the one pound in the recipe at \$1/pound is being calculated at the cost of a \$20 case containing 20 pounds of carrots. The system can handle cases and pounds and each just fine; BUT it depends upon the pack units being properly setup to do so. This, in the end, is a simple matter of logic that every kitchen manager knows intuitively. But it may take a little extra thought to make sure the database is set up properly to reflect that logic.

Therefore,

RULE #1 for LINKING is: portion units must exactly match. Your recipe always counts cost by portion unit. So all linked products must resolve to the cost of the item with matching portion units -- normally ounce, or gram, or each, etc.

RULE #2 for LINKING is: “counting” units must match. Your pack units must be set up properly so that the “counting” unit from product to product is the same. The purchase unit and pricing unit may differ. But each linked product must contain the same “counting” unit so that the final portion unit calculation is being done in a way that is consistent between all linked products. If you count carrots by the pound, make sure each linked product can be counted by the pound. (Or each, or kilo, or pouch, or bag, or container, or.... Etc.)

The words don't necessarily have to be the same. But the logic of the math involved must. For instance one product can have the word “Bottle” while the other has the word “each” or “750ml” or “btl”. But the pack unit must REFER to the same unit of measure. In this example we all understand that “each” really means “one bottle”. If “750ml” refers to a bottle but “each” refers to a case, you will have problems.

For instance, here is a screen shot of 3 carrot items that are linked and are setup properly. These 3 items will resolve properly so that linking will work for you:

Carrots#1 @ .55/pound (Priced by the pound)

	Purchase Unit	Pricing Unit			Portion Unit
	1 ↓	2 ↓	3	4	5 ↓
Inventory Units	Pound	Pound			Ounce
Modifiers	1	1	1	1	16
Price	0.55	0.00	0.00	0.0344	

Carrots#2 @ .31/pound (Priced by the BAG) Note that the Purchase unit is per BAG (25 pound bag). This will still work fine since we set up the item so that we can count it by the pound.

	Purchase Unit	Pricing Unit			Portion Unit
	1 ↓	2 ↓	3	4	5 ↓
Inventory Units	Bag	Bag	Pound		Ounce
Modifiers	1	1	25	1	16
Price	7.80	0.31	0.31	0.0195	

Carrots#1 @ .65/pound (Priced by the pound)

	Purchase Unit	Pricing Unit			Portion Unit
	1 ↓	2 ↓	3	4	5 ↓
Inventory Units	Pound	Pound			Ounce
Modifiers	1	1	1	1	16
Price	0.65	0.00	0.00	0.0406	

This is a pretty typical issue with linked items. Quite often vendors will ship and price similar items in different pack units. But in the end, you must be able to set them up so that the portion units and count units match, or you cannot successfully link the items. Perhaps it would be useful to point out a pack unit that would NOT work as an illustration of this point.

Carrots#4 (by the case only. No other modifier available.) This item WILL NOT LINK PROPERLY to the three above, nor would it be useable in a recipe. Note that if linking were to substitute this into a recipe, you would actually be specifying 16 CASES (each) of carrots. Not 16 ounces. And if

you were to count 23 pounds of carrots as the end-of-month-inventory, you would be telling the system to calculate 23 CASES. Your value for carrots would be outrageous – and wrong.

	Purchase Unit	Pricing Unit			Portion Unit
	1 ↓	2 ↓	3	4	5 ↓
Inventory Units	Case	Case			Each
Modifiers	1	1	1	1	1
Price	13.00	0.00	0.00	13.0000	

How does this happen in real life? (And it does happen because we see it everyday in real life database reviews.) The reason is that the admin inputting an invoice does not grasp the importance of, nor the logic of pack units and how they relate to counting inventory or as calculation modifiers within recipes. When you enter an invoice, the all-important thing is that the system invoice total matches the vendor’s invoice total. So if the vendor says you bought a case of carrots at \$13.00 per case, the above item will work fine. The admin can balance the invoice, produce will show \$13.00 of purchases during the month and IF you indeed counted 23 cases of carrots at month end, then all is just fine. BUT, if this is linked to the other carrots above, and you counted 23 pounds, your month end reports are probably now really out of whack because the system thinks you just counted 23 CASES of Carrots#4! So the above item is not “technically” wrong for some circumstances. But it is certainly not optimum and will probably get you in trouble. It should have been setup like our Carrots#2 above.

