



Why Should I Pay You for This Software? It's Just an Excel Template, Right? I Can Do That Myself!

By Frank Gallinelli

Lots of folks can play Chopsticks on the piano with two fingers. But would you pay for orchestra seats at Carnegie Hall to listen to them? Probably not. When you put cash on the table, you expect and deserve a professional performance.

It's possible to develop Excel-based applications that range in sophistication from the very basic to the highly polished and automated – and at almost every imaginable level in between. A reasonably proficient end-user of Excel can easily put together a model that looks presentable and performs some basic math – summing columns or rows, compounding values – and pulls data together from different worksheets. A more experienced user can expand a model's usefulness with conditional statements, financial functions and perhaps even some simple macros.

A professional software developer can employ complex conditional formulas, customized user forms, menus and toolbars, databases, data validation and Visual Basic modules that greatly enhance an Excel model's functionality and trap out potential user errors. When a model reaches this level it really becomes not so much a template in the way most users understand the term but more of a custom program that uses Excel as its underlying engine.

RealData's products – and in particular our flagship program, *Real Estate Investment Analysis, Standard Edition* – fall into the category of professional software, not simple or mid-level templates. They represent thousands of hours of the combined efforts of professionals in both the fields of real estate and programming. In trying to decide whether professional software is what you need, you should first consider what it is you want to accomplish and how you want to work. And maybe most importantly, you should consider just how valuable your own time is.

Let's examine three possible scenarios.

The Basic, End-User Template

You're a beginner in real estate and have a passing familiarity with the field's specialized terminology. You want to examine some small properties – maybe multi-families – to get an idea of what they're worth and how much cash flow they might generate. The good news is that creating a template to perform a very basic analysis is not terribly complicated. The bad news is that you need a good understanding of the underlying concepts in order to design the analysis correctly. What, for example, should be included in the calculation of Net Operating Income and what should not? Ask the same question about taxable income, cash flow, potential sale proceeds and financing.

To provide an alternative to this kind of basic, end-user model we recently released a low-cost *Ultra-Lite* edition of our investment analysis. It's not particularly complex, but if you're accustomed to compensation greater than the minimum wage, it's still not worth your while to try to reinvent it. With this program you have the assurance that it's correct (if you've made any Excel models of your own, you know how easy it is to leave a big "whoops" that compromises your results). You also have some nice special features, such as pre-formatted reports and creation of Adobe Acrobat® files, that you would not have attempted on your own.

The Mid-Level Commercial Model

You're familiar with real estate investing and need a good cash flow or resale analysis to evaluate potential deals that are not too complex. You'll probably come across some mid-level templates designed to tackle this task. By "mid-level" we mean that they're more polished than something a typical end-user might construct but not as robust as a true commercial software package.

Typically such a template is the product of a real estate investor who has created something for his or her own use and thinks it might be salable to others. We certainly can't criticize that idea – it's how we started in 1981. It's a good start but there are some shortcomings that often characterize programs in this category. Because the program is usually someone's part-time venture, support tends to be spotty, documentation light or non-existent and upgrades or maintenance infrequent. You're also likely to observe that a program like this may take some shortcuts: Does it overlook the half-month convention for depreciation? Does it ask you to specify a single capital gain tax rate (when in fact there are currently several rates that might apply in future years – different rates in different years)?



Why Should I Pay You for This Software? It's Just an Excel Template, Right? I Can Do That Myself!

By Frank Gallinelli

(Page 2 of 5)

The user interface tends to be confusing. That's not uncommon when someone develops a program for his or her own use (hey, looks fine to me) and doesn't have many independent users to provide feedback. In our 20+ years, we've been told more than a few times what makes sense, what looks good and what doesn't. Fix it or perish.

Finally, there is the issue of reporting. It took us several years (back in the 80's) before we came to appreciate that it's not enough just to calculate the numbers. Figuring the results is only part of the mission. Conveying the results is the other part. A good investment or development program needs to produce attractive and readable reports – reports that will make a project's value clear to a client or potential investor, or supply the evidence needed to close a loan. For too many software developers, a program's output is little more than an afterthought. With templates in particular, the solution is often “just print the spreadsheet.” If you really want to see a client's eyes glaze over, hand him or her a page filled with amoeba-sized numbers. When your investment presentation looks like a slide from your high-school biology class, you know you're in trouble.

