ARGUS
SOFTWARE

Argus Translator 1.0

David Malone

Executive Summary

I have been working at Utah Retirement Systems on the real estate investment team for approximately 6 months. One of my assignments has been to assess the risk within the real estate portfolio of the overall state pension fund. Monte Carlo simulation is one of the tools that I and my team have chosen to use to assess risk.

Argus DCF is a software platform used to build financial models for various types of commercial real estate properties. A single Argus file contains significant amounts of data related to a single property, which is then used to build a discounted cash flow schedule and determine the IRR associated with those cash flows.

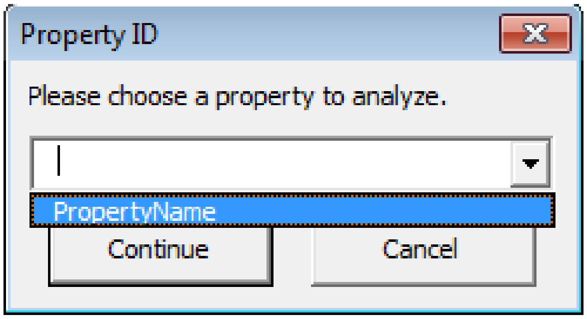
Though it is considered an industry standard software platform, it is not without its weaknesses. One of these is its Monte Carlo simulation engine, which is neither as fast nor as powerful as Excel add-ins such as Risk Solver and ModelRisk. While Argus will export its assumptions and results into Excel, the exported data are in a static format. In order for this exported data to be useful, it is necessary to reconnect the assumptions to the results using formulas in order to create a dynamic model. Only then can the model be used in Monte Carlo simulations. This requires a significant time commitment, often several days, for each property. With a portfolio in excess of 300 properties, the time commitment required to accurately assess portfolio risk is prohibitive.

The program I have designed pulls data from an Access database containing Argus data and then builds a dynamic cash flow model in Excel that is simulation-ready. I will further discuss this program and its functionality in the pages that follow.

Implementation

The code contained in this VBA project performs the following functions:

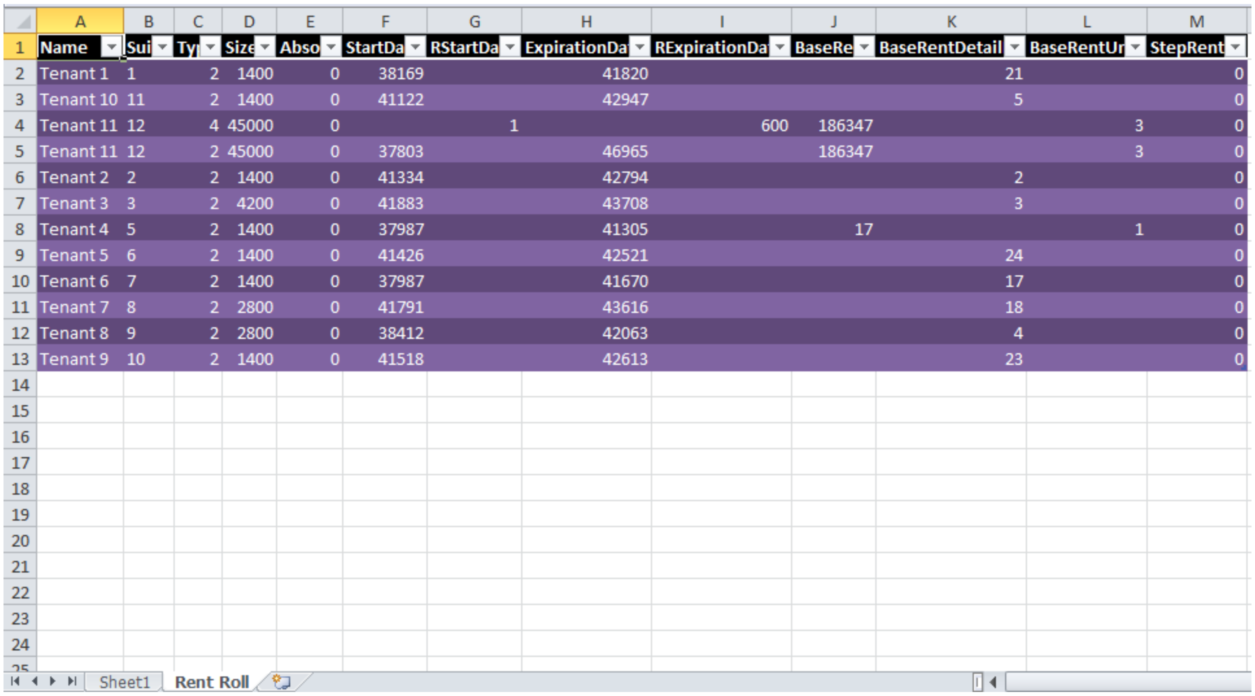
1. Pulls a list of all properties that currently have data in the Access database for the user to choose from, as shown here (property specific information has been disguised):