This could be avoided simply by making sure that Admin understands the issues involved with “counting” and with “linking” for purposes of counting and swapping recipe line item ingredients for current “costing” purposes. Make sure your Admin understands the 4 important logical choices when setting up a new vendor product:

1. A vendor requires that you PURCHASE by some quantity (Case).
2. He PRICES it by some quantity (for instance Case or Pound).
3. You put it in inventory and COUNT it by some logical breakdown (by case, packet within the case, pound, “each”, etc.)
4. AND finally, IF this product might be used in some menu recipe item, what is its logical “PORTION UNIT”, the smallest unit it can be broken down to?

ADVICE: Take your Admin back to help count at month end if you are finding mismatches or there seems to be inaccuracies in the inventory value on month end reports. It almost always can be traced to improperly setup inventory pack units that probably work fine for invoices, but do not work the way the kitchen staff actually use, refer to, and count that product.

Scenario #2

Another use for linking products has nothing to do with keeping recipe costing current. It has to do strictly with inventory value properly calculating when using the FIFO inventory method at month end. Assume that you buy Vodka by the case of 12 bottles for \$82, or 6.83 per bottle. You intend to count “bottles” at month end (not Cases), and your vodka item is setup with the FIFO inventory method.

	Purchase Unit		Pricing Unit		Portion Unit
	1 ↓	2 ↓	3	4	5 ↓
Inventory Units	Case	Case	Liter		Ounce
Modifiers	1	1	12	1	33.92
Price	82.00	6.83	0.00	0.2015	

However, there are times when you have to buy just one bottle. So you setup a separate item for this. The Single Bottle Item without the discount costs you 9.50.

	Purchase Unit		Pricing Unit		Portion Unit
	1 ↓	2 ↓	3	4	5 ↓
Inventory Units	Bottle	Bottle	Liter		Ounce
Modifiers	1	1	1	1	33.92
Price	9.50	9.50	0.00	0.2801	

You bought 1 case last month and you bought one bottle at the end of this month. If your month end inventory count was 3 bottles, it means that you should have one bottle valued at 9.50 and 2 bottles valued at 6.83. Here is where you can get into trouble without linking. If you enter your count into the single bottle item, AND you have not linked it to the case pricing, you just valued all 3 bottles at 9.50. If you entered the count into the case purchase item without linking on, then you just valued all 3 bottles at 6.83. However, if you link these two items, the system will take all invoices for

these two items and starting with the last invoice entered (the single non-discounted bottle) calculate the value of one bottle at 9.50, then go to the next invoices entered (the case purchase last month) and value the next 2 bottles at 6.83.

ALLOW LINK SWAPPING?

Every item you link can be marked to Allow Link Swapping or not.

Every time you purchase an item, the system checks to see if it is a linked item. If it is, it checks further to see if it is “swappable”. IF it is swappable, then it will mark the last item purchased to “Show On Inventory Forms” and mark all the other linked items to NOT show on forms. This means that if you have 3 carrot items linked, and swapping is turned on for all three, you will only see one on your count forms, the last one purchased. This way your count forms are not filled with several items describing the same thing all at different prices. Since they are linked, and swapping is turned on, your count will always be the last one purchased at the last price paid. AND if you have FIFO turned on, the system will use real invoices for all linked items to calculate the true inventory value.

The screenshot shows a software dialog box titled "Inventory Value Method" with the following sections and settings:

- Inventory Value Method:** Radio buttons for FIFO, LAST PRICE, and AVERAGE.
- UNIT TO COUNT:** A table with columns for Bottle, Bottle, Liter, and Ounce. Below it is an EACH and a small input field with the value 0.
- GL Account:** A text field containing "Liquor".
- Location(s):** A list box with minus and plus buttons.
- Show on Inventory Forms?:** Radio buttons for Yes and No.
- Allow Link Swapping:** Radio buttons for ON and OFF.
- Track on Variance Reports?:** Radio buttons for Yes and No.
- Set Pars:** A row of seven input fields for days of the week: Sun, Mon, Tue, Wed, Thu, Fri, Sat, each containing the value 0.