Again, we can suggest an alternative from our own product line. Several years ago we released a *Lite* edition of our investment analysis. It's positioned both in cost and functionality between our *Ultra-Lite* program and our *Real Estate Investment Analysis Standard Edition, Version 12.0*. It avoids the key pitfalls described above, is easy to use, provides a solid analysis of residential or small commercial income property and produces excellent reports.

Professional Excel-Based Applications

We return now to our original bias: professional investment analysis software. If you're planning to buy or to develop an income property – an investment that in all likelihood will involve at least several hundred thousand and quite possibly millions of dollars – then your best interests are served by using quality decision-making tools, i.e., professional, time-tested analysis software. If you're the seller, you want to be able to defend your asking price with hard numbers. If you're the buyer, you want to make an intelligent and informed estimate of the property's value; and if that happens to be significantly less than the asking price, you also want to be able to show the seller why your offer is reasonable. No matter which side you're on, when you finally strike a deal, you want to convince a lender that it makes sense to ante up the mortgage financing.

How can an application built on Microsoft Excel achieve that lofty estate? After all, isn't it just a template, just a collection of rows and columns of numbers and formulas?

In a word, no – or at least it doesn't have to be. As we'll discuss shortly, the standard edition of our *Real Estate Investment Analysis, Version 12* provides an example of the kind of advanced functionality that a software developer can achieve with Excel.

An appropriate prior question, however, ought to be, “Why use Excel at all? Why not simply develop “standalone” applications?” The reasons are important and they benefit both you as the customer and us the developers. Virtually everyone who does any serious number-crunching (and real estate investors fall squarely into that category) owns a copy of Excel. By using Excel, we achieve two very important goals:

- 1) We give the user a familiar and intuitive working environment, one that he or she doesn't have to figure out from scratch and
- 2) We eliminate the cost of re-creating everything that Excel can do – the underlying user interface, file handling, math and logical functions, etc. If we had to write all of that to build a standalone program, the cost would be enormous. With a niche product we won't sell millions of copies as Microsoft does with Excel, so dividing the cost among mere thousands of users would make our program prohibitively expensive. It would also make it costly and time-consuming to upgrade, so new features would become available less frequently. By using Excel as the engine, everyone comes out ahead.

At last we get to the point: What kinds of things do we do to make our application more than just a template?

Why Should I Pay You for This Software? It's Just an Excel Template, Right? I Can Do That Myself!

By Frank Gallinelli

(Page 3 of 5)

1. Structure

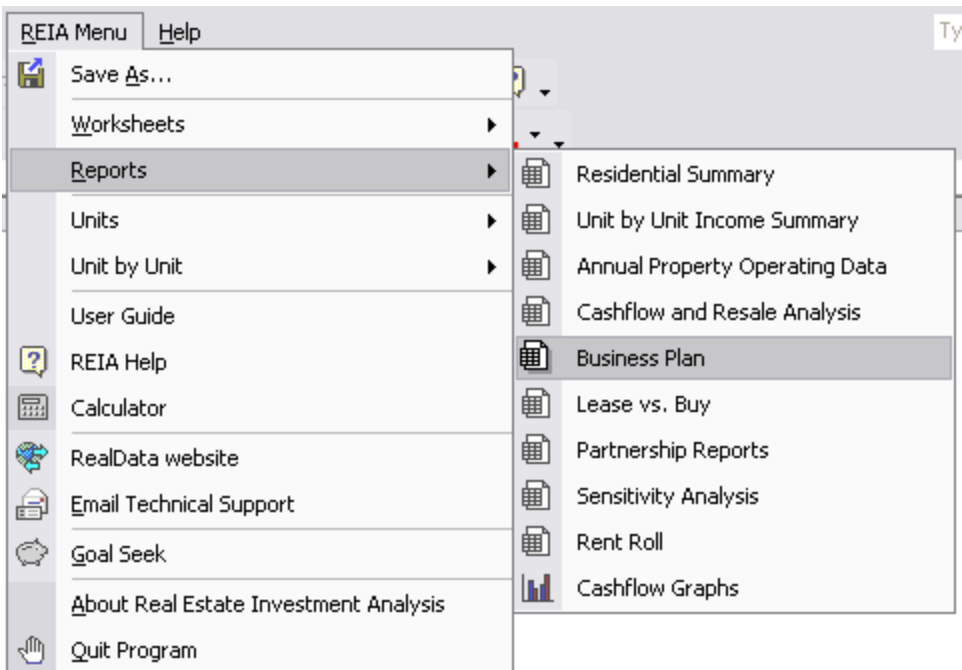
At the simplest level, we take the Excel workbook structure, dividing it into both visible and invisible worksheets. The visible sheets, to the extent possible, follow a logical, linear workflow. We start with general property information, then sources of residential rental income, then commercial rental income, then operating expenses, then financing/cash flows/resale, then partnership considerations and so on.

