

Executive Summary

Background

Matrix Scientific is a company based in South Carolina that buys chemicals in bulk from overseas and sells in smaller packages to research centers all throughout the United States. Matrix Scientific currently has over 100,000 different items in its inventory and competes with a handful of other companies in the market, both domestic and foreign.

Problem

Due to the amount of items Matrix Scientific sells, employees spend many hours looking up competitor pricing on the internet to ensure Matrix Scientific is offering the lowest prices. However, this time can be better spent on other tasks throughout the warehouse.

Solution

I wrote a system that automates the price comparison process by bringing in information from a website called discoverygate that contains the catalogs of all competing companies and informs Matrix Scientific, through color coding the spreadsheet, if they are offering the cheapest prices, as well as other bits of information. All the user needs to do is to push the “compare prices” button in the ribbon, enter the information in the prompts, and the information is readily available.

Detailed Write-up

Matrix Scientific is a company based in South Carolina that buys chemicals in bulk from overseas and sells in smaller packages to research centers all throughout the United States. Matrix Scientific currently has over 100,000 different items in its inventory and competes with a handful of other companies in the market, both domestic and foreign. Due to slow business in the past months, employees who were in charge of taking orders are now spending many hours in the day comparing the 100,000 product prices of Matrix Scientific to its main competitors.

The remaining pages will describe 1) my **solution** to making the process of informing Matrix Scientific when competitor's prices are cheaper, 2) the **help** I received during the project, and 3) **the things I learned**, as well as the difficulties I faced, while creating the solution

The Solution

The following is a quick overview of my solutions that I will expound upon throughout this section:

1. I brought information from **Discovery Gate** into Excel to compare current prices of competitors to Matrix Scientific.
2. I created an easy-to-use input box for **sign in** and decide how many items to loop through the code.
3. I made the system to inform the user about out-of-date CAS and MDL numbers.
4. I created a **color-coding system** and a **user form** that allows for clear analysis of results and an easy option to change prices.

For my final project, I automated the Matrix Scientific excel spreadsheet catalog so that it can pull in competitor pricing from the internet and show which products at what quantities (50mg, 100mg, 250mg, 500mg, 1g, 2.5g, 5g, 10g, 25, 50g, 100g, 250g, 500g, 1kg) are being offered for cheaper by competitors. Since prices of chemicals are changing frequently, automating the process of finding which products are being offered cheaper by competitors will save Matrix Scientific thousands of dollars in employee wages.

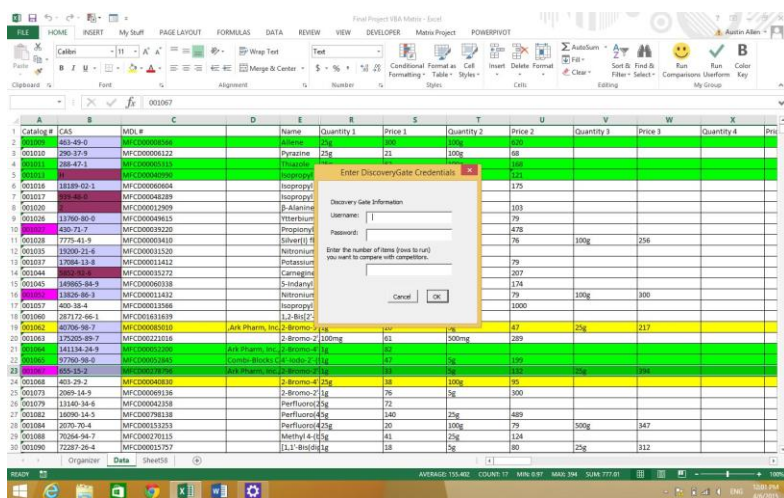
[illegible]

I will now explain from start to finish **how the user will use the system**.

After the user selects the row from which to start the price comparisons, the user

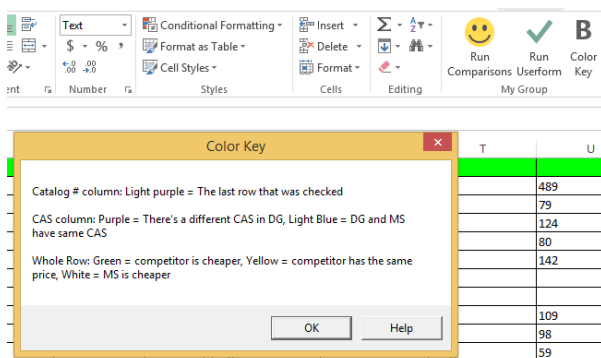
After the user selects the row from which to start the price comparisons, the user

will push the “Run Comparisons” button in the ribbon (smiley face icon in the top right). An input box will appear where the user will enter their Discovery Gate username, password, and the number of rows the user wishes to compare, as seen in the graphic below.