Callouts from the image:

- Each product can be set for specific inventory value method here: FIFO, LAST PRICE, and AVERAGE
- Show on forms: YES or NO
- Allow Link Swapping: YES or NO

There are times, though, when you don't want to swap items in and out of recipe ingredients or on and off of inventory forms. The Vodka single bottle above is a good example. You decide you only want to display the case item

on forms, not the single bottle, as “case” is the more normal way you purchase it. If you mark these items as NON-swappable, then no matter which Vodka you purchased last time, the one you manually mark as “Show on Inventory Forms” will be the one that you see on count forms.

This Vodka is probably not in a menu recipe so marking it non-swappable will have no bearing on recipe costing. However, this strategy should be well thought out if you want to link items that appear in recipes to items that could not swap in and out of ingredients. The menu auto costing mechanism could potentially be compromised. As long as the user understands the system well, this can be used to great benefit by linking several items that would be used for FIFO inventory value purposes and at the same time control which of those items were allowed to be swapped in and out of menu ingredients as well as on and off inventory forms.

Inventory Valuation

The system provides a mechanism for entering “inventory on hand” counts and calculating the value of the inventory on hand. This is done in the “Enter Inventory” screen. Upon entering this screen, the system displays only those items that have “Show On Inventory Forms” turned ON. If you see items you do not wish to see, go into the item record and change the “Show On Inventory Forms” button to off. And vice versa. Be aware also that the system may not show some items on the count forms, but it will display ALL items that have ANY activity on the various Inventory Reports. This fact has many ramifications for the inventory value the system calculates. It is entirely possible, for instance, to buy a product that has been turned off for showing on count sheets, and count a product that has never been bought nor used. This could be valid and desirable in some circumstances. But it could also lead to gross inaccuracies as well. You are responsible for knowing your inventory and reviewing the many system reports and keeping your data current and valid. We feel we have a system that makes this very easy to do and yet provides the opportunity to have very detailed and accurate inventory activity reports. Normally common logic is all that is needed. One does not have to be a computer whiz to get it right. Hopefully the following discussions of Inventory Valuation will be helpful in understanding our system logic.

Counting

Your first line of defense is in the actual counting of your inventory. Normally this is done using an inventory form printed for this purpose. It is important to print these forms on the same day the count is to occur so that the form contains the latest system info. Using a stash of copied forms you printed last month will get you into trouble as things often change within the system and data on those forms may now be invalid. After the forms are filled out during a physical count, someone will type those amounts into the “Enter Inventory” screen. That is the perfect time to correct inventory problems. When you enter an amount in the “count” column and tab out, the system will use the valuation method chosen for that inventory item in conjunction with its current “Pack Unit Setup” to calculate the cost of the on hand amount you just entered.

The important thing to always be on the look out for when you enter inventory is the Unit of Measure for the item being counted. Everything

depends upon this “Inv Unit” being correctly setup, and correctly interpreted at count time.

Ending Inventory Entry					
Prod No	Description	Brand	Vendor	Pack Unit	Inv Unit
5916	Ice Cream	Barbers	Barber	3 Gal. Tub	Gallon
50849	Margarine Prints	Kraft	Kraft	30/1#	Pound
1234	Monterey Jack Cheese	Kraft	Kenneth	4/10# Pkg.	Case
30163	Sour Cream	Breakstone	Kraft	4/5# Cont'r	Cont'r
B*9	Table Butter Blend	House Item	Batch	Pound	Pound
75047220	Whipping Cream, 36%	Olde Style	Kraft	12/Quarts	Quart

Make sure you check the Inv Unit (Unit of Measure) to count!

In other words, if the UM is “Contr” and someone just counted this item in “Pounds”, inventory value will not be correct. A common mistake is when two people count the same item in two locations and one counts “pounds” while the other counts something else (“each” etc). It is important to review the “Inv Unit” on the count form and make sure everyone understands what they are counting. Sometimes it is simply impossible to count what is printed on the form because the Admin just didn’t “get it” and put in anything at all just to make an invoice come out right. This happens especially with an Admin who simply had little training when he/she took over the job from a buddy or has little experience in the kitchen. We think anyone who is in charge of inputting new products at invoicing time should help count inventory from time to time. This will probably help clean up your inventory file better than any other one thing you can do.

