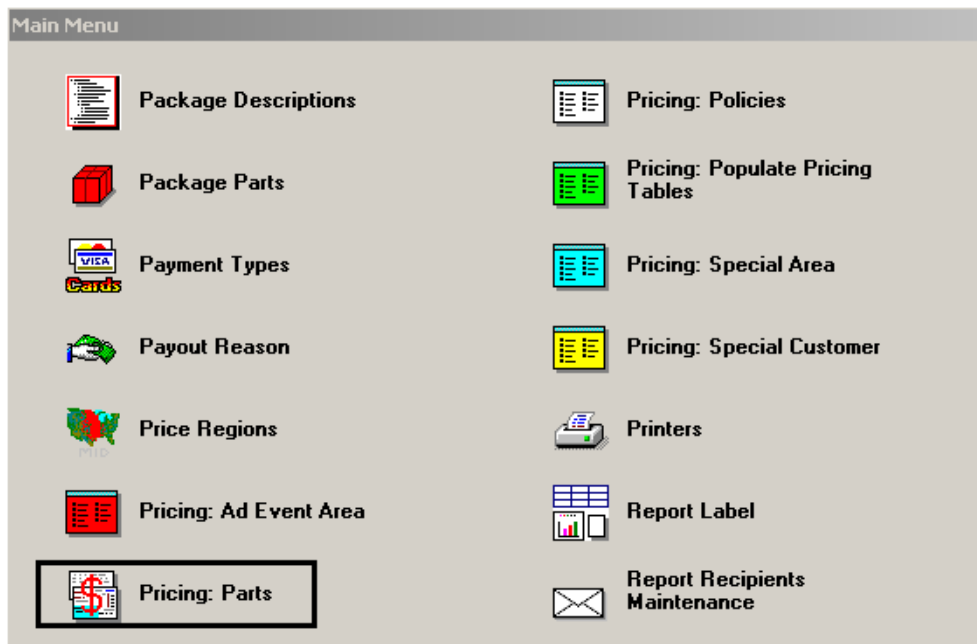




Pricing Policies

Pricing polices allow you to create **GLOBAL PRICING CHANGES** for your parts. For one or two parts, it is usually standard practice to go into the master table, change the price and then update the shop from latest master. Pricing Policies take this one step higher by giving you the ability to change pricing for not only a few parts but also your entire master table if needed.

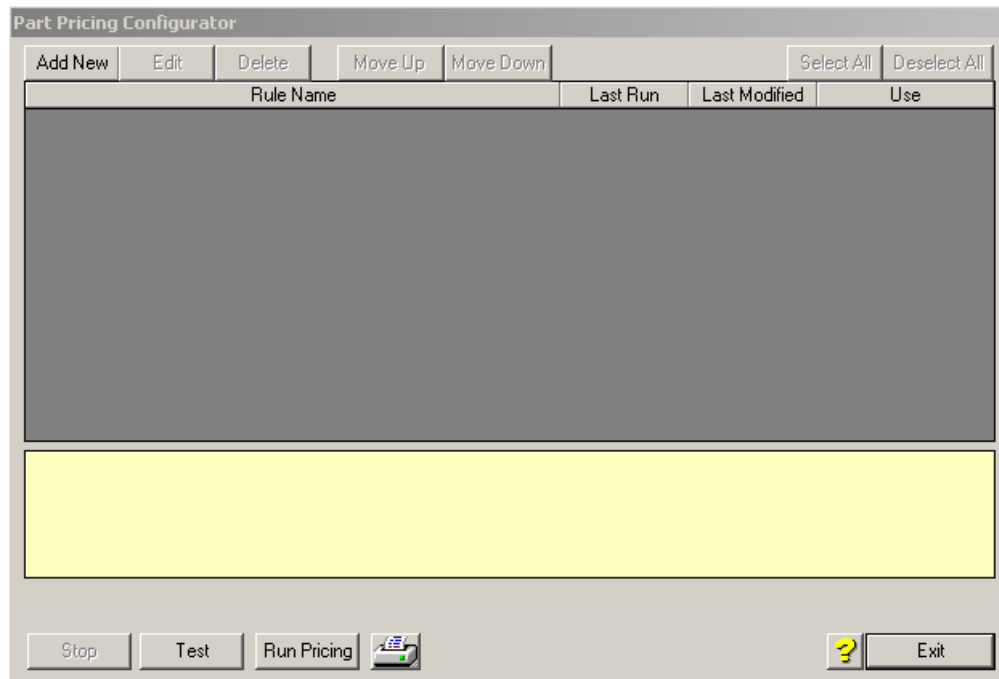
Pricing polices is located in **VAST MAINTENANCE** under the module called **PRICING: PARTS**. This module will allow you to add, edit, delete and run user made pricing policies.



