

Tommy Smith  
Professor Gove Allen  
I SYS 520  
December 7, 2010

## **Final Project: Stock Index Chart**

### EXECUTIVE SUMMARY

While working at an investment bank this summer, an organization that advises organizations on corporate finance issues, I had to pull stock information from the web and create graphs showing stock price performance of various industry segments to be used in presentations nearly daily. The graphs needed to be formatted with the bank's formatting and color schemes, so I couldn't just paste graphs from Yahoo or Capital IQ. I had to download data from the Internet to Excel and construct a chart by hand. Also, I had to combine the performance of any number of stocks to create customized industry segment performance indices. I therefore attempted to automate a program by recording macros that would pull information from Capital IQ and graph the data. However, with my limited knowledge of VBA, the program I created was inefficient, slow, and riddled with errors.

The new program I created pulls stock price information from Yahoo! and plots a line graph with stock performance of several industry segments against the performance of the S&P. The user selects which stocks to plot, which segments they are included in, and a time period through which to pull information. I will use this program extensively upon beginning work in June at an investment bank in New York City.

### WHAT THE USER SEES: EXPLANATION OF THE PROGRAM

Before code executes, the workbook is composed of three worksheets, one of which is hidden to the user. The first worksheet is the Inputs sheet, where the user enters the stocks to be graphed, the category titles, and the time period from which to pull data. On the next page is what the input sheet looks like, with example data filled in:

**INPUT SHEET** = Required Fields Dow Jones = ^DJ NASDAQ = ^IXC

STARTING Date:  Create Stock Index Graph This button graphs stock performance of indicated tickers against the performance of the S&P 500 Index for time period indicated.

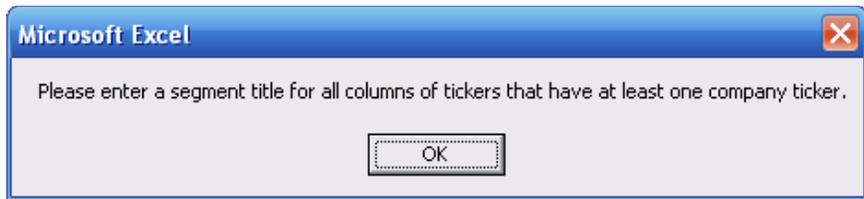
ENDING Date:   \*Takes 5 - 10 seconds per ticker to download data from web

Segment Title:

Tickers:

INPUTS Index Graph Index Data

If the required fields are not filled in correctly, error messages appear to inform the user, and the program stops. Below is an example of two such alerts:



