REVELWOOD

How We Solve Problems – IBM Planning Analytics Use Cases

Gold Business Partner

Competency
Data Science & Business Analytics

About Revelwood

For nearly 30 years, Revelwood has worked with clients of all sizes, in all industries, to maximize their use of IBM Planning Analytics. Not only do we design and implement FP&A systems built on Planning Analytics, but we also work with clients to enhance, update and optimize their existing Planning Analytics environments.

Revelwood's IBM Champion, Lee Lazarow

We are the IBM Planning Analytics / TM1 experts. Our Planning Analytics Practice Leader, Lee Lazarow, has been implementing Planning Analytics and TM1 solutions since 2003 and has been recognized worldwide as an IBM Champion multiple times, including in 2024.

> "In 20+ years, I have yet to find an FP&A challenge that can't be solved with Planning Analytics."
>
> Lee Lazarow
> Planning Analytics Practice Leader

Contact Information

Revelwood

World Headquarters
Florham Park, NJ | 201 984 3030

European Headquarters
London & Edinburgh | +44 (0)131 240 3866

Latin America Office
Miami, FL | 201 987 4198

Email
info@revelwood.com

www.revelwood.com

Problem: A Cube Using a Massive Amount of RAM

Client Profile: A publicly traded real estate investment trust (REIT) with more than 3,000 properties in nearly all the U.S. states.

Scenario: Revelwood was brought in to help with an upgrade of their Planning Analytics environment and to help convert their existing reports to the new version. During this process, we discovered that a Planning Analytics allocation cube was using a massive amount of RAM, resulting in abnormally slow processing and performance.

We dug deeper to better understand why that cube was so large. The company has approximately 1,000 tenants, with some expenses for each property allocated to the associated tenant. The model was originally built to process allocations to every tenant – even in situations where the property only had one tenant. As a result of this, the cube was almost 1,000 times larger than the pre-allocation expense cube. The pre-allocation cube used 40MB of memory and the post-allocation cube was using more than 30GB of memory.

How We Helped: Working with the client, we discovered that almost all of the properties were associated with only one tenant. We created a TurboIntegrator script to determine the applicable property-tenant combinations by analyzing data in one of the allocation definition cubes. If data existed, an assumption was made that the combination existed. We then updated the rules to ensure that the calculations were only processed for the applicable combinations. The end result was that we trimmed down a cube that was using more than 30GB to one requiring only 65MB of memory. This was an improvement of more than 99%.

Problem: Needing to Add Staff Planning

Client Profile: A business support systems (BSS) company that provides software and services to help their customers with billing, customer service and operations activities.

Scenario: The company did not have a staff planning model and was looking for a way to incorporate this into their existing IBM Planning Analytics environment. The planning process had many complexities including:

- Contractor planning (hourly) and employee planning (both hourly and salary)
- Multi-year planning to calculate raises and bonuses
- Currency/FX calculations associated with both Act exchange rates and Bud exchange rates
- Both headcount and FTE analyses
- Ability to plan both by person and at a summary level for the cost center

How We Helped: Revelwood created a model that would function for both US and international cost centers. The model was introduced into the existing environment and also linked to the core financial model. This allowed the customer to use templates and reports that already existed while adding new functionality to their planning process.

Problem: Repetitive Scripts

Client: A financial services company specializing in retirement and life insurance.

Scenario: A review of their existing TurboIntegrator scripts found that many of their processes used pre-built source views. Each view differed only slightly. For example, one view would define subsets in approximately eight dimensions and focus on a single time period while a second view would contain the same subsets and focus on a different time period. This approach led to more than 50 repetitive scripts that were run via chores.

How We Helped: We changed the approach to create a source view in the prolog. This allowed a single script to be run over and over via parameters that were defined during each iteration. This resulted in faster maintenance since changes only had to be coded once. It also ensured that changes to public views and subsets would not impact these scripts.

Problem: Converting a Manual Process to TurboIntegrator Scripts

Client: A financial services company specializing in retirement and life insurance.

Scenario: This client's business process included allocating summary quarterly forecast data to a more detailed level. When
looking at months and other dimensions, there were over 200 trillion possible target combinations. The process was performed manually and took two days to consolidate the source data, perform the allocation calculations in Excel and load the results into Planning Analytics.

How We Helped: We converted the full process into automated TurboIntegrator scripts that took under 10 minutes to run.

Problem: Slow-Performing Scripts

Client: An accounts receivable provider that offers collection and contact center services.

Scenario: This client had a very large model that led to slow-performing scripts and reports. We analyzed their model and determined that much of the data was associated with historical information that was no longer needed. This included multiple versions of plans dating back to 2015.

How We Helped: We worked with our client to determine an asymmetrical approach to removing data (example: keep three years of budget data, one year of forecast data and remove all historical reforecast data) and reordering the dimensions in the model. These changes reduced the four largest cubes from 56% to 70% and removed more than 8.5GB of memory. This optimization led to faster reports calculated in seconds instead of minutes.

Problem: Reports Taking 10 Minutes to Refresh

Client: A publicly traded television broadcasting company that owns and operates more than 175 stations across the U.S.

Scenario: The company's budgeting process consists of a very granular process that plans down to the individual employee. The calculations in the model include detailed benefits and taxes, combined with multiple bonus calculations that depend on results from the core financial model. Due to the size and complexity of the process, "top of the house" reports took more than 10 minutes to properly refresh.

How We Helped: Revelwood analyzed the existing calculations and the interaction of the various models to determine an efficient order of processing. Due to the need for an order of operations, many of the live rules were converted into on-demand TurboIntegrator scripts. This approach shrunk an individual company's refresh time from one minute down to a few seconds, while also bringing the "top of the house" analysis down to under 20 seconds.

Problem: Convert to Driver-Based Planning

Client: A private university with campuses throughout New York and New Jersey.

Scenario: The university was planning revenue using an Excel-based spreadsheet. The spreadsheet contained many repetitive sheets, had very complex formulas and had some erroneous calculations built into the file.

How We Helped: Revelwood reviewed their existing model and helped convert it into a driver-based planning approach that used a combination of headcounts with various lookup tables. The model accommodated various review sources such as tuition and other fees, while also calculating contra accounts such as tuition and bad debt. Users planned by semester. We converted the results into calendar months, taking into consideration both calendar years and school years. This simplified approach allowed the client to easily load and update data into the model and then make modifications for what-if analyses.