DEFINITIONS

PRICING PARTS CONFIGURATOR

The **PRICING: PARTS MODULE** opens up a window separate from VAST Maintenance. This window is called the **PART PRICING CONFIGURATOR**, seen below:



This is the opening window; this window is where you will find the basics of starting a pricing policy. The buttons on this screen are defined as follows:

1. **ADD NEW**- Add a new Pricing Policy.
2. **EDIT** - Edit an existing Pricing Policy.
3. **DELETE** - Delete an existing Pricing Policy.
4. **MOVE UP** - Allows you to move a Pricing Policy up the list.
5. **MOVE DOWN** - Allows you to move a Pricing Policy down the list.
6. **ENABLE ALL** - Allows you to flag all Pricing Policies to be available.
7. **DISABLE ALL** - Allows you to unflag all Pricing Policies to be available.
8. **SELECT ALL** - Sets all Pricing Polices to Use and prepares them to be run.
9. **DESELECT ALL** - Clears all Pricing Polices Use flag to prevent them from running.
10. **STOP** - Button allows you to stop the Pricing Policy during its run-time.
11. **TEST** - Button will bring up a test of a run policy (This is important).
12. **RUN PRICING** - Executes all Pricing Policies that have the Use checkbox flagged.
13. **PRINTER (PICTURE)** - Prints all created Pricing Policies.
14. **?** - Takes the user to the SMS Help Menu.
15. **EXIT** - Allows the user to exit the Part Pricing Configurator.

Now that we have a brief overview of the buttons on the **PART PRICING CONFIGURATOR**, lets give a brief explanation of the fields in the middle of the screen.

Rule Name	Last Run	Last Modified	Use

The fields on the screen are defined as follows:

1. **RULE NAME** - The name of the Pricing Policy.
2. **LAST RUN** - The last time the Pricing Policy was run.
3. **LAST MODIFIED** - The last time the Pricing Policy was edited.
4. **USE** - This flags the Pricing Policy to be allowed to process.
5. **DISABLED** -This disables the Pricing Policy and doesn't allow the **USE** to be flagged.

PART PRICING RULE DETAIL SCREEN

The **PART PRICING RULE DETAIL** screen is shown below. The next section discusses what is shown on this screen:

The format is broken into three parts:

- a. The **TOP** part is the **NAME** of the pricing rule.
- b. The **MIDDLE** part has fields that contain **ANY VALUE**. This is the filtering mechanism of the Pricing Program. The filtering is how you tell the system what information needs to be changed.
- c. The **LAST** part has fields that contain **NO CHANGE**. This is the updating field. The updating fields are what you will be updating your pricing information to.

FILTERING FIELDS

There are many different ways you can select what parts you want to change. Listed below is a brief description of what each field is and how you can filter them. If you hit the drop down arrow to the right, you will see each selection listed below.

Description	Brake Pads end in .99
Account Code	Any Value ▾
Class	Any Value ▾
Manufacturer	Any Value ▾
Part Number	Any Value ▾
List Price	Any Value ▾
Price Add-On	Any Value ▾
Labor Price	Any Value ▾
Labor Add-on	Any Value ▾
Core Value	Any Value ▾
Part Cost	Any Value ▾
Published Cost	Any Value ▾
Selling Price	Any Value ▾
Selling Labor	Any Value ▾
Part Description	Any Value ▾

- a. **ACCOUNT CODE:** This field allows the user to filter by particular account codes.
 - Any Value:** The default selection that does not filter a value.
 - Is equal to:** Allows filtering for a particular account code.
 - Not equal to:** Excludes a particular account code.
 - Is Less Than:** Any account code that has less than the input value will be included in the filter.
 - Is Greater Than:** Any account code that has more than the input value will be included in the filter.
 - Is Less Than and Equal To:** Any account code that is equal or less than the input value will be included in the filter.
 - Is Greater Than or Equal To:** Any account code that is equal or more than the input value will be included in the filter.
 - Is in Range:** Assign two Account Codes and all Account Codes between them will be included in the filter.

- b. **CLASS:** This field allows the user to filter by Class.
 - Any Value:** The default selection that does not filter a value.
 - Is Equal To:** Allows filtering for a particular class
 - Is Not Equal To:** Excludes a particular class from the filter.

- c. **MANUFACTURER:** This field allows the user to filter by Manufacturer
 - Any Value:** The default selection that does not filter a value.
 - Is Equal To:** Allows filtering for a particular manufacturer
 - Is Not Equal To:** Excludes a particular class from the filter.