Comparing Prices

Using Dr. Allen’s “agent” class module, I wrote code that pulls information about the chemicals from the website into excel so that comparisons between prices can be made in excel and use color fills to inform different things about the searched chemical. The color legend is shown in a message box upon clicking the “Color Key” in the ribbon.



Below is an example of how the spreadsheet looks like after prices were compared and the color coding was finished.

Catalog #	CAS	MDL #	Name	Quantity 1	Price 1	Quantity 2	Price 2	Quantity 3	Price 3	Quantity 4	Price 4
001001	443-89-0	MFC0000000000	Pyrazine	25g	21	100g	88				
001002	290-37-9	MFC0000000000	Thiophene	25g	82	100g	166				
001003	288-47-1	MFC0000000000	Thiophene	25g	71	100g	166				
001004	18339-02-1	MFC0000000000	Isopropyl 45g	40	25g	175					
001005	339-46-0	MFC0000000000	Isopropyl 350g	79							
001006	13760-80-0	MFC0000000000	Isopropyl 350g	79							
001007	430-71-7	MFC0000000000	Isopropyl 350g	79							
001008	1773-41-9	MFC0000000000	Isopropyl 350g	79							
001009	1000-21-6	MFC0000000000	Nitrobenzene	170	25g	78	100g	256			
001010	17084-13-8	MFC0000000000	Potassium 100g	20	200g	79					
001011	1000-21-6	MFC0000000000	Carbazole 100g	100	200g	307					
001012	14885-94-9	MFC0000000000	Isopropyl 45g	52	2g	124					
001013	13885-86-1	MFC0000000000	Nitrobenzene	27	25g	79	100g	300			
001014	400-18-4	MFC0000000000	Isopropyl 100g	100	200g						
001015	28717-72-0	MFC0000000000	1,2-Bis(2-ethyl-5-hydroxy-2-methyl-1,3-dioxane-5-yl)ethane	79							
001016	40705-98-7	MFC0000000000	2-Bromo-2-iodoethane	47	2g	47	79g	117			
001017	17026-85-7	MFC0000000000	2-Bromo-2-iodoethane	47	2g	100g	289				
001018	14134-26-9	MFC0000000000	2-Bromo-2-iodoethane	47	2g	100g	289				
001019	17026-85-7	MFC0000000000	2-Bromo-2-iodoethane	47	2g	100g	289				
001020	655-15-2	MFC0000000000	2-Bromo-2-iodoethane	47	2g	100g	289				
001021	403-29-2	MFC0000000000	2-Bromo-2-iodoethane	47	2g	100g	289				
001022	2065-34-9	MFC0000000000	2-Bromo-2-iodoethane	47	2g	100g	289				
001023	13140-34-6	MFC0000000000	Perfluorooctane	72	2g	300					
001024	16950-14-5	MFC0000000000	Perfluorooctane	140	25g	489					
001025	2070-70-4	MFC0000000000	Perfluorooctane	29	100g	79	300g	347			
001026	70384-94-7	MFC0000000000	Methyl 4-ethyl-2-methyl-1,3-dioxane-5-carboxylate	41	25g	124					
001027	72287-26-4	MFC0000000000	1,1,1-Trichloro-2,2,2-trifluoroethane	18	2g	80	25g	112			

CAS and MDL numbers

In more detail, each chemical can be searched by either the MDL number or the CAS number. I am not sure what either of them mean, but my code searches Discovery Gate by MDL number, and if no results are shown, the chemical is then searched using the CAS number. Not only does Matrix Scientific want information regarding their prices, but to know if Matrix Scientific's catalog includes the correct CAS number, so I also wrote code to make sure that Matrix Scientific has the most current information regarding the correct CAS number (If the information is up to date, the cell in the CAS column is color filled light blue, otherwise dark maroon).

User form

Upon completion of comparing prices, the user then has the option of putting the data in a table to format the results and use the programmed user form to change prices if desired. The graphic shown below is an example of organizing the items from those items whose competitor's pricing is cheaper (green), then

competitor pricing is the same (yellow), then competitor's pricing is more expensive. The user is free to change prices, the catalog, CAS and MDL numbers, and use check boxes to highlight rows.

