Strategies & Considerations for Custom Reporting in Yardi Voyager™

By: David Wolfe, CPA
President - Lupine Partners
Is a custom report really necessary?

This is an important question. A more important question is this one: Do you need one all-inclusive report, or will a combination of existing (free) reports do the trick? You need to do a cost/benefit analysis determining whether the internal and external costs of developing a custom report are worth it. You only have a finite amount of time and money. Is the benefit of having a custom report worth the internal/external costs? I have written reports that have cost the client over $5,000 (due to their complexity). They could have the same information at no cost just by using 2 canned reports. The goal should be to have the information necessary to make decisions - period.

“You only have a finite amount of time and money”

Should the expertise of creating custom reports be developed in-house?

Without a doubt, this should be a goal for your organization. You’ve made a large investment in Yardi, and you need to be able to control your reporting destiny. You must be able to get the data out of the system in the format you want in the time frame you want. You don’t always want to rely on outside consultants and the support desk at Yardi. If you do decide to get that expertise in house, be sure you get two people who have this knowledge, not just one. Otherwise, you can get yourself into a ‘hostage’ situation with that particular employee.
Do you have a list of questions that you ask a client when they come to you to build a custom report?

Absolutely – here are a handful of them:

- What is your desired output?
- What items do you want on the report filter?
- Do you have a mock-up of what you want to see?
- What are the selection criteria?
- What is the report trying to accomplish?
- Do you have a company standard with regard to the heading and other notations?
- What is the desired completion date?

After creating their specifications, I will ask them if the specs are a good translation of what they are asking for, in order to confirm that I understood their requirements.

As you go through an implementation, how do you decide which reports should be custom reports?

One of the things we ask clients to do during the module design and configuration meetings is bring all of the reports they are currently using to the meeting. These reports could be from their existing system or ‘offline’ reports, such as side Excel sheets. All these reports are taken and broken into a spreadsheet with three columns:

- There is a match with an existing Yardi canned report (no cost)
- There is a match with an existing Yardi canned report that needs some modifications (cost)
- Or there is no match and there needs to be a custom report (cost)

From there, a time estimate is normally given for each report. The estimate is given to the client, and they decide whether a custom report will be developed. After seeing the amount of effort required to create a particular report, it is not unusual for a client to decide to use one or more canned reports to fulfill the reporting requirements.
What are report specifications, and why are they necessary?

Report ‘specs’ give the report developer the Who, What, When, and Where of the report. It basically draws a box around what the user is requesting for a report.

The specifications outline the purpose of the report -- what you are trying to get done, and any sort of special logic that the report developer needs to take into account as he’s developing the report. Without the report specifications, the developer is just guessing.

Sometimes, even a minor change can be a complete redo ($$$!!) of the report, because of the approach taken. I’ve had instances where I would have developed a report differently had I had all of the desired/required information from the start.

Some items on the spec document include the:

- layout of the report
- report rows
- report columns
- output device or medium the report needs to go to, such as Word or Excel or Crystal
- report filter - when you run the report, typically there are some data elements (like property, tenant, or fiscal period) that can be chosen at runtime that have a major influence on how the report is developed and the actual report itself.
Do you have to know SQL to write reports in Yardi Voyager™?

You do, to an intermediate level. The better question is: How hard is it to learn SQL? The answer to that question is: Not that hard. The basic elements can be grasped in a day. The most important thing to know is what the Yardi Voyager™ database schema is. All databases have tables, and within each table there are columns. If you know SQL but you don’t know where the data is, then you are not going to be able to develop reports in Yardi. Both the knowledge of SQL and the Yardi schema is attainable in a fairly short period of time. It does help to go to a training class, which will shorten your time frame with both efforts.

How do you approach creating a custom report from scratch? What is your process?

The most important part is making sure the correct data is pulled into the report. I have seen consultants who will actually start at the end and make the report look pretty. That work normally ends up being a throw-away, and it is an immature approach to report development. We start in Microsoft Query Analyzer, and create the SQL select statement necessary to pull the data into the report. The data is not going into the Yardi SQL script yet; this is just some meat-and-potatoes hard work to pull the data.

Once that is done, I place the ‘good’ SQL statement within the confines of a Yardi script, conforming to all of the Yardi script rules and requirements. From there, I will integrate a dynamic filter in with the script. (A filter is the mechanism at runtime to decide how you want to limit the report. It’s the criteria of the report.) Once I see that the correct data is being pulled and that the report complies with Yardi’s requirements for scripting, then, and only then, do I make the report look pretty. Do the cosmetics of the report last. Do the guts of the report first. You’ll save a lot of time.
I've heard that sometimes it is easier to have multiple versions of a report with only a slight change between them than having one larger report that takes into account all of the functionality of the individual reports. How can this be?