- d. **PART NUMBER:** This field allows the user to filter by Part Numbers.
- Any Value:** The default selection that does not filter a value.
 - Is equal to:** Allows filtering for a particular part number.
 - Not equal to:** Excludes a particular part number.
 - Is Less Than:** Any part number that has less than the input value will be included in the filter.
 - Is Greater Than:** Any part number that has more than the input value will be included in the filter.
 - Is Less Than and Equal To:** Any part number that is equal or less than the input value will be included in the filter.
 - Is Greater Than or Equal To:** Any part number that is equal or more than the input value will be included in the filter.
 - Is in Range:** Assign two part numbers and all part numbers between them will be included in the filter.
 - Begins With:** Type in the beginning in the part in the value field and all parts beginning with that value will be included in the filter. (Example: BP will include BP1614M, BP0136, and BP0000)
 - Ends With:** Type in the end in the part in the value field and all parts ending with that value will be included in that filter. (Example: 002 will include 1002, 3002, ABC002)
- e. **LIST PRICE:** This field allows the user to filter by the List Price.
- Any Value:** The default selection that does not filter a value.
 - Is equal to:** Allows filtering for a particular List Price.
 - Not equal to:** Excludes a particular List Price.
 - Is Less Than:** Any List Price that has less than the input value will be included in the filter.
 - Is Greater Than:** Any List Price that has more than the input value will be included in the filter.
 - Is Less Than and Equal To:** Any List Price that is equal or less than the input value will be included in the filter.
 - Is Greater Than or Equal To:** Any List Price that is equal or more than the input value will be included in the filter.
 - Is in Range:** Assign two List Prices and all List Prices between them will be included in the filter.
- f. **PRICE ADD ON:** This field allows the user to filter by Part Add Ons.
- Any Value:** The default selection that does not filter a value.
 - Is equal to:** Allows filtering for a particular Add On.
 - Not equal to:** Excludes a particular Add On.
 - Is Less Than:** Any Add On that has less than the input value will be included in the filter.
 - Is Greater Than:** Any Add On that has more than the input value will be included in the filter.
 - Is Less Than and Equal To:** Any Add On that is equal or less than the input value will be included in the filter.
 - Is Greater Than or Equal To:** Any Add On that is equal or more than the input value will be included in the filter.
 - Is in Range:** Assign two Add Ons and all Add On between them will be included in the filter.
- g. **LABOR PRICE:** This field allows the user to filter by the Labor Price.
- Any Value:** The default selection that does not filter a value.
 - Is equal to:** Allows filtering for a particular Labor Price.
 - Not equal to:** Excludes a particular Labor Price.
 - Is Less Than:** Any Labor Price that has less than the input value will be included in the filter.
 - Is Greater Than:** Any Labor Price that has more than the input value will be included in the filter.
 - Is Less Than and Equal To:** Any Labor Price that is equal or less than the input value will be included in the filter.
 - Is Greater Than or Equal To:** Any Labor Price that is equal or more than the input value will be included in the filter.
 - Is in Range:** Assign two Labor Price and all Labor Prices between them will be included in the filter.

- h. **LABOR ADD ON:** This field allows the user to filter by the Labor Add On.
- Any Value:** The default selection that does not filter a value.
 - Is equal to:** Allows filtering for a particular Labor Add On.
 - Not equal to:** Excludes a particular Labor Add On.
 - Is Less Than:** Any Labor Add On that has less than the input value will be included in the filter.
 - Is Greater Than:** Any Labor Add On that has more than the input value will be included in the filter.
 - Is Less Than and Equal To:** Any Labor Add On that is equal or less than the input value will be included in the filter.
 - Is Greater Than or Equal To:** Any Labor Add On that is equal or more than the input value will be included in the filter.
 - Is in Range:** Assign two Labor Add Ons and all Labor Add On between them will be included in the filter.
- i. **CORE VALUE:** This field allows the user to filter by the value of the Core.
- Any Value:** The default selection that does not filter a value.
 - Is equal to:** Allows filtering for a particular Core Value.
 - Not equal to:** Excludes a particular Core Value.
 - Is Less Than:** Any Core Value that has less than the input value will be included in the filter.
 - Is Greater Than:** Any Core Value that has more than the input value will be included in the filter.
 - Is Less Than and Equal To:** Any Core Value that is equal or less than the input value will be included in the filter.
 - Is Greater Than or Equal To:** Any Core Value that is equal or more than the input value will be included in the filter.
 - Is in Range:** Assign two Core Value and all Core Value between them will be included in the filter.
- j. **PART COST:** This field allows the user to filter by the Cost of the part.
- Any Value:** The default selection that does not filter a value.
 - Is equal to:** Allows filtering for a particular Part Cost.
 - Not equal to:** Excludes a particular Part Cost.
 - Is Less Than:** Any Part Cost that has less than the input value will be included in the filter.
 - Is Greater Than:** Any Part Cost that has more than the input value will be included in the filter.
 - Is Less Than and Equal To:** Any Part Cost that is equal or less than the input value will be included in the filter.
 - Is Greater Than or Equal To:** Any Part Cost that is equal or more than the input value will be included in the filter.
 - Is in Range:** Assign two Part Cost and all Part Cost between them will be included in the filter.
- k. **PUBLISHED COST:** This field allows the user to filter by the Published Cost.
- Any Value:** The default selection that does not filter a value.
 - Is equal to:** Allows filtering for a particular Published Cost.
 - Not equal to:** Excludes a particular Published Cost.
 - Is Less Than:** Any Published Cost that has less than the input value will be included in the filter.
 - Is Greater Than:** Any Published Cost that has more than the input value will be included in the filter.
 - Is Less Than and Equal To:** Any Published Cost that is equal or less than the input value will be included in the filter.
 - Is Greater Than or Equal To:** Any Published Cost that is equal or more than the input value will be included in the filter.
 - Is in Range:** Assign two Published Cost and all Published Cost between them will be included in the filter.