The screenshot shows a spreadsheet application with a table of chemical data. The table has columns for Catalog #, CAS, MDL #, Name, Quantity 1, Price 1, Quantity 2, and Price 2. The data is organized into rows, with some rows highlighted in green, yellow, and blue. A 'Matrix Prices' dialog box is open, allowing the user to edit prices for a specific catalog, CAS, and MDL. The dialog box includes input fields for Catalog No., CAS, and MDL, and a table of quantities and prices. Checkboxes for 'Competitor Cheaper' and 'Competitor Same Price' are also present. The spreadsheet application has a menu bar with 'Organizer', 'Data', and 'Sheet58'.

Catalog #	CAS	MDL #	Name	Quantity 1	Price 1	Quantity 2	Price 2
001009	463-49-0	MFCD00008566	Allene	25g	299	100g	620
001011	288-47-1	MFCD00005315					
001013	141134-24-9	MFCD00040990	Ark Ph				
001064	97760-98-0	MFCD00052200	Comb				
001065	2069-14-9	MFCD00069136	Comb				
001067	655-15-2	MFCD00278796	Ark Ph				
001079	13140-34-6	MFCD00042358	Ark Ph				
001010	290-37-9	MFCD00006122					
001016	18189-02-1	MFCD00060604					
001017	939-48-0	MFCD00048289					
001020	2	MFCD00012909					
001026	13760-80-0	MFCD00049615					
001027	430-71-7	MFCD00039220					
001028	7775-41-9	MFCD00003410					
001035	19200-21-6	MFCD00031520					
001037	17084-13-8	MFCD00011412					
001044	5852-92-6	MFCD00035272					
001045	149865-84-9	MFCD00060338					
001052	13826-86-3	MFCD00011432					
001057	400-38-4	MFCD00013566					
001060	287172-66-1	MFCD01631639					

The Help I Received

Using the agent class module proved to be a difficult task, and I had trouble navigating through Discovery Gate using just code. Therefore, Nathan, the TA, helped me write code that allows me to sign in, as well as push the “SUBMIT” and “SEARCH” buttons in Discovery Gate as shown in the below graphic.



Also, referring to the graphic below, if there is so much data that a scroll appears on the window, excel wouldn't input the data that hidden. I went to Dr. Allen for help, and after spending a few hours on the problem, we decided that the problem cannot be fixed; however, as a small solution, we decided to have internet explorer up on **full screen** so that more data can be shown in the window.

The screenshot displays the DiscoveryGate Sourcing interface. At the top, the DiscoveryGate logo is visible, along with navigation links like 'ABOUT', 'FORUM', 'HELP', and 'CONTACT US'. A search bar contains the identifier 'MFCD00085010'. Below the search bar, a 'Sourcing' section shows a chemical structure and a table of search results. The table lists various suppliers, including Ark Pharm, Inc., and provides details such as Catalog Number, Size, Price, Purity, and S/Gram. A sidebar on the left offers filters for 'Filter By Supplier Catalogs' and 'Filter By Supplier Type'. The right sidebar shows 'Properties & Other Data' and 'Chemical Names'.

Catalog Rating	Supplier Catalog	Catalog Number	Size	Price	Purity	S/Gram	Action
★★★★★	Ark Pharm, Inc.	AK-87704	100 MG		98 %		Add to Cart
★★★★★	Ark Pharm, Inc.	AK-87704	250 MG		98 %		Add to Cart
★★★★★	Ark Pharm, Inc.	AK-87704	100 G		98 %		Add to Cart
★★★★★	Ark Pharm, Inc.	AK-87704	25 G		98 %		Add to Cart
★★★★★	Ark Pharm, Inc.	AK-87704	500 G		98 %		Add to Cart
★★★★★	Ark Pharm, Inc.	AK-87704	1000 G		98 %		Add to Cart

Things I Learned

Among several things, the most important thing I learned is the importance of well documenting the code throughout the building process. This is the biggest programming project I've done, and several times I would be searching for what code performed what action, and that time would've been reduced dramatically if I organized and documented the code better. As a beginning programmer, I am happy I caught onto this mistake to help my efficiency on possible future project.

Learning how to more fully use the agent class module is another valuable skill I learned. I didn't know how to use it, but with the help of Nathan and Dr. Allen,

I was able to see how powerful the agent module is, and I was able to eventually learn how to use it. I am confident that I can use it for future projects.

The only real difficulties that I faced throughout the project was how to properly use arrays to search for a certain value in the excel sheet, and not being able to scroll down in Discovery Gate to retrieve the rest of the data. In the end, I decided to just copy and paste about 30 lines worth of code to take care of the array problem, and Dr. Allen's and my solution is described in the previous section.

Overall, this was a great capstone project for the class, and I was able to use a lot of skills that I learned throughout the semester. Although my coding skills may not be the fastest, or most efficient, compared to experienced programmers, I am confident I can solve problems with the skills I have been able to attain.