Let's say you have a report that is working, we'll call it Report A, and somebody requests that an item be added to the report filter. If this item is to either run the report in edit mode, which only produces a report, or in update mode, which goes into the system and actually updates, (for example, the Trans table) then it takes some skill (and time) to actually build this dual functionality. The fastest way would be to keep the original Report A, copy Report A and call it Report B, and make it so the only purpose of Report B is to do the update. Then, there is no dynamic filter item. You run one report to edit, and one to update.

Creating dynamic filters is doable, but now the report has to be written dynamically to do one of two things at the time you run the report, and there can be a fair amount of complexity with that. It is actually shorter to create those two reports individually (by copying the first report). If time is of the essence, this is how I always handle situations like that. We might go back at a later time to make that report be one report, but that is up to the client. It may seem counter-intuitive, but I have done this many times for clients, because I can give them the functionality but keep consulting fees down. Having two reports is usually easier (and cheaper) than creating one more intelligent report.

“Having two reports is usually easier than creating one more-intelligent report.”
Which is better YSL (Yardi Spreadsheet Link) or SQL scripting?

Neither is better. YSL is a little more user intuitive because it uses Excel as a foundation, which most people are familiar working with. It works best with financial statements, although it can be used in a limited fashion for non-financial reports. SQL scripting is generally more powerful. You have to be trained in SQL, and then learn how to use that knowledge within Yardi’s SQL scripting rules. Both are successful approaches.

Is it better to develop your reports in ad Hoc or Crystal?

Actually neither. All Yardi reports start with a SQL script, and these scripts are stored as a .txt file. The script pulls the data from the Yardi tables and sends it to a desired output such as Word, Excel, Crystal or Adobe PDF. You can do further manipulation once you get them into those various products. Even the ad Hoc report writer creates a Yardi script. There is a little button you can click which will show you the script. This is how I learned the scripting syntax when I started writing reports in Yardi in 2000 – by toggling back and forth between the Yardi scripting user guide and the script that is created in the ad Hoc report writer after you ‘click the little SQL button’. 
If you add a custom table or field to Yardi Voyager™, can you report off of it?

Yes, there just needs to be a correlation between that custom table and the data being pulled. For example: If you created a custom table ‘below’ or attached the unit table to track the size and types of windows in a unit, you would not be able to pull that information into a financial statement, because there is no connection or correlation between the two bits of information – financial data and this custom table around windows for a unit. Generally speaking, yes - if you have added a new or custom table, then the columns in that table are a candidate for being used in a custom report. Also, you will need to know the table name and the column name as you must use that table.column in the actual SQL script.

I want to be able to 'slice and dice' data on my reports. How can this best be accomplished?

The database needs to be normalized in order to ‘slice and dice’. Normalization refers to the concept of breaking data into as many discrete bits of information as possible. For example: there may be 5 different types of discreet data (attributes) related to the property you would want to ‘slice and dice’ on. Typical examples include:

- property type
- region
- owner
- state
- portfolio

If those particular attributes are set up and populated for that property, then you can ‘slice and dice.’ In other words, it’s not so much a reporting mechanism as a database strategy mechanism. Once that is done, you create the report. On the report filter, you would include those five attributes, and then you can begin your slicing and dicing – running reports by property type, region, etc.
Is it better to develop the reports on the test database rather than the live database?

It doesn’t really matter, because you are just selecting and pulling data. Even if you pull the wrong data on the report, it’s no harm, no foul. It just means the report will have bad data – but the core system is not harmed. The only time to be concerned about developing a report in the live database would be if you had an Insert or Update statement in the Yardi SQL script. This scenario is a possibility, depending upon what you are looking to do with the report. It’s rare, and it’s a more advance report. In that instance, I would create that report in a test database.

Is it easier to develop a custom report using a local version of the database, as opposed to connecting over the internet and using Yardi’s environment?

It is easier to develop locally for a number of reasons. You will have access to tools that you might not have going through Yardi’s environment, such as the entire MS SQL Server Management Studio and SQL Profiler. Also, you don’t have to deal with the internet and slow connectivity speed. The reports usually run a little faster, cutting down on development time. The downside with developing locally is that the report might run on your system, but not the Yardi production system for reasons unknown. Then you have to debug on the Yardi system to sort out why it ran on your system and not on theirs. Know that if you develop on Yardi’s environment, you will have to download and install a backup copy of the database. This takes a little bit of time (you can do it in the background), but it could take a few hours depending on the size of the file.
**Without a safe point, you run the risk of undoing portions of the report that previously worked.**

What is the term 'safe point' and what does it mean with regard to report development?