- l. **SELLING PRICE:** This field allows the user to filter by the Selling Price.
****Note:** Selling price is the Part Price + Add On. This is the Price the user sees when they enter a part on an estimate.
 - Any Value:** The default selection that does not filter a value.
 - Is equal to:** Allows filtering for a particular Selling Price.
 - Not equal to:** Excludes a particular Selling Price.
 - Is Less Than:** Any Selling Price that has less than the input value will be included in the filter.
 - Is Greater Than:** Any Selling Price that has more than the input value will be included in the filter.
 - Is Less Than and Equal To:** Any Selling Price that is equal or less than the input value will be included in the filter.
 - Is Greater Than or Equal To:** Any Selling Price that is equal or more than the input value will be included in the filter.
 - Is in Range:** Assign two Selling Price and all Selling Prices between them will be included in the filter.

- m. **SELLING LABOR:** This field allows the user to filter by the Selling Labor.
****Note:** Selling price is the Labor Price + Labor Add On. This is the Price the user sees when they enters Labor on an estimate.
 - Any Value:** The default selection that does not filter a value.
 - Is equal to:** Allows filtering for a particular Selling Labor.
 - Not equal to:** Excludes a particular Selling Labor.
 - Is Less Than:** Any Selling Labor that has less than the input value will be included in the filter.
 - Is Greater Than:** Any Selling Labor that has more than the input value will be included in the filter.
 - Is Less Than and Equal To:** Any Selling Labor that is equal or less than the input value will be included in the filter.
 - Is Greater Than or Equal To:** Any Selling Labor that is equal or more than the input value will be included in the filter.
 - Is in Range:** Assign two Selling Labor and all Selling Labor between them will be included in the filter.

- n. **PART DESCRIPTION:** This field allows you to filter by the Description of a part.
 - Any Value:** The default selection that does not filter a value.
 - Contains:** Allows filtering when user enters part of a description.

The filtering mechanisms allows for the user to enter as many filters as needed to reach a desired result. The more filters put into a policy, the more detailed the data will be. The less filters put onto a policy, the more general the policy will be.

UPDATING FIELDS

The following are the fields that you can use to **UPDATE** your pricing:

- a. **ADJUSTED SELLING PRICE:** This field will allow the user to change the amount of the Selling Price.
 - No Change:** No changes will be made to the Selling Price.
 - Replace With \$ Amount:** Changes the Dollar amount of the price after the pricing policy is run.
 - Increase By \$ Amount:** Adds the inputted dollar amount to the price after the pricing policy is run.
 - Decrease By \$ Amount:** Subtracts the inputted dollar to the price after the pricing policy is run.
 - Increase By %:** Increases the price of the part by the allocated percentage.
 - Decrease By %:** Decreases the price of the part by the allocated percentage.
 - Replace Cents With:** Replaces the cents of the price with the value input by the user.

Set Add On To: Changes the amount of the Add On by the inputted value.

Multiply Cost By: Takes the cost of a part and multiplies the cost by the value inputted by the user to create a new price.

Multiply Cost By At Least: Takes the cost of a part and will multiply it by the value inputted by the user and if the cost multiplier is already greater on some of the parts that happen to fall on the filter, it will not change the price.

Set to GM % Of: Allows the user to change the price based off the gross margins. This will compare the current price with the cost and change the price to compensate for the value of the percentage inputted by the user.

Set to GM % Of At Least: Same as above except if the GM is already greater than the value inputted by the user, then it will not change that percentage.

- b. **ADJUSTED SELLING LABOR:** This field will allow the user to change the amount of the Selling Labor.

No Change: No changes will be made to the Selling Labor.

Replace With \$ Amount: Changes the Dollar amount of the labor after the pricing policy is run.

Increase By \$ Amount: Adds the inputted dollar amount to the labor after the pricing policy is run.

Decrease By \$ Amount: Subtracts the inputted dollar to the labor after the pricing policy is run.

Increase By %: Increases the price of the labor by the allocated percentage.

Decrease By %: Decreases the price of the labor by the allocated percentage.

Replace Cents With: Replaces the cents of the labor with the value input by the user.

Set Add On To: Changes the amount of the Add On by the inputted value.

- c. **ADJUSTED PART COST:** This field will allow the user to change the amount of the Part Cost.

No Change: No changes will be made to the Part Cost.

Replace With \$ Amount: Changes the Dollar amount of the cost after the pricing policy is run.

Increase By \$ Amount: Adds the inputted dollar amount to the cost after the pricing policy is run.

Decrease By \$ Amount: Subtracts the inputted dollar to the cost after the pricing policy is run.

Increase By %: Increases the price of the cost by the allocated percentage.

Decrease By %: Decreases the price of the cost by the allocated percentage.

Replace Cents With: Replaces the cents of the cost with the value input by the user.

Increase % of the List Price: Increases the cost by the allocated percentage inputted by the user.

Decrease % of the List Price: Decreases the cost by the allocated percentage inputted by the user.

Increase % of the Published Cost: Increases the cost by percentage based off the Published Cost.

Decrease % of the Published Cost: Decreases the cost by percentage based off the Published Cost.