Property ID

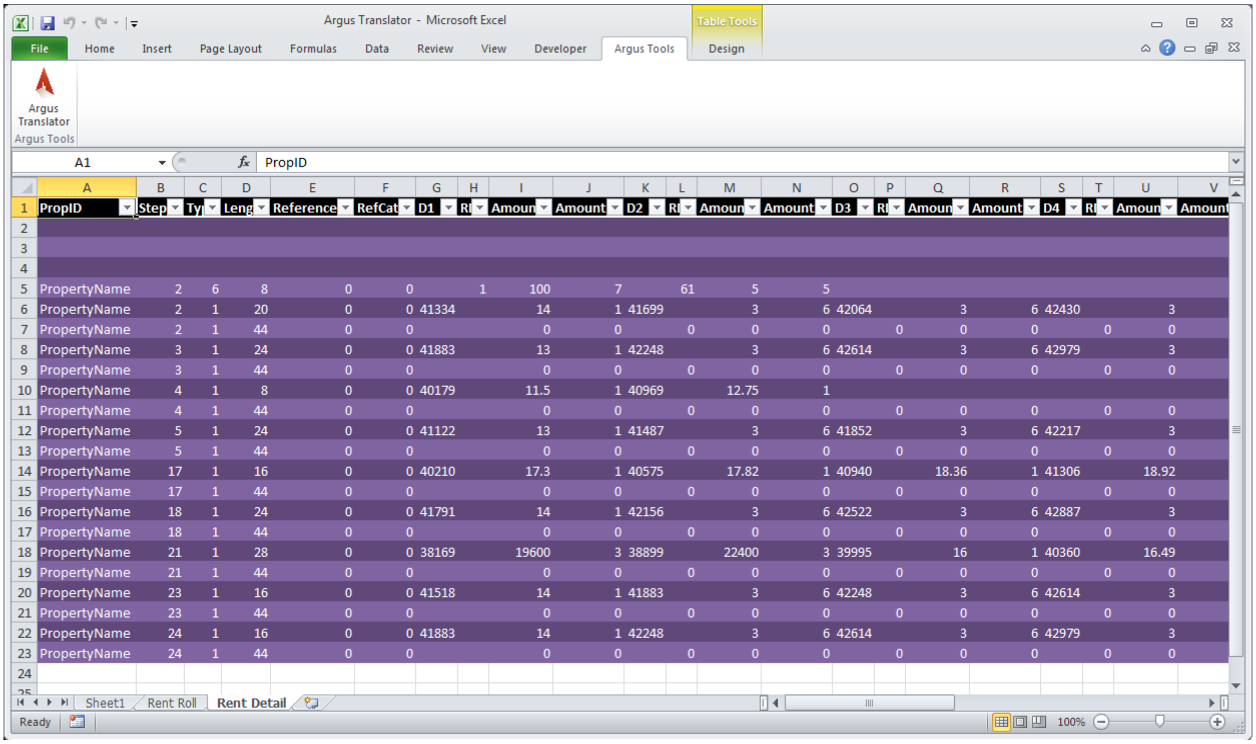
Please choose a property to analyze.

