

# Managing Project-Level Planning and Reporting with OneStream

[This article first appeared on the OneStream blog by Robin Hankey](#)

Many planning applications in the market are built based on a multidimensional cube approach that provides high performance for slicing and dicing summary level information. However, these solutions typically are not well-suited to handling large volumes of detailed data for planning and reporting. OneStream is the only corporate performance management (CPM) platform to offer a tightly integrated relational and cube concept in a single solution. This blog article explores my use of this combination of relational and cube data at one of my customers to address their unique requirements related to project-level reporting and planning.

## Project-Level Reporting and Planning

In addition to the more conventional financial consolidation and planning needs, project reporting is key to one of our customers' business. As part of this, they require a sub-ledger to manage financial details of their various ongoing and anticipated projects. Prior to implementing OneStream, they managed these projects in Excel. They have thousands of ongoing projects at any given time, and many more being booked each month.



For each project, they need to capture things like contract size, customer, location, product classification, the revenue and the costs by several major buckets (e.g., labor, material, etc.), how much is left of the project, and the gross margin. On the planning (forecast/budget) side, they need to plot out the revenues and bookings into future months. This customer also needs to run various reports at a project level, including being able to see which projects or customers have the greatest impacts, or to identify concerning projects. The aggregated project details from this sub-ledger also needs to be fed into the consolidation cube just like any other sub-ledger to augment their more typical cube-based Actuals and Planning reporting needs.

## Cubes vs Tables

Typical financial cubes do not work very well with capturing data down to a transaction level, be it individual Employee, Project, fixed asset, GL journal, for which you could have thousands of items (be it employees, projects, leases, transaction numbers, GL journal numbers, etc.), with many such items coming and going over time. This sort of transactional data is better suited to more conventional relational database/table storage. And in fact, that is exactly how General Ledgers, Payroll/HR/Fixed Asset registers, etc. are built. When this sort of data needs to be fed into a cube for more convenient slicing/dicing type reporting, the transactional (item level) information is typically aggregated out of the data, such that the cube does not know or have any of the item level detail.

Traditional consolidation and planning systems typically only offer cubes and cube centric

slicing/dicing type reporting. Whereas transactional systems have data down to the item or transaction level but are not so good at slicing/dicing to find exceptions.

## Introducing Relational Blend

Along comes OneStream. OneStream's [Intelligent Finance platform](#) offers a rich set of cube centric slicing/dicing capabilities. But as a relatively unique feature, OneStream also offers an ability to enter, maintain, compute and report on relational/transactional stored data too. This OneStream feature means it is possible to build things like subledgers directly in OneStream relational tables which sit alongside the OneStream cubes, and all accessible through a single application/user interface. These subledgers could be for things like Employee Planning, Lease or Fixed Asset Management, or in the case of my client, project-level reporting, and planning, among just a few examples.



While it is possible to build a purpose built/custom relational/transaction stored module entirely from scratch that resides entirely within OneStream, it would usually be cost prohibitive to start from scratch, and it is not usually required. Somewhat similar to Android/iOS products, OneStream has an app store-like concept (the [OneStream MarketPlace™](#)) whereby one can very easily add optional solutions to the core cube-based platform to serve as a starting point for these sorts of relational/transactional storage needs.

## OneStream MarketPlace™

In particular, OneStream's MarketPlace includes a family of "relational blend" solutions, including People Planning, Sales Planning, and a more generic Thing Planning.

All the relational blend solutions include a "Register" in which users enter/maintain the details of the items (eg people, leases, projects, etc.) that they wish to manage, and a "Plan" where various OneStream calculated values are stored based on what is in the register, and potentially other sources of information.

These solutions include a calculation engine to define what calculations are to be performed from the Register to the Plan. For something like People Planning, the users would enter employees, department, location, type, status, base salary into the Register, and the Plan would hold the OneStream computed monthly wages benefits, taxes, etc. by employee. Once the data has been entered into the Register, and the Plan computed, this sub-ledger like data can then be optionally loaded into OneStream (or even sent to some other system) just like General Ledger data is loaded into OneStream. I will describe reporting options later in this blog post.