- d. **WORK HOURS:** This field will allow the user to change the amount of work hours that is assigned to a part.

No Change: No changes will be made to the Work Hours.

Replace Hours With: Changes the amount of hours existing currently with the value the user input.

Increase Decimal Hours By: Adds to the existing work hours, the value that the user input.

Decrease Decimal Hours By: Subtracts from the existing work hours, the value that the user input.

Increase %: Increases the work hours by the percentage that the user allocates.

Decrease %: Decreases the work hours by the percentage that the user allocates.

- e. **MIN PAYABLE HOURS:** This field changes the Minimum Payable hours that are based off of parts.
- No Change:** No changes will be made to the Min Payable Hours.
 - Replace Hours With:** Changes the amount of hours existing currently with the value the user input.
 - Increase Decimal Hours By:** Adds to the existing hours, the value that the user input.
 - Decrease Decimal Hours By:** Subtracts from the existing hours, the value that the user input.
 - Increase %:** Increases the work hours by the percentage that the user allocates.
 - Decrease %:** Decreases the work hours by the percentage that the user allocates.
- f. **MAX PAYABLE HOURS:** This field changes the Maximum Payable hours that are based off of parts.
- No Change:** No changes will be made to the Max Payable Hours.
 - Replace Hours With:** Changes the amount of hours existing currently with the value the user input.
 - Increase Decimal Hours By:** Adds to the existing hours, the value that the user input.
 - Decrease Decimal Hours By:** Subtracts from the existing hours, the value that the user input.
 - Increase %:** Increases the work hours by the percentage that the user allocates.
 - Decrease %:** Decreases the work hours by the percentage that the user allocates.
- g. **CORE VALUE:** Allows the user to change the value of the core.
- No Change:** No changes will be made to the Core Value.
 - Replace With \$ Amount:** Changes the Dollar amount of the core value after the pricing policy is run.
 - Increase By \$ Amount:** Adds the inputted dollar amount to the core value after the pricing policy is run.
 - Decrease By \$ Amount:** Subtracts the inputted dollar to the core value after the pricing policy is run.
 - Increase By %:** Increases the price of the core value by the allocated percentage.
 - Decrease By %:** Decreases the price of the core value by the allocated percentage.
 - Replace Cents With:** Replaces the cents of the core value with the value input by the user.

With these definitions, we now have a better understanding of what changes and what gets filtered. Although you can update more than one field at a time, it is best that you break your policies up and update one field and then create another policy and update the other field. One basic goal to keep in mind is to keep things as simple as you can. If you can understand what you are trying to do, the better chance you have to make it work.

CREATING YOUR PRICING POLICY

Now that we have taken a look at all the definitions and functions of a pricing policy, it is now time to actually try one out. The best way to create a pricing policy is to write down the policy in English and then try to apply it to a policy.

For Example:

“For all my domestic mufflers, I want to multiply their prices at four times my cost.”

Taking our example lets put this into a policy form. Lets take the first half of our English sentence: “For all my domestic mufflers.” In our example we have successfully isolated the filtering mechanism of our sentence. Now all that is needed is to put it in play. I know in my data I store all my domestic mufflers in account code 101. These are the steps I take to form my new policy and filter:

1. Open the **PRICING: PARTS MODULE**.
2. Click on the **ADD NEW** button.
3. Change my description at the top to **SET PRICES TO DOMESTIC MUFFLERS AT 4 TIMES THE COST**.

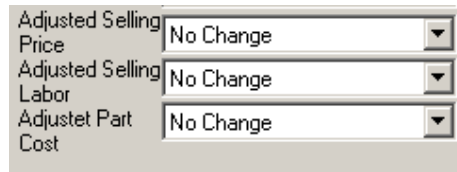
The screenshot shows the 'Part Pricing Rule Detail' window. The 'Description' field contains the text 'SET PRICES TO DOMESTIC MUFFLERS AT 4 TIMES COST'. Below this, there are several dropdown menus for 'Account Code', 'Class', 'Manufacturer', 'Part Number', 'List Price', 'Price Add-On', 'Labor Price', 'Labor Add-on', 'Core Value', 'Part Cost', 'Published Cost', 'Selling Price', 'Selling Labor', and 'Part Description', all of which are currently set to 'Any Value'. At the bottom of the form, there are dropdown menus for 'Adjusted Selling Price', 'Adjusted Selling Labor', and 'Adjusted Part Cost', all set to 'No Change'. To the right of these, there are dropdown menus for 'Work Hours', 'Min Payable Hr', 'Max Payable Hr', and 'Core Value', also set to 'No Change'. A 'Notice' dialog box is overlaid on the form, with the text 'Cumulative Rules determine update values based upon the previously applied rules.' and a checkbox labeled 'This Is A Cumulative Rule' which is currently unchecked. At the bottom right of the form, there are buttons for '?', 'Cancel', and 'Save'.

4. Click on the **ACCOUNT CODE DROP BOX** (Default says Any Value)
5. Select **IS EQUAL TO**.
6. I put in my value. **Example:** 101. ****Note:** Different franchises will see different account codes.