When you are developing a report there are times where you get stuck on, or are struggling with, a particular portion of the report. Without a safe point, you run the risk of undoing portions of the report that previously worked. I’ve been down this road many times, and years ago developed the safe point strategy, because it can be difficult to reconstruct logic that you’ve already created once.

As you work through a report, you will have mini-successes – those points where things finally work. When you know that section is good, instantly save the report. Save the report as a ‘stage’ – stage 1, then stage 2 at the next mini success, etc. This gives you a recovery point if things go poorly while trying to proceed to the next stage. Essentially, this is a development practice of saving at certain milestone events as you are working through a solution. It is a bad feeling to not be able to reconstruct logic that you successfully created before. It happens...
If a custom report causes an error message when run from the system, what is your process for debugging the error?

There are several different steps to take when encountering an error message:

- As a first step, I normally take the SQL code into Microsoft Query Analyzer, and if you don’t get an error, then you know it’s not the SQL select statement, and it must be something else.
- Next, you might run the report in debug mode in Yardi to see if anything pops out as being ‘weird.’
- Third, check for any stray commas or apostrophes. Commas and apostrophes can cause problems within SQL, because of some special meanings they have within that programming syntax, especially the single apostrophe.
- Next, run for just one property, and determine if it is a property mix issue.
- If it’s a financial statement, I might run the report for just one GL account. I would set up an account tree with only one account. Error? Yes or no.
- After that, begin to take out portions of the select statement and putting them back in one by one until you can identify the break.

Bottom line with debugging reports is always, always change just one thing at a time so you can identify the problem when it occurs. If you are changing 2 or 3 things at a time and the report begins to run properly, you don’t know which element is the one that was causing the problem. It’s a very methodical process. I normally turn off the phone and email because I don’t want that mental string broken when in debug mode. The last line of defense is to ask somebody else to look at the code. More times than not, they will see the error immediately. (You’ve been too close to it, and the answer is probably staring you right in the face...) I have been on both sides of this -- where I found someone else’s error, and also when someone finds the error for me.
Is it better to output reports to Crystal or Excel?

There are advantages to both. If you output to Excel, you can further manipulate the data and create macros to automate some tasks. You also get the use of Excel’s formulas and formatting, which most people are familiar with. With Crystal, you can’t further manipulate the data. This inability to manipulate can be an advantage, since a fair number of our clients don’t want the end users to have the ability to change the financial statements. In Crystal, you also get the use of subreports, which gives you the ability to do some pretty interesting programming. And, there are formulas conditional of formatting in Crystal just like Excel. You should know that if you have outputted the report to Crystal, it does not format well to Excel from there. You are better off going directly to Excel if that is your goal or end medium.

How do you decide whether to output a report to Excel, or Crystal, or Adobe or Word...or just to the screen?

Usually, you don’t. The end user usually tells you what they want. Your job is to determine if the output they want is possible and advisable. I can promise that, if you guess what the output should be, you will almost always guess wrong. Get the user of the report to tell you.
99% of the time, a custom report will continue to work as Yardi adds service packs and plug-ins

If Yardi Voyager™ is upgraded, will the custom report still run?

Almost always, a custom report will work just fine after an upgrade. There are a few instances where it might not. One instance is if there is a total re-write of a module, such as Yardi’s upgrade from 5.0 to 6.0 for commercial. This was a complete schema change. Custom reports that worked for 5.0 did not work for 6.0. They wouldn’t necessarily error-out, but they certainly did not return the correct information. Another instance to consider is when there are some minor functionality changes - like how traffic used to be handled in residential for previous versions of Voyager™. There were some new tables set up around prospects, so box-score reports were no longer valid.

In general though, you are going to be fine. 99% of the time, a custom report will continue to work as Yardi adds service packs and plug-ins, and you will know in advance when major upgrades are coming along that will change the database schema.

Is there one large library somewhere of all the reports that have been developed in Yardi?

No – not even within Yardi. You will have reports loaded onto your default path, but know that these do not represent all of the reports that exist in the Yardi universe. There are tens of thousands of reports that have been created – by Yardi personnel, by independent consultants, and by Yardi users. Joining a regional user group is a good way to get your hands on some other reports. User-group members have been known to share reports between each other.
About the Author

David Wolfe, CPA is widely regarded as a leading authority on software selection and evaluation. He has been the lead consultant on numerous software selections and implementations since he founded his software consulting firm, Lupine Partners (http://www.lupinepartners.com), in 1993. His rational and systematic approach to software selection and implementation has won him loyal clients across the United States. When he is not traveling and assisting companies with their software concerns, David lives in Dallas, TX with his wife Susan.