2. Pulls the tenant-specific rent data associated with the property chosen in step 1, as shown here:



Name	Sui	Ty	Size	Abso	StartDa	RStartDa	ExpirationDa	RExpirationDa	BaseRe	BaseRentDetail	BaseRentUr	StepRent
Tenant 1	1	2	1400	0	38169		41820			21		0
Tenant 10	11	2	1400	0	41122		42947			5		0
Tenant 11	12	4	45000	0		1		600	186347		3	0
Tenant 11	12	2	45000	0	37803		46965		186347		3	0
Tenant 2	2	2	1400	0	41334		42794			2		0
Tenant 3	3	2	4200	0	41883		43708			3		0
Tenant 4	5	2	1400	0	37987		41305		17		1	0
Tenant 5	6	2	1400	0	41426		42521			24		0
Tenant 6	7	2	1400	0	37987		41670			17		0
Tenant 7	8	2	2800	0	41791		43616			18		0
Tenant 8	9	2	2800	0	38412		42063			4		0
Tenant 9	10	2	1400	0	41518		42613			23		0

3. Pulls additional rent data saved in a different table, as shown here:



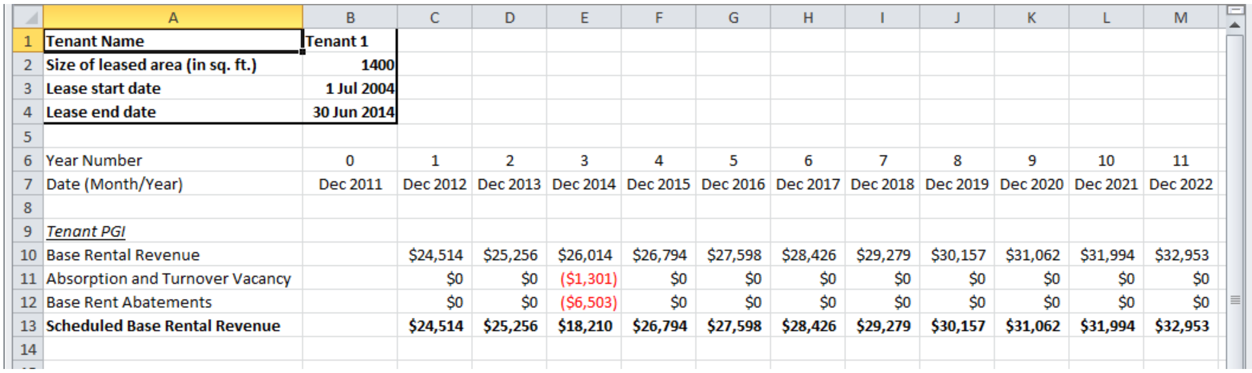
A1 | PropID

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	PropID	Step	Ty	Leng	Reference	RefCat	D1	Rl	Amoun	Amount	D2	Rl	Amoun	Amount	D3	Rl	Amoun	Amount	D4	Rl	Amoun	Amount
2																						
3																						
4																						
5	PropertyName	2	6	8	0	0		1	100	7		61	5	5								
6	PropertyName	2	1	20	0	0	41334		14	1	41699		3	6	42064		3	6	42430		3	
7	PropertyName	2	1	44	0	0			0	0		0	0	0		0	0	0		0	0	
8	PropertyName	3	1	24	0	0	41883		13	1	42248		3	6	42614		3	6	42979		3	
9	PropertyName	3	1	44	0	0			0	0		0	0	0		0	0	0		0	0	
10	PropertyName	4	1	8	0	0	40179		11.5	1	40969		12.75	1								
11	PropertyName	4	1	44	0	0			0	0		0	0	0		0	0	0		0	0	
12	PropertyName	5	1	24	0	0	41122		13	1	41487		3	6	41852		3	6	42217		3	
13	PropertyName	5	1	44	0	0			0	0		0	0	0		0	0	0		0	0	
14	PropertyName	17	1	16	0	0	40210		17.3	1	40575		17.82	1	40940		18.36	1	41306		18.92	
15	PropertyName	17	1	44	0	0			0	0		0	0	0		0	0	0		0	0	
16	PropertyName	18	1	24	0	0	41791		14	1	42156		3	6	42522		3	6	42887		3	
17	PropertyName	18	1	44	0	0			0	0		0	0	0		0	0	0		0	0	
18	PropertyName	21	1	28	0	0	38169		19600	3	38899		22400	3	39995		16	1	40360		16.49	
19	PropertyName	21	1	44	0	0			0	0		0	0	0		0	0	0		0	0	
20	PropertyName	23	1	16	0	0	41518		14	1	41883		3	6	42248		3	6	42614		3	
21	PropertyName	23	1	44	0	0			0	0		0	0	0		0	0	0		0	0	
22	PropertyName	24	1	16	0	0	41883		14	1	42248		3	6	42614		3	6	42979		3	
23	PropertyName	24	1	44	0	0			0	0		0	0	0		0	0	0		0	0	
24																						