This now the official filter, according to this policy it is going to affect prices only on parts that exist in account code 101. Now that a filter is set up, we can look to the update portion of our sentence. The update portion reads, “I want to multiply their prices at four times my cost.” The goal here is to change the Selling Price to four times our cost.

These are the steps I take to form the update portion of my policy:

1. Click on the **ADJUSTED SELLING PRICE** drop box (Default says No Change)



A screenshot of three stacked dropdown menus. The first menu is labeled 'Adjusted Selling Price' and shows 'No Change'. The second menu is labeled 'Adjusted Selling Labor' and also shows 'No Change'. The third menu is labeled 'Adjusted Part Cost' and shows 'No Change'.

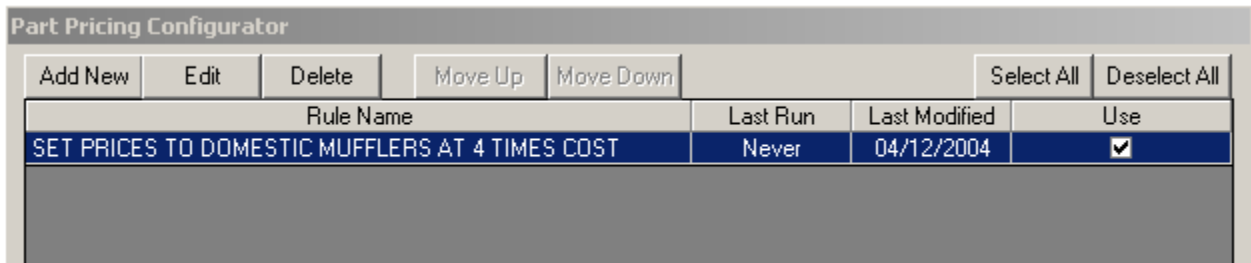
2. Select **MULTIPLY COST BY**.
3. **INPUT A VALUE** of 4.
4. Click the **SAVE** button.
5. This policy is now **READY TO BE RUN**.

TESTING YOUR PRICING POLICY

One important aspect of pricing policies is that they are very beneficial, but they also can be very damaging if they are done wrong. The next part of the process is reserved to protect the shop from any pricing wrongdoing. This button is the most important button in the entire module. This is the **TEST BUTTON**.

The one thing that needs to be stressed is to **TEST** the pricing policy. This is only way that it will be ensured that it is essentially correct. If the test button is not used and you run a new pricing policy without testing it, you can run the risk of having unnecessary changes to your pricing. In order to use the test button, please do the following:

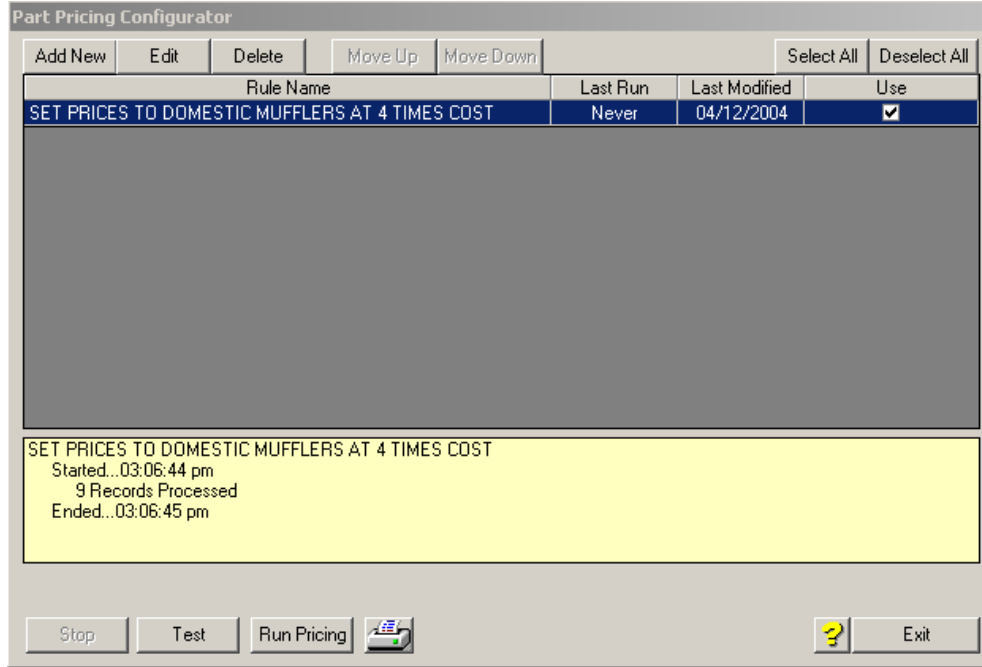
1. Click on the pricing policy to **HIGHLIGHT IT** and then **CLICK ON THE USE CHECKBOX**.