The second sheet is the Index Data sheet, which is blank before the code executes for the first time. In this sheet, stock price data is compiled. Below is a screenshot of part of the Index Data worksheet after data has been compiled:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Date	S&P 500	S&P Index	GS	MS	CS	UBS	DB	JPM	C	GS Index	MS Index	CS Index	UBS Index	DB Index	JPM Index	C Index	Bulge Bra	PJC
2	12/6/2010	1223.12	-19%	162.65	25.54	38.61	15.77	50.73	39.9	4.45	76.94673	52.41125	67.87975	31.2401	38.55742	91.68199	13.72186	-47%	32.43
3	11/29/2010	1224.71	-19%	162.31	25.64	39.79	16.13	52.2	39.61	4.45	76.78588	52.61646	69.95429	31.95325	39.6747	91.01563	13.72186	-46%	31.96
4	11/22/2010	1189.4	-21%	157.87	24.7	38.33	15.36	50.55	37.5	4.11	74.6854	50.68746	67.38748	30.42789	38.42061	86.16728	12.67345	-49%	30.47
5	11/15/2010	1199.73	-20%	166.3	25.62	41.41	16.64	55.5	39.41	4.27	78.67348	52.57542	72.80239	32.96355	42.18287	90.55607	13.16682	-45%	31.38
6	11/8/2010	1199.21	-20%	165.47	25.57	42.4	17.22	55.23	39.61	4.29	78.28082	52.47281	74.5429	34.11252	41.97765	91.01563	13.22849	-45%	32.57
7	11/1/2010	1225.85	-19%	170.69	27.28	43.8	18.05	58.98	40.94	4.49	80.75031	55.98194	77.00422	35.75674	44.82785	94.07169	13.84521	-43%	34.52
8	10/25/2010	1183.26	-21%	160.78	24.87	41.5	17.02	57.88	37.63	4.17	76.06207	51.03632	72.96062	33.71632	43.99179	86.46599	12.85846	-46%	30.97
9	10/18/2010	1183.08	-21%	157.41	24.47	42	17.93	58.69	37.7	4.11	74.46778	50.21547	73.83966	35.51902	44.60743	86.62684	12.67345	-46%	29.18
10	10/11/2010	1176.19	-22%	150.36	24.97	43.84	17.79	56.67	37.15	3.95	71.13256	51.24153	77.07454	35.24168	43.07213	85.36305	12.18008	-46%	29.92
11	10/4/2010	1165.15	-23%	152.32	25.27	44.02	17.67	56.46	39.31	4.19	72.0598	51.85717	77.391	35.00396	42.91252	90.32629	12.92014	-45%	28.82
12	9/27/2010	1146.24	-24%	147.38	24.97	43.24	17.12	55.12	38.76	4.09	69.72277	51.24153	76.01969	33.91442	41.89405	89.0625	12.61178	-47%	29.39
13	9/20/2010	1148.67	-24%	146.96	25.1	44.44	17.81	55.71	39.7	3.9	69.52408	51.50831	78.1294	35.2813	42.34248	91.22243	12.0259	-46%	29.81
14	9/13/2010	1125.59	-25%	150.65	26.42	45.81	17.9	60.24	40.01	3.95	71.26975	54.21711	80.53797	35.45959	45.78551	91.93474	12.18008	-44%	29.23
15	9/7/2010	1109.55	-26%	150.06	27.13	45.15	17.86	60.5	39.71	3.91	70.99063	55.67412	79.37764	35.38035	45.98313	91.2454	12.05674	-44%	28.81
16	8/30/2010	1104.51	-27%	146.97	26.6	46.6	18.05	64.61	39.12	3.91	69.52881	54.5865	81.92686	35.75674	49.10694	89.88971	12.05674	-44%	29.86
17	8/23/2010	1064.59	-29%	139.09	24.98	43.99	16.76	64.12	36.55	3.76	65.80093	51.26206	77.33826	33.20127	48.73451	83.98438	11.5942	-47%	28.79
18	8/16/2010	1071.69	-29%	147.54	25.87	44	16.48	64.94	37.09	3.75	69.79847	53.08845	77.35584	32.64659	49.35776	85.22518	11.56337	-46%	28.24
19	8/9/2010	1079.25	-28%	147.38	25.85	44.35	16.43	67.13	37.45	3.88	69.72277	53.0474	77.97117	32.54754	51.02227	86.05239	11.96423	-45%	27.81
20	8/2/2010	1121.64	-25%	154.45	27.59	48.2	17.5	74.49	40.39	4.06	73.06746	56.6181	84.7398	34.66719	56.61625	92.8079	12.51927	-41%	31.06
21	7/26/2010	1101.6	-27%	150.11	26.93	45.37	16.97	70.24	40.23	4.1	71.01429	55.2637	79.76442	33.61727	53.38603	92.44026	12.64261	-43%	31.19
22	7/19/2010	1102.66	-27%	146.69	26.78	42.69	14.9	64.54	39.78	4.02	69.39635	54.95588	75.05274	29.51664	49.05374	91.40625	12.39593	-45%	29.6
23	7/12/2010	1064.88	-29%	145.48	24.64	40.65	14.14	61.23	38.95	3.9	68.82392	50.56433	71.46624	28.01109	46.53796	89.49908	12.0259	-48%	29.71
24	7/6/2010	1077.96	-28%	137.41	24.6	42.33	14.47	62.97	38.8	4.04	65.00615	50.48225	74.41983	28.66482	47.86045	89.15441	12.4576	-47%	30.82
25	6/28/2010	1022.58	-32%	130.46	22.74	38.1	13.19	56.5	35.78	3.79	61.71823	46.6653	66.98312	26.12916	42.94292	82.21507	11.68671	-52%	30.1
26	6/21/2010	1076.76	-28%	139	24.91	38.85	13.83	58.99	39.34	3.94	65.75835	51.11841	68.30169	27.39699	44.83545	90.39522	12.14924	-49%	34.81
27	6/14/2010	1117.51	-26%	137.53	25.6	41.08	14.36	61.98	39.08	4.01	65.06292	52.53437	72.22222	28.44691	47.108	89.79779	12.36509	-47%	34.55
28	6/7/2010	1091.6	-27%	135	25.9	39.45	13.08	58.59	37.99	3.88	63.86602	53.15001	69.35654	25.91125	44.53143	87.2932	11.96423	-49%	33.09
29	6/1/2010	1064.88	-29%	141.58	25.89	36.54	12.54	55.33	37.52	3.79	66.9789	53.12949	64.24051	24.84152	42.05366	86.21324	11.68671	-50%	32.09
30	5/24/2010	1089.41	-28%	143.58	27	38.69	13.33	59.57	39.48	3.96	67.92056	55.40735	68.02039	26.4065	45.27628	90.71691	12.21092	-48%	33.11
31	5/17/2010	1087.69	-28%	139.61	27	40.01	13.58	60.19	39.41	3.75	66.04693	55.10735	70.34107	26.90174	45.74751	91.7739	11.56337	-47%	33.38