4. Associates the additional rent data pulled in step 3 with the correct tenant data pulled in step 2

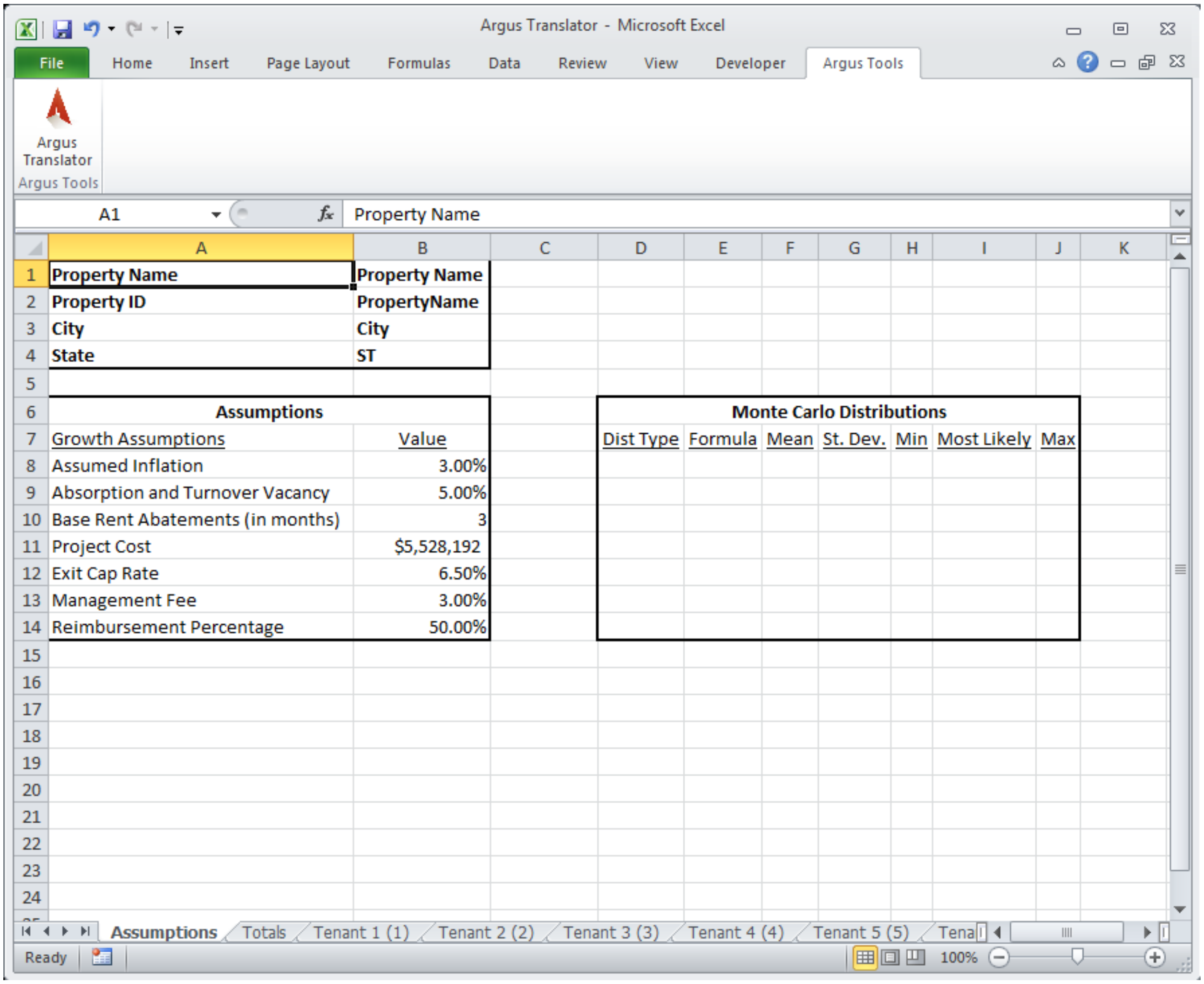
5. Saves this rent data in an array, which is referred to throughout the rest of the program