A screenshot of the 'Part Pricing Configurator' interface. It features a table with columns for 'Rule Name', 'Last Run', 'Last Modified', and 'Use'. The first row is highlighted in blue and contains the text 'SET PRICES TO DOMESTIC MUFFLERS AT 4 TIMES COST', 'Never', '04/12/2004', and a checked checkbox. Above the table are buttons for 'Add New', 'Edit', 'Delete', 'Move Up', 'Move Down', 'Select All', and 'Deselect All'.

Rule Name	Last Run	Last Modified	Use
SET PRICES TO DOMESTIC MUFFLERS AT 4 TIMES COST	Never	04/12/2004	<input checked="" type="checkbox"/>

- Click on the **TEST BUTTON**. A message will appear just above it.



The yellow text screen will show the date and time in which this policy is tested. To the right of the Test button, a picture will appear with numbers to the right of the picture. This picture is ramping up all the information related to the policy. Eventually the yellow text box will show how many records it processed and take you to the test results screen.

- The **TEST RESULTS SCREEN** will show you all the parts that the pricing policy will affect.

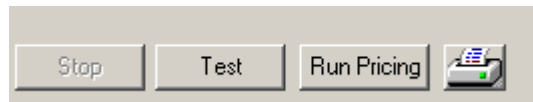
Part Number	Current Selling Price	New Selling Price	Current Labor Price	New Labor Price	Current Part Cost	New Part Cost	Current Core	New Core	Default Hours	New Hours	Pay Hours	New Pay	M
17694	87.85	91.08	0.00	0.00	22.77	22.77	0.00	0.00	0.00	0.00	0.0000	0.0000	0.0
17711	51.85	53.60	0.00	0.00	13.40	13.40	0.00	0.00	0.00	0.00	0.0000	0.0000	0.0
17712	55.85	57.92	0.00	0.00	14.48	14.48	0.00	0.00	0.00	0.00	0.0000	0.0000	0.0
17713	51.85	53.60	0.00	0.00	13.40	13.40	0.00	0.00	0.00	0.00	0.0000	0.0000	0.0
17714	55.85	57.92	0.00	0.00	14.48	14.48	0.00	0.00	0.00	0.00	0.0000	0.0000	0.0
17715	51.85	53.60	0.00	0.00	13.40	13.40	0.00	0.00	0.00	0.00	0.0000	0.0000	0.0
17716	72.85	75.60	0.00	0.00	18.90	18.90	0.00	0.00	0.00	0.00	0.0000	0.0000	0.0
17717	72.85	75.60	0.00	0.00	18.90	18.90	0.00	0.00	0.00	0.00	0.0000	0.0000	0.0
17718	72.85	75.60	0.00	0.00	18.90	18.90	0.00	0.00	0.00	0.00	0.0000	0.0000	0.0

This window will display all the old information and new information on Prices, Labor, Cost, Core Values, Work Hours, Minimum hours and Maximum hours. As with the example from earlier, the only change that should be shown is the change in price. Here we can compare the new price with the old. If the prices are suitable and no other changes need to be made, then it is time to actually run the pricing.

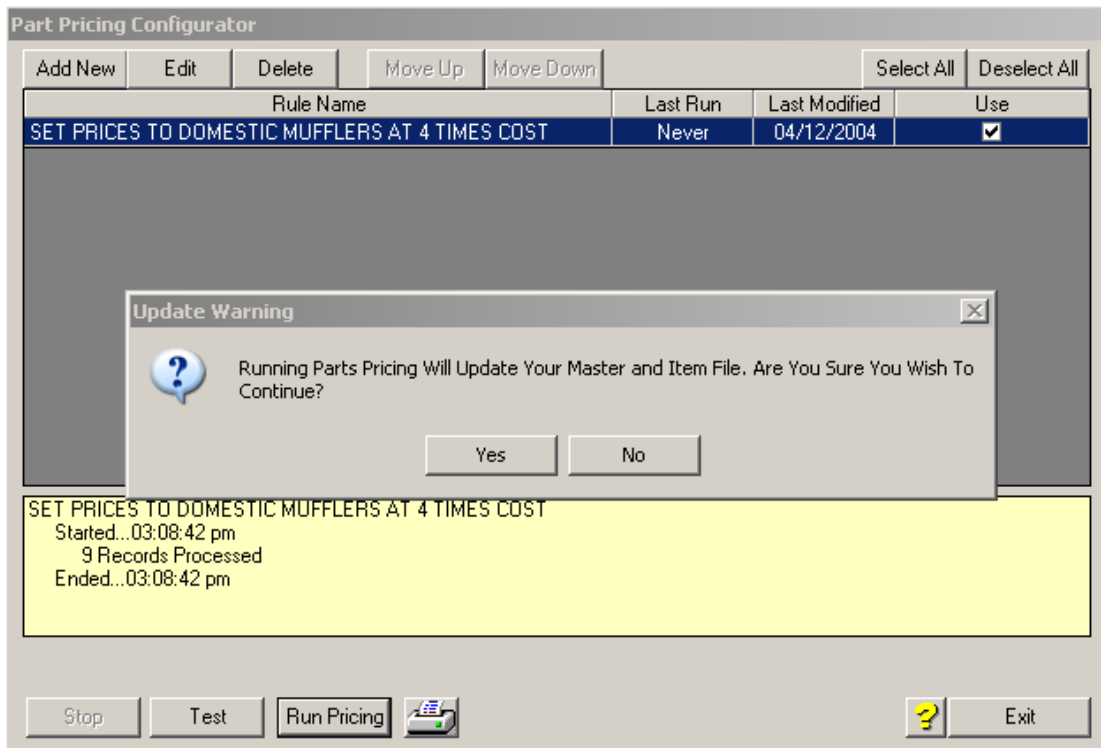
If some **CHANGES NEED TO BE MADE** and the policy is not correct, then we would exit out of the test results and **CLICK ON THE POLICY TO HIGHLIGHT IT** and click the **EDIT BUTTON**.