The dollar closing price of each stock is pulled into the third worksheet, a hidden worksheet, from a .csv file downloaded from Yahoo! Finance. Those dollar figures are moved to the visible Index Data worksheet on the bottom of the previous page. In this particular example, the prices are moved to columns B (for the S&P) and D through J (for specific stocks) as seen in the figure above. The other columns are used to compute the return on the portfolio of stocks, as can be seen in columns C (for the S&P) and R (for the “Bulge Bracket Bank” segment). Columns K through Q compute the return on each individual stock, as a 100 point index, which is used in computing column R, the return for the entire portfolio.

Every time the code executes, a new chart is created in a fourth, newly-added worksheet called Index Graph. Below is the chart that the code creates using the inputs indicated in the previous pages:

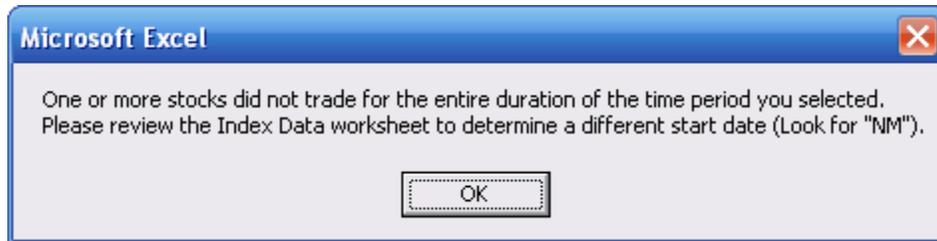


Each “index” is NOT capitalization weighted; that is, a company with a higher market capitalization does not receive more weight. The indices are computed as if an investor placed an equal amount of dollars into stocks of each company in each index. It then computes the return he or she would receive in the entire portfolio. For example, the “Mid Market Banks” index in the graph above includes the stock performance of Piper Jaffray, Jefferies, and Raymond James investment banks. The model assumes that an investor places a fixed, equal dollar amount into each stock, and measures the return for the entire

portfolio of the three stocks. The resulting return is shown in the green line in the chart above. The line chart is then ready to be pasted into a PowerPoint presentation, as needed by the user.

#### When a Stock Is Not Traded for the Entire Period

Sometimes, a specific stock will not have data available for the entire period a user has selected. This could be for various reasons, but is usually because the stock was not trading during the entire period. In that case, the following message box will appear:



The line chart will still be created. If one stock in a category did not trade for the entire time period, it is simply not included in the category return plotted on the chart. As can be seen in the blue columns below, PJC (Piper Jaffray) began trading during the week of December 29, 2003. Stock price information is in column B. Before that date, stock information was not available. In column E, where the PJC index is measured, "NM" is shown, which stands for "Not Meaningful." In column H, PJC is not included in the