6. Adds a tenant template worksheet for and builds an individual cash inflow stream for each tenant, as shown here:



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Tenant Name**	**Tenant 1**											
2	**Size of leased area (in sq. ft.)**	**1400**											
3	**Lease start date**	**1 Jul 2004**											
4	**Lease end date**	**30 Jun 2014**											
5													
6	Year Number	0	1	2	3	4	5	6	7	8	9	10	11
7	Date (Month/Year)	Dec 2011	Dec 2012	Dec 2013	Dec 2014	Dec 2015	Dec 2016	Dec 2017	Dec 2018	Dec 2019	Dec 2020	Dec 2021	Dec 2022
8													
9	*Tenant PGI*												
10	Base Rental Revenue		$24,514	$25,256	$26,014	$26,794	$27,598	$28,426	$29,279	$30,157	$31,062	$31,994	$32,953
11	Absorption and Turnover Vacancy		$0	$0	($1,301)	$0	$0	$0	$0	$0	$0	$0	$0
12	Base Rent Abatements		$0	$0	($6,503)	$0	$0	$0	$0	$0	$0	$0	$0
13	**Scheduled Base Rental Revenue**		**$24,514**	**$25,256**	**$18,210**	**$26,794**	**$27,598**	**$28,426**	**$29,279**	**$30,157**	**$31,062**	**$31,994**	**$32,953**
14													