And OneStream also provides drill back capabilities from the cube back to relational data, so if a user were wondering why a department's payroll expenses are so high, they could drill back to the People

Planning Register and/or Plan and/or custom report to see that it was a result of giving a specific employee a big raise. To the user, moving from cube data to relational data is seamless – they don't even know (or care) where the data resides. As OneStream says – this permits the data to live where it belongs, not all data belongs in the cube.

## Thing Planning

For our client's project reporting and planning needs, we used OneStream's Thing Planning solution, which was the closest-fitting for their project-based needs. In fact, we used two instances of Thing Planning (Thing1 & Thing2), as their project reporting and planning needs varied a fair bit between Actuals and Planning.

- We use Thing1 for Actuals and Thing2 for Forecast/Budgeting. For Actuals/Thing1, we need to capture more project details into the Register of the ongoing or just started projects, and also compute many more amounts into the Plan (See Figure 1).
- For Planning/Thing2, we take the current projects from Actuals/Thing1, and users then project out what is going to happen for future months for existing projects and add in new bookings. We compute about 20-30 amounts for the current month into the Plan for Thing1, but there is no need to compute anything for Thing2. These calculations can range from very simple to as complicated as one could desire.
- For Thing2, the seeding process captures details of the ongoing projects from Thing1 and marries it to the projections from a user selected previous Thing2 plan. We built several validation rules to prevent users from loading the project data to the cube until validation rules pass, thus ensuring and maintaining data integrity. We built a number of relational reports in Thing1 & Thing2 so users can analyze their Projects before or after loading the summarized project data to the cube. Many of these reports can also show monthly, QTD, YTD, LTD amounts in local or reporting currencies.



Figure 1: Thing1 Actuals Register in OneStream



Figure 2: Thing1 Actuals Calculations in OneStream

Though these relational blend solutions include a web interface for editing the register directly, my client had too many projects to make the web interface viable, so opted to export the register to Excel where they perform their register edits and import the sheet back to the register when done, leveraging out of box export/import capabilities. I included a method to seed the Register at the start of each month, or Planning cycle, so users are not starting with a blank sheet, which greatly reduces their manual effort and chances for human errors.

My client opted not to use the drill back from cube capabilities, as it generally takes fewer key clicks to simply switch over and run a Thing1/Thing2 report to get the specific detail they are looking for – one of the benefits of having OneStream cubes and Thing Planning in the same application and

sharing the same user interface.

## Reporting & Table Views

For reporting on relational data, OneStream includes several reporting tools out of the box. Web based Dashboards (Studio, BI Viewer) are the main relational reporting tools. OneStream's Cube Views, which is the main reporting tool for surfacing cube data can, with some considerations, be adapted to surface relational data too. As an extreme example, I have been able to surface entire relational reports within Cube Views, though there are some performance and formatting considerations. I mention the cube views more as a means to demonstrate how flexible OneStream is, even if not the right tool for the task.



And with OneStream V6, they now offer Table Views in Excel, which can be used to surface just about any relational data or OneStream cube data that can be represented into a tabular sort of layout. Table Views can optionally support write back. Table Views is potentially a big plus for users who have too many things to enter into the register, and thus aren't keen to use the out of box web interface but might not be too fond of exporting/importing the register to Excel. With Table Views, it could be possible to build an Excel cohesive workbook that includes the relational blend Register editor and provides a number of canned cube or relational reports.

## Summary

In summary, the Relational Blend feature of OneStream is a great extension of their solution and dramatically extends the capabilities of the platform. In the example above we demonstrated how Relational Blend could be applied to project-level reporting and planning. This combination of cube and relational blending is unique in the market and the use cases are countless.

I, for one, look forward to finding novel solutions to expand the use of OneStream into new frontiers in the CPM world. To learn more about the specialized planning solutions that leverage the Relational Blend feature of OneStream, visit the [OneStream MarketPlace](#).

management, or, in the case of my client, project-level reporting and planning, are just a few examples of what these subledgers may be used for