	A	B	C	D	E	F	G	H
1	Date	PJC	JEF	RJF	PJC Index	JEF Index	RJF Index	Mid Market
351	3/29/2004	54.25	16.48	15.32	NM	170.777	121.395	46%
352	3/22/2004	54.5	16.38	14.96	NM	169.741	118.542	44%
353	3/15/2004	52.24	16.44	14.45	NM	170.363	114.501	42%
354	3/8/2004	55	16.97	15	NM	175.855	118.859	47%
355	3/1/2004	56.9	17.78	15.6	NM	184.249	123.613	54%
356	2/23/2004	51.52	17.08	15.39	NM	176.995	121.949	49%
357	2/17/2004	51.56	17.31	15.64	NM	179.378	123.93	52%
358	2/9/2004	50.6	17.16	15.41	NM	177.824	122.108	50%
359	2/2/2004	46.75	16.8	15.03	NM	174.093	119.097	47%
360	1/26/2004	46.95	17.48	15.32	NM	181.14	121.395	51%
361	1/20/2004	47.55	17.78	15.74	NM	184.249	124.723	54%
362	1/12/2004	46.17	16.83	15.47	NM	174.404	122.583	48%
363	1/5/2004	41.35	15.71	14.72	NM	162.798	116.64	40%
364	12/29/2003	43	15.29	14.95	NM	158.446	118.463	38%
365	12/22/2003		14.87	14.71	NM	154.093	116.561	35%
366	12/15/2003		14.83	14.69	NM	153.679	116.403	35%
367	12/8/2003		14.7	15	NM	152.332	118.859	36%
368	12/1/2003		14.75	15	NM	152.85	118.859	36%
369	11/24/2003		14.41	14.85	NM	149.326	117.67	33%
370	11/17/2003		14.19	14.64	NM	147.047	116.006	32%
371	11/10/2003		14.53	15.8	NM	150.57	125.198	38%
372	11/3/2003		15.39	16.34	NM	159.482	129.477	44%
373	10/27/2003		14.25	16.39	NM	147.668	129.873	39%
374	10/20/2003		13.42	15.96	NM	139.067	126.466	33%
375	10/13/2003		13.88	15.2	NM	143.834	120.444	32%

Mid Market Banks category, but the other stocks are included and graphed in the line chart.

If all the stocks in a given category are not traded for the duration of the dates selected, the same message box appears to inform the user to check the Index Data page, and no line is graphed. The user must choose a different starting date if he or she wants to measure the performance of that category.

#### The Reset Button

There is also a reset button on the Inputs sheet that clears all the sheets and deletes the chart. It is there for convenience, but it is not necessary every time a user wants to create a new graph. If a user simply fills in new inputs, the code will delete the previous chart, clear and refill the Index Data tab, and remake a new chart.

## WHAT THE USER DOES NOT SEE: THE CODE AND DIFFICULTIES IN WRITING IT

### *Downloading Stock Information*

What the user does not see is the downloading of data from Yahoo! Finance. When I started this project, I wanted to use the Capital IQ Excel plug-in to download the data since the formulas would be held within the actual spreadsheet and would have made the project much easier. Also, the user would be able to see exactly how the data was pulled. Investment bankers like to see everything so they can check each others' work, and because they all have very "red" personalities and like to be in control. However, CapIQ would not give me a free trial and BYU does not own any contracts with CapIQ. Therefore, I had to adapt some code we used in class to download data from Yahoo!, in which VBA pulls .csv files from Yahoo! Finance and places the data into Excel. This proved to be more difficult than simply using CapIQ, but also provided some benefits because I was forced to write a lot more code to populate cells instead of holding complex CapIQ formulas within cells. This resulted in a file much smaller than a CapIQ file would be.

The actual code which uploads the stock data is very similar to what we used in class, so I will not explain it here. To the right is a screenshot of the hidden spreadsheet in which the data is initially placed before being moved to the Index Data sheet:

Column H holds the adjusted closing prices for the share (adjusted for stock splits, dividends, and other events) and is the column that is moved to the Index Data sheet. Subsequent to the copying of column H, this page is cleared and data for the next stock on the Input sheet is downloaded. Again, this page is not visible to the user and is used simply to hold the downloaded stock information before pasting it into the Index Data sheet.