7. Adds an assumptions template and populates it, including adding named ranges to aid in calculation, as shown here:



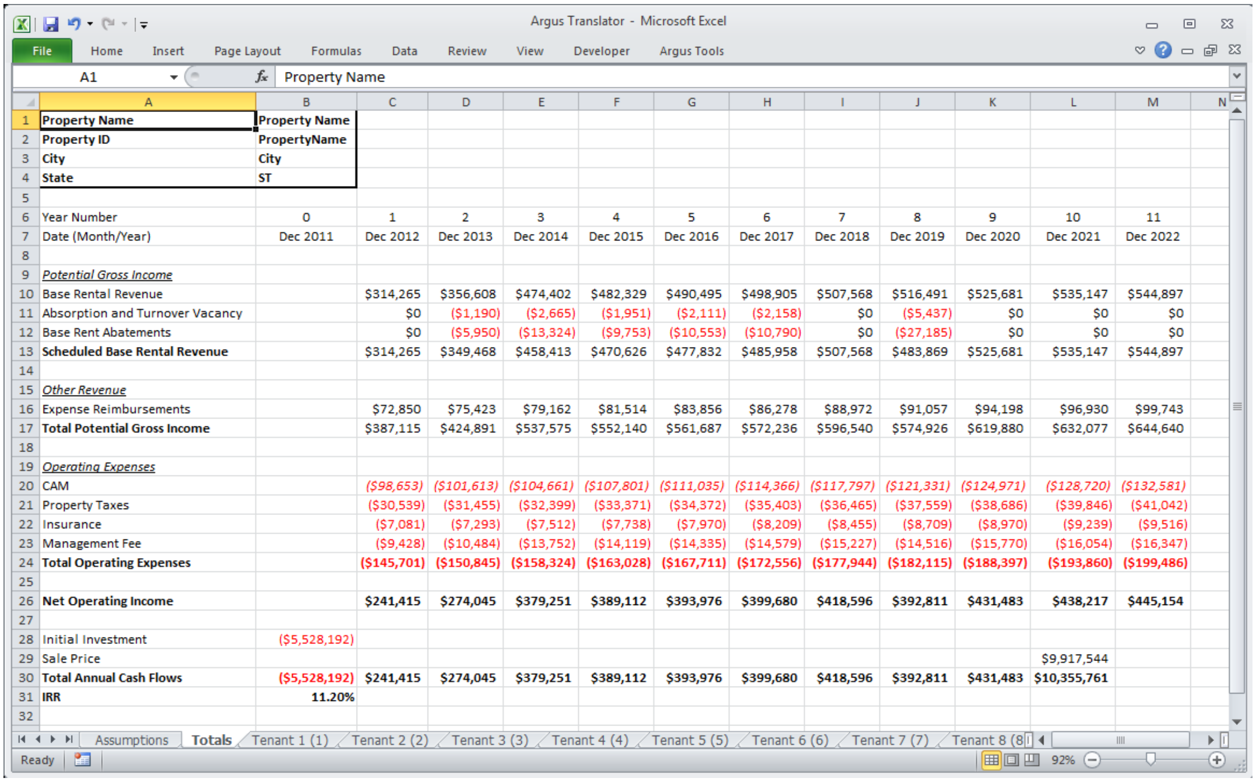
A1: Property Name

	A	B	C	D	E	F	G	H	I	J
1	**Property Name**	**Property Name**								
2	**Property ID**	**PropertyName**								
3	**City**	**City**								
4	**State**	**ST**								
5										
6	**Assumptions**			**Monte Carlo Distributions**						
7	Growth Assumptions	Value		Dist Type	Formula	Mean	St. Dev.	Min	Most Likely	Max
8	Assumed Inflation	3.00%								
9	Absorption and Turnover Vacancy	5.00%								
10	Base Rent Abatements (in months)	3								
11	Project Cost	$5,528,192								
12	Exit Cap Rate	6.50%								
13	Management Fee	3.00%								
14	Reimbursement Percentage	50.00%								

8. Pulls property-level expenses from the Access database and saves the necessary values

9. Adds a totals template and populates it with the revenue data from each of the tenant worksheets as well as adds and calculates the property-level expenses

10. Adds some formatting to the various worksheets to present a more polished look, as shown here:



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Property Name**	**Property Name**											
2	**Property ID**	**PropertyName**											
3	**City**	**City**											
4	**State**	**ST**											
5													
6	Year Number	0	1	2	3	4	5	6	7	8	9	10	11
7	Date (Month/Year)	Dec 2011	Dec 2012	Dec 2013	Dec 2014	Dec 2015	Dec 2016	Dec 2017	Dec 2018	Dec 2019	Dec 2020	Dec 2021	Dec 2022
8													
9	*Potential Gross Income*												
10	Base Rental Revenue		$314,265	$356,608	$474,402	$482,329	$490,495	$498,905	$507,568	$516,491	$525,681	$535,147	$544,897
11	Absorption and Turnover Vacancy		$0	($1,190)	($2,665)	($1,951)	($2,111)	($2,158)	$0	($5,437)	$0	$0	$0
12	Base Rent Abatements		$0	($5,950)	($13,324)	($9,753)	($10,553)	($10,790)	$0	($27,185)	$0	$0	$0
13	**Scheduled Base Rental Revenue**		$314,265	$349,468	$458,413	$470,626	$477,832	$485,958	$507,568	$483,869	$525,681	$535,147	$544,897
14													
15	*Other Revenue*												
16	Expense Reimbursements		$72,850	$75,423	$79,162	$81,514	$83,856	$86,278	$88,972	$91,057	$94,198	$96,930	$99,743
17	**Total Potential Gross Income**		$387,115	$424,891	$537,575	$552,140	$561,687	$572,236	$596,540	$574,926	$619,880	$632,077	$644,640
18													
19	*Operating Expenses*												
20	CAM		*($98,653)*	*($101,613)*	*($104,661)*	*($107,801)*	*($111,035)*	*($114,366)*	*($117,797)*	*($121,331)*	*($124,971)*	*($128,720)*	*($132,581)*
21	Property Taxes		($30,539)	($31,455)	($32,399)	($33,371)	($34,372)	($35,403)	($36,465)	($37,559)	($38,686)	($39,846)	($41,042)
22	Insurance		($7,081)	($7,293)	($7,512)	($7,738)	($7,970)	($8,209)	($8,455)	($8,709)	($8,970)	($9,239)	($9,516)
23	Management Fee		($9,428)	($10,484)	($13,752)	($14,119)	($14,335)	($14,579)	($15,227)	($14,516)	($15,770)	($16,054)	($16,347)
24	**Total Operating Expenses**		**($145,701)**	**($150,845)**	**($158,324)**	**($163,028)**	**($167,711)**	**($172,556)**	**($177,944)**	**($182,115)**	**($188,397)**	**($193,860)**	**($199,486)**
25													
26	**Net Operating Income**		**$241,415**	**$274,045**	**$379,251**	**$389,112**	**$393,976**	**$399,680**	**$418,596**	**$392,811**	**$431,483**	**$438,217**	**$445,154**
27													
28	Initial Investment	($5,528,192)											
29	Sale Price											$9,917,544	
30	**Total Annual Cash Flows**	**($5,528,192)**	**$241,415**	**$274,045**	**$379,251**	**$389,112**	**$393,976**	**$399,680**	**$418,596**	**$392,811**	**$431,483**	**$10,355,761**	
31	**IRR**	**11.20%**											
32													