RUNNING YOUR PRICING POLICY

1. **CLICK THE USE BUTTON** and then **CLICK THE RUN PRICING** button.



2. Upon clicking the **RUN PRICING BUTTON**, a message will appear as follows:



By clicking **YES**, the pricing policy will begin to execute, by clicking **NO** the policy will just cancel until you either click Test or the Run Pricing button.

3. Just like before the yellow text message area will show the date and time of when the pricing policy was run and the picture will appear to the right of the Run Pricing button and ramp up the data. **THE PRICING IS NOW OFFICIALLY RUNNING.** The process runs into a total of three steps:
 - a. Data begins to ramp up and it processes the records to be written.
 - b. Updates the Master Table
 - c. Updates the Item Table.
4. When the Run Pricing is complete the process will show as being **FINISHED**, the **RUN PRICING BUTTON WILL GRAY ITSELF OUT** and the **USE BUTTON WILL BE UNCHECKED.**

PRICING POLICY TECHNIQUES

There are **TWO TECHNIQUES** that can be used to create pricing policies:

- a. **DIRECT RULE TECHNIQUE** - The Direct Rule Technique it, you create a policy and that's all that you do. (Example: Changing all your Selling prices cents to .95). Just one simple rule the will bring forth the desired result.
- b. **DIVIDE AND CONQUER** - Unlike the Direct Rule Technique, the Divide and Conquer breaks down the desired result one rule at a time. The good thing about the policies is that you can layer them as deep as you want. You can create 5 rules to make the changes you need as desired. Lets go back to the original example and change the English sentence differently.

“For all my Domestic Mufflers, I want to adjust my selling price to 4 times the cost. If any of my Domestic Mufflers are under 59.95, I want to those mufflers to be 59.95.”

In this particular case, one stand-alone rule will not work. This is a perfect candidate for the Divide and Conquer technique. As from our example, the multiplying the cost is already completed. In this case, we would need to make a rule that will change the price to 59.95 if we have any Mufflers that are less than that price. After the rule is created, we would want to run this policy immediately after we run the first rule. A good idea would keep it organized is to put Part 1 and Part 2 in the pricing description accordingly.

MULTIPLE PRICING RULES

Eventually there will come a time that all the policies will need to be run. They have been correct for many years and they have become accurate. Some questions do come up every now and then about testing multiple rules at once. Whenever multiple pricing policies are tested at once, the only results that will be seen are whichever policy is last on the list that is clicked for use. Please bear this in mind when testing, especially for multiple rules. It is best to test the pricing rules one at a time.

Order of pricing is a very important factor when updating your rules. That is why the Move Up and Move Down button exists. As an example of how multiple rules should be run, please keep in mind this general rule of thumb:

1. Run anything that multiplies selling price by cost
2. Run anything that increases percentage
3. Run anything that increases GM percentage
4. Run anything that replaces Dollar amount.
5. Run any policies that update work hours/times
6. Run any policies that sets the Add-On
7. Run the replace cents with Policy

One important thing to remember is to be careful of which rules will affect which parts. It wouldn't be a good idea to increase the percentage for account 101 and then increase the GM percentage for account 101 immediately after.

****NOTE:** Depending on how general the rule is or how many rules are being run, it is the possibility that the pricing update will take a very long time to run. Please consider that before you run multiple or general policies.

PRICING POLICY PRACTICE WORKSHEET

Below are several plain English sentences, analyze the sentences below and apply them into a pricing policy.

1. “For all my Brake Pads I want to set the Maximum Work hours to 1 and I want to replace the cents with .50.”
2. “For any thing that has Align in the description, I want to add twenty dollars to labor.”
3. “If I have any oil filters less than five dollars then I will change their amounts to 6 dollars.”
4. “It seems that my cost for my one tie rod is too high and I am selling a lot of them, I want to increase my Gross Margin for it by 10 percent.”
5. “All my Tail Pipes are need to have an add on of 5 dollars.”

Below is a complex rule written in an English sentence. Come up with a way get make this rule apply properly. (Hint: Use the Divide and Conquer technique.)

“For all my Remanufactured Calipers, I want to increase their selling price to 5 times the cost and I want to decrease there labor by 10 percent. If I have any more Calipers that are less than 44.99, I want to set their price to 45 dollars. If any of my Calipers are 45 dollars I want to set the work hours to 2 and increase the Labor by 10 dollars.”