	A	B	C	D	E	F	G	H
1	Ticker	Date	Open	High	Low	Close	Volume	Adj Close
2	GHL	12/6/2010	78.08	78.1	77.52	77.74	61900	77.74
3	GHL	11/29/2010	75.39	78.18	74.31	78.06	196700	78.06
4	GHL	11/22/2010	76.1	76.82	75.06	76.13	201400	75.68
5	GHL	11/15/2010	79.38	79.63	76.34	76.5	251900	76.05
6	GHL	11/8/2010	81.1	82.45	78.69	78.96	200700	78.49
7	GHL	11/1/2010	77.99	81.59	76.05	81.36	229500	80.88
8	GHL	10/25/2010	78.21	78.34	74.21	77.67	296200	77.21
9	GHL	10/18/2010	76.02	79.06	74.87	76.91	189200	76.46
10	GHL	10/11/2010	79.57	79.7	75.05	76.13	240500	75.68
11	GHL	10/4/2010	78.8	79.78	77.13	79.41	236400	78.94
12	GHL	9/27/2010	79	80.81	78.02	79.4	204500	78.93
13	GHL	9/20/2010	81.84	82.39	76	78.78	254200	78.31
14	GHL	9/13/2010	78.37	82.01	77.93	81.68	321800	81.2
15	GHL	9/7/2010	76.6	78.46	75.83	77.92	158500	77.46
16	GHL	8/30/2010	73.29	76.95	70.25	76.89	229600	76.44
17	GHL	8/23/2010	72.93	74.05	70.27	73.72	293100	72.84
18	GHL	8/16/2010	65.56	72.57	65.37	72.2	256300	71.34
19	GHL	8/9/2010	69.91	70.52	64.85	65.88	139300	65.09
20	GHL	8/2/2010	68.99	70.14	67.66	69.4	138100	68.57
21	GHL	7/26/2010	68.5	70.33	67.83	68.05	208900	67.23
22	GHL	7/19/2010	64.64	68.39	61.96	67.86	270000	67.05
23	GHL	7/12/2010	67.56	68.44	64.3	64.35	228600	63.58
24	GHL	7/6/2010	63.93	67.71	62.56	67.71	260500	66.9
25	GHL	6/28/2010	62.35	64.45	60.8	62.5	394500	61.75
26	GHL	6/21/2010	63.81	65.61	60.73	62.11	474300	61.37
27	GHL	6/14/2010	65.89	66.29	62.4	63.48	240600	62.72
28	GHL	6/7/2010	65.88	66.32	62.56	65.3	300500	64.52
29	GHL	6/1/2010	68.81	69.66	65.33	65.6	308600	64.81
30	GHL	5/24/2010	69.55	70.85	65.5	69.29	386900	68.46
31	GHL	5/17/2010	74.3	75.25	67.88	69.91	431800	68.63

### *Creating the Chart*

The most difficult part of the project for me was figuring out how to create a dynamic chart that could be created no matter the size of the data field. I experimented a lot by recording myself selecting data fields, writing interesting formulas, etc. After a lot of thinking, searching online, and looking at some class examples, I determined that I could name ranges in VBA and use those names in the Series data name properties of a chart. The following code names the ranges of returns in the Index Data sheet AFTER the data is populated, with the following code:

```
indexData.Range(Cells(2, colPaste), Cells(dataNum, colPaste)).Name = "stockIndex" & grIndexNum
```

This code names each range of cells used in the graph, "stockIndex1", "stockIndex2", "stockIndex3", etc., for any number of categories selected on the Input tab (colPaste, dataNum, and grIndexNum are variables used in the many loops I use in the code).

After naming the range, VBA uses the following code to create each series within a line chart (all within a loop):

```
.SeriesCollection(x + 2).Values = "'Index Data'!" & "stockIndex" & x + 1
```

This line of code gives values to as many series in the chart as there are named ranges. Therefore, the chart graphs as many lines as there are named ranges in the Index Data sheet.

Further, the default chart type created by Excel did not appear as I wished. When percentages are on the vertical axes, for some reason Excel sets the horizontal axis to cross the vertical axis on the 0% y-value. I had to solve this problem by adding code that would set the minimum chart value to around 10% lower than the minimum of all the data included in the chart, and then set the value at which the horizontal axis crosses the vertical axis at that same value. That way, the horizontal axis is always at the bottom. I used a similar process to set the maximum chart value. The code is shown below. hAxis is variable that holds the minimum data value in the chart series, and vAxisMax holds the maximum value.

```
.Axes(xlValue).CrossesAt = Round(hAxis - 0.1, 1)  
.Axes(xlValue).MinimumScale = Round(hAxis - 0.1, 1)  
.Axes(xlValue).MaximumScale = Round(vAxisMax + 0.1, 1)
```

The remainder of the chart is in the default Excel formatting.

## CONCLUSION

This program will be of great use to me and my fellow analysts at the investment bank I will start working for in June. I anticipate we will use it almost daily and that it will save hours in manually finding such stock information and creating charts by hand. When I arrive, I can easily alter the code to change the color scheme and font to the company standard.

The programming subject I learned the most about in this project is looping. I created seven or eight different loops in this code, including "for each next" loops and "do" loops. I had to define a lot of variables to use in the loops. Before this class, I knew almost nothing about loops in code, and I now know it is a basic element of writing programs and necessary for this program. I now feel very comfortable with loops and could implement them in future programs I create.