The data required to perform this analysis was pushed to Access directly from Argus, which created over 100 tables. This data is not stored in an easily digestible format. In order to build the discounted cash flow model, significant amounts of data need to be pulled into Excel from this Access database. This data then needs to be scrubbed and analyzed to ensure that only relevant data is being used and to guarantee that this data is properly assigned to the correct tenants. The program needs to be able to recognize the different nuances associated with each tenant in order to calculate the cash flows associated with that tenant correctly and independently of how it calculates another tenants' cash flows. This was accomplished by building various cases into the code so that it would react based on the data held within the Access tables. In this way, I was able to "teach" the program how to handle the different values for each tenant.

Once the code has finished executing, the cash flow model is ready to be utilized in Monte Carlo analysis. The model is fully dynamic, meaning that changing a value on the assumptions worksheet will alter the results in the rest of the model. The distributions for Monte Carlo analysis of these assumptions can be easily added manually to the assumptions worksheet. (This task will still have to be performed manually due to the differences in assumptions across property types, locations, and throughout time.)

Learning and Conceptual Difficulties

I learned the following from this project:

1. How to write the code to be able to handle multiple cases and scenarios – The revenues from each tenant needed to be treated slightly differently in order to get the desired results from the data in the OARentRoll and OASteppedInput tables.
2. How to build the code to react to different amounts of data – The rent roll data and other tables may have fixed widths, but the number of rows for each property will be different. I attempted to design the code to be able to respond to the variable nature of real estate analysis.
3. How to better consolidate repeated sections of code – I attempted to use one section of code for repeated actions as much as possible by writing one piece of code and passing it different values to make it react slightly differently. For example, the AddSheet sub procedure accepts a sheet name as well as a numeric value to determine whether to add a blank worksheet or a template worksheet.

4. How to add and use named ranges via VBA – Using named ranges drastically simplified the formulas required to calculate the values on the Totals worksheet as well as each of the individual tenant worksheets.
5. How to connect to databases – I originally tried to pull the data I needed by using an ActiveX Data Objects database connection. This method failed to produce results, however, due to the fact that several of the columns I was attempting to extract data from included reserved words in Access, such as the word "size." As a result, I built the necessary queries using Microsoft Query Wizard and then modified the code. I was later able to use the ADOdb connection to pull some other additional data.

The most important thing I learned through this project is the importance of focusing on getting the computer to complete just one task at a time. The programming for the overall cash flow model seems very complicated when viewed as a whole, but breaking the project down into various individual tasks makes execution much easier to achieve, though I recognize that the code I have written may not be the shortest or best way to perform these tasks.

The financial model that this VBA code produces is also very basic. As I mentioned before, Argus stores data in over 100 tables, some of which is irrelevant data to this analysis, but I have surely overlooked some very important information in the 95 other tables that I did not query. Furthermore, this code was written for and has only been used on one property. No two properties are the same, meaning that the code as written will be insufficient to handle all of the nuances of other properties. However, before I can improve this code I will need to further analyze the Access database that Argus exports in order to rewrite the code to handle these various new exceptions. I plan to continue to develop this project with the hope of improving and simplifying the code as I build in increased functionality for use with other properties.

Assistance

I did not receive direct help from any other person in writing the code for this project. I did utilize Google and the responses to others' questions on forums to guide me when I got stuck, but I did not post any of my own questions or seek the help of anyone in person or otherwise for this project. In addition, I previously had some basic experience using VBA to manipulate text in Microsoft Word. This course is the first time that I have used VBA in Excel.