The invisible sheets, like the engine room of a cruise ship, are where much of the grunt work gets done. There are thousand and thousands of unseen calculations that test various scenarios and bring the appropriate results back to the visible areas (what if the buyer is a 28% taxpayer, buys in 2004 and sells in 2009? What capital gain rate should apply?) In fact, if we printed all the formulas in *REIA*, we would have a document well over 1,000 pages long.

The hidden sheets are also used to reformat information for printing, so that you don't have to just "print the spreadsheet." This allows you to create presentation-quality reports with just a mouse-click and without having to figure out how to do so in Excel. It also allows you in most cases to print just data that you've chosen – for example, the first 7 years of a 20-year projection.

2. Custom Menus and Toolbars

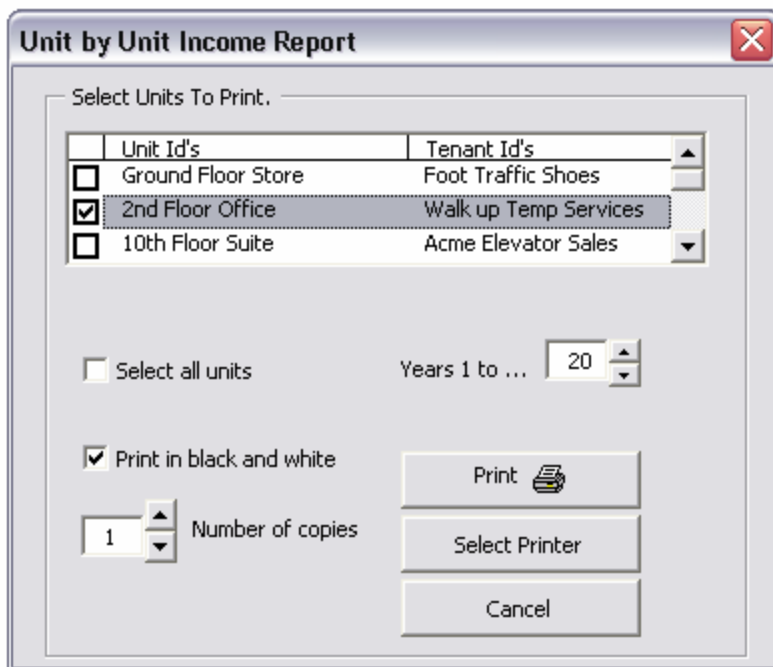
One of the many ways we transform an Excel module into an application that you can learn and use easily is to add custom menus and toolbars, such as you see here:



(Page 4 of 5)

3. Customized User Forms

As we discussed earlier, report generation is a good example of the type of automation that should differentiate a real program from what you may have thought of as a template. Selecting any of the reports from our *REIA* menu will take you to yet another advanced feature, a customized “user form.” In our Unit-by-Unit report you can select the specific data records you want to print, the number of years of information to include and other options. You can even choose to create an Adobe Acrobat file so that you can email the report to a client.



Unit Id's	Tenant Id's
<input type="checkbox"/> Ground Floor Store	Foot Traffic Shoes
<input checked="" type="checkbox"/> 2nd Floor Office	Walk up Temp Services
<input type="checkbox"/> 10th Floor Suite	Acme Elevator Sales

Select all units Years 1 to ...

Print in black and white

Number of copies

If you examine further the custom menu shown above, you’ll find clues to other features that tell you this is not your father’s spreadsheet template. You can navigate among sheets, open the User Guide, access a searchable Help file, pop up the Windows Calculator and contact RealData tech support.

4. Error Trapping and Data Validation

We all know that spreadsheet formulas operate on values that you as the user enter into cells and also on values calculated by other formulas – and so on down the line. We also know that each of us from time to time has a brain cramp. Combine the complexity of a model having tens of thousands of intertwined formulas with our human propensity for an occasional loss of focus and the opportunities for error certainly exist.