Advice: Teach those taking count to make notes on the form that Admin can understand in order to properly correct the Inventory Record Setup screen. The best time to clean up your inventory file and deal with ill-conceived Pack Unit Setups is at the Inventory Count Entry screen.

As you type in the count, review the “Inv Unit”. If it does not seem right or definitely does not match what was counted, change the UM immediately in the inventory setup screen (double click the line to get there) or re-count using the proper unit of measure. In any case, do not type in a number that is obviously a wrong count for the Inv Unit showing on this screen. You will simply get “garbage in/garbage out” results. If you do change it you **MUST** be aware of the consequences if that item was entered into an invoice with the existing “wrong” pack unit in play. This most often means simply don’t change the existing “Pricing Unit” so that it is just not the same as what may be in existing invoices. If historical invoices are not an issue, there will be no consequences. Invoices will **ONLY** be an issue if the product is setup for FIFO **and** your count exceeds the last purchased amount. Only then will the system go back through past invoices trying to calculate the “real” value of

the items you just counted. The safest correction to an inventory item is a simple change to the “count” unit. You can change the entire pack unit or add a missing count unit and the product will still calculate properly through past invoices as long as you don’t completely change the way the system calculates purchase/pricing units. In other words, it is probably not a good idea to change the Pricing Unit from “case” to “pound” at the end of the month. It would be safe to ADD a count unit of “pound” as an extra column for a case pack unit as long as it properly equates to the existing case. If you do need to change the product setup entirely you are probably best advised to get through the closing and change it at the very beginning of the new month before you enter any new invoices.

As an example, lets say that you buy an item by the case, and you break it out to inventory it by the container (6 per case). The case price on the last invoice was 29.28. Admin entered an invoice for 2 cases and had to setup the item since it was new to the system. This admin knew nothing about the item or how it would be counted, so he/she simply entered the easiest thing possible, a simple “case” pack unit. The invoice will look and calculate just fine.

	Purchase Unit 1 ↓	Pricing Unit 2 ↓	3	4	Portion Unit 5 ↓
Inventory Units	Case	Case			Each
Modifiers	1	1	1	1	1
Price	29.28	0.00	0.00	29.2800	

At Month End you count 10 containers of this item, but it is obviously not setup correctly since it is stored and counted as individual small containers and there is no way you have 10 cases. You decide that you need to change the product to match the count taken so you make the following change.

	Purchase Unit 1 ↓	Pricing Unit 2 ↓	3	4	Portion Unit 5 ↓
Inventory Units	Cont'r	Cont'r			Each
Modifiers	1	1	1	1	1
Price	29.28	0.00	0.00	29.2800	

BEWARE! You just changed the Pricing Unit and the math logic underneath it will no longer match your invoice reality. When you count this item, you will definitely be placing a very high value on it! Those 10 containers you

counted actually cost you 4.88 each, NOT 29.28 each. A difference of \$234 in value!

CORRECT SETUP AND LOGIC: Follow the 4-step Cimarron Logic:

1. **PURCHASE UNIT:** The vendor makes you buy it by the “CASE”.
2. **PRICING UNIT:** The vendor prices it by the “CASE” on invoices. Admin sees the “Pricing Unit” column on invoices so they will see and enter the proper number of cases being purchased and your invoice will match the vendor’s invoice.
3. **COUNT UNIT:** You COUNT by the container. There are 6 containers in a case. So if you count 10 containers the system calculates 1.6666 cases. The system will use past invoices to see how much you paid for those 1.6666 cases and your total inventory value will calculate properly.
4. **PORTION UNIT:** Always try to carry the logic into the last column.
- 5.

This setup will calculate your container count properly and still work properly with the “wrong” setup in existing invoices using FIFO (see first setup screen above).

	Purchase Unit	Pricing Unit		Portion Unit
	1 ↓	2 ↓	3	4
Inventory Units	Case	Case	Cont'r	
Modifiers	1	1	6	1
				5 ↓
				Dunce
				16
Price	29.28	4.88	0.00	0.3050