In Excel models that are not professionally developed you will seldom find much error trapping. With a virtually unlimited number of possible combinations of data entries, there is no way anyone could hope to trap every user error, but we can certainly intercept a large number of them.

For example, you might tell the program that the analysis begins in July of year 1 but then try to specify a tenant rent increase or place a second mortgage in June of that same year. If this were “just a template” you could make any entry you want and would have to monitor your actions carefully to insure that you didn’t enter information that was contradictory. In *REIA*, you can’t make these

Why Should I Pay You for This Software? It's Just an Excel Template, Right? I Can Do That Myself!

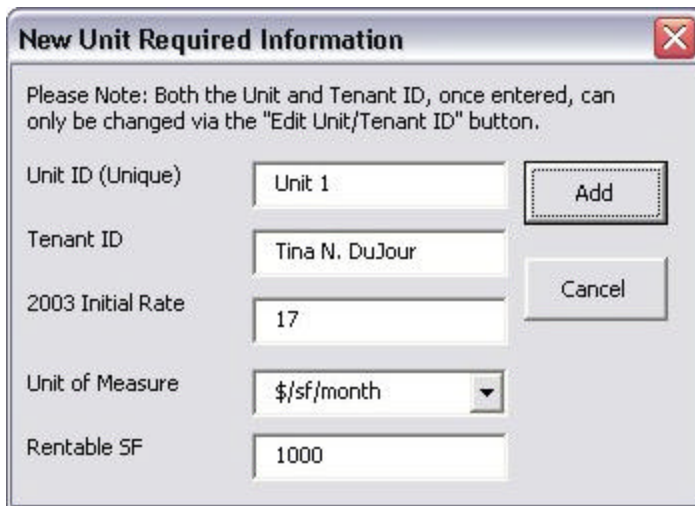
By Frank Gallinelli

(Page 5 of 5)

errors. The program checks items like these for internal consistency and prevents you from making such errors. It's far better for a good program to catch errors before you make them than for you to spend time puzzling over why the results appear incorrect.

5. Database Functions

Some of *REIA*'s most advanced features reside in its commercial Unit-by-Unit Income Summary. This section of the program functions more like a database than a spreadsheet. For example, you create and delete records through user forms like this one:



New Unit Required Information

Please Note: Both the Unit and Tenant ID, once entered, can only be changed via the "Edit Unit/Tenant ID" button.

Unit ID (Unique)	<input type="text" value="Unit 1"/>	<input type="button" value="Add"/>
Tenant ID	<input type="text" value="Tina N. DuJour"/>	<input type="button" value="Cancel"/>
2003 Initial Rate	<input type="text" value="17"/>	
Unit of Measure	<input type="text" value="\$ /sf /month"/>	
Rentable SF	<input type="text" value="1000"/>	

The advantage, of course, is that you're prompted to provide the necessary information in an easy-to-follow format. Information is actually stored in a hidden file so that it can be reformatted one way to display on screen in a familiar spreadsheet format and yet another for printing of reports.

6. Visual Basic for Applications

Unless you're a mechanic, you probably don't know how the fuel-injection system in your car works and don't really care so long as the car runs. The same can probably be said for the Visual Basic for Applications (VBA) code that makes many of the special features in *REIA* run. Usually, the easier something looks, the harder it is to accomplish, and that is certainly true of the programming code that underlies some of these features. In particular, months of development time were devoted to creating the database and reporting features.

Nearly all of the VBA modules in our investment analysis – code to add and remove toolbars, generate reports, create and maintain databases, operate goal seeking and more – are not something an end-user could create with Excel's macro recorder. They require programmers with professional skill. As the user, you benefit because they make your work easier.

The Bottom Line

It's time at last to get back to our original and obviously rhetorical questions: "Why should I pay you for this software? It's just an Excel template, right? I can do that myself!" There are Excel templates and then there are professional, Excel-based programs. If you truly are an active real estate investor then you understand well enough the concept of highest-and-best use. Your time is too valuable to spend trying to reinvent the proverbial wheel. The highest-and-best use of your time is in making deals, so you should select the top investment analysis tools – tools that have more than 20 years of real-world use and refinement behind them – and then put them to work for you.