

How to streamline the foundation of your ETF workflow

An agile Microsoft Power Platform-driven solution for an ETF service provider (and anyone else keen to source, transform and deliver data in an instant)

The problem

My client needed to calculate Portfolio Composition Files (PCFs) for several clients who wanted to launch Exchange Traded Funds (ETFs) for their multi-class funds.

Until now, PCFs were calculated at a fund level where all funds only had a single class.

Many, if not most, of the serviced ETFs are ANTs (Active Non-Transparent ETFs), where solution flexibility and PCF accuracy are front and centre.

Although they'd provided ETF services for almost a decade, the Fund Administrator was unsure how to best to do this differently.

The existing solution

PCFs were calculated through VBA in a macro-enabled Excel workbook. The tool sourced the underpinning data from the investment accounting system, HiPortfolio, via SQL queries through ODBC and combined them with data from other raw data files.



Despite being modified several times in response to client demands, the fundamentals of this decade-old solution remained robust.

That said, Consider-it's discovery process highlighted the following issues that the client chose to address:

Time and Scale

The US market closes at 8 am in much of the APAC region for six months of the year, with end-of-day prices available at 8.20 am, where Portfolio Composition Files (PCFs) need to be delivered to iNAV calculation agents by 9 am, leaving just 40 minutes to finalise a unit price for global equity funds and get a PCF out the door.

My client's solution only handled PCF calculation for one fund at a time. Consequently, it needed to be triggered as many times as there are funds.

The result? A high-pressure, headache-inducing bottleneck for teams tasked with manually generating PCFs for growing numbers of Exchange Traded Funds (ETFs), with each run taking 60 to 90 seconds – for thirty funds, that's between 30 and 45 minutes for calculating the PCFs.

The time constraints imposed by this problem put pressure on a dozen analysts striving to provide accurate PCFs in the early morning rush.

Key Person Risk

Historically, my client had tweaked the logic and layered complexity into the VBA-based solution. However, when they contacted Consider-it, there was limited internal VBA programming expertise coupled with the specific business knowledge needed to interpret and address the new requirements.

Transparency and Accessibility

Anyone unfamiliar with VBA can struggle to track down where a macro (workbook, worksheet, module) lives. Furthermore, stepping through a macro with a lot of code and numerous loops can be frustrating for the non-expert.

In addition, this transparency was limited by the very specific SQL queries sent to HiPortfolio, where the data was filtered and transformed at source, obscuring the breakdown of the data for the end user.

In summary, my client's end users' distance from the mechanics of the solution made them and it vulnerable if glitches arose. The team hoped to get the expected result. If they didn't, and there were issues with the output, troubleshooting was next to impossible in such a short timeframe and without solid in-house expertise and/or a user-friendly solution.

What we did

Having pinned down the major pain points and finalised the required logic for calculating PCFs for multi-class funds, we needed to decide what tools to use to create a new solution from amongst three viable options VBA, SSRS (SQL Server Reporting Services) and Microsoft's Power Platform.

We decided on The Power Platform (specifically, Power BI with Power Automate), the most forward-looking and flexible of the three.

VBA – why did we rule it out?

Limited transparency: VBA lacks Power BI's user-friendly UI. Every step in transforming data within the Power BI Query Editor is only ever a couple of clicks away. In comparison, VBA is antiquated and confusing.

Security risks: A macro-enabled workbook is always a security concern. Many organisations are disabling macros, firm-wide, to avoid malicious macros where attackers frequently use VBA's excellent programming functionality to compromise the firm.

SSRS – why did we rule it out?

SSRS is widely used in the organisation, so it merited serious consideration. However, we were looking for a solution that focused on the end-user, and in that context, SSRS (like VBA) is antiquated and lacks transparency.

In summary, the drawbacks we would have encountered by using SSRS were that it's:

- A static rather than an interactive environment with no customisable UI or report interface
- Neither low code nor intuitive. Hence there's a steep learning curve. Developing reports typically calls for an understanding of various querying languages such as SQL, Visual Basic, and MDX, so development time can be lengthy
- Not compatible with end-user self-sufficiency. A separate team would be responsible for developing the SSRS reports. We aimed to make end-users self-sufficient in maintaining their solution without needing to bridge a knowledge gap to another team.

Power BI and Power Automate – why did it get the nod?

Power BI and Power Automate are two dynamic products in Microsoft's Microsoft Power Platform. Power Platform is a collection of highly adaptable products that support data-dependent businesses to optimise the value of their data.

As a low-code solution developed with end-users front and centre, Power Platform was an excellent fit for my client's needs. Its tools, including Power BI and Power Automate, simplify accessing, presenting, and analysing data, even for users with limited or no technical expertise.

On-premises over a cloud-based solution

Having chosen the Microsoft Power Platform, we aimed to use the full cloud-based Power BI solution. However, my client's IT infrastructure strictly favoured an on-premises solution over a cloud-based one, so facilitating the storing of data (Power BI datasets) in the cloud wasn't a feasible option for the company. This ruled out the cloud-based Power BI Service.

We were, however, given the green light to use the cloud-based Power Automate tool, where data storage wasn't an issue. While Power BI would handle the sourcing, transforming and presentation of the PCFs, Power Automate handled the straight-through PCF delivery to stakeholders via emails and file drops to sFTP sites and local folders.

Hurdles with HiPortfolio

Requesting data from HiPortfolio through Power Query (ODBC) presented challenges. Power Query's default setting is to send multiple requests simultaneously when refreshing data. However, the HiPortfolio application killed the process and returned errors.

We enabled the setting to disable 'parallel loading of data' in Power BI Desktop and found a workaround within the 'on-premises gateway' application when the option to use the Power BI Service (through limiting the MashupDefaultPoolContainerMaxCount setting) becomes available. However, we found no suitable resolution for the Power BI Report Server – when refreshing a dataset on the Power BI Report Server, it's not possible to disable the concurrent loading of data.

Extra clicks in a cloud-free environment

Using the Power BI Desktop application to refresh our dataset and produce our PCFs calls for a couple of extra mouse clicks in the workflow compared to using a cloud-based Power BI Service. On balance, this is still the best option. The team still benefits from the power of this free application to source, transform and prepare their data.

Furthermore, the company is ideally placed when the time comes to migrate to the cloud where they will avail of The Power Platform's complete range of applications, accessing its full potential. And, although cloud-based, Power Automate interacts seamlessly with the local environment through the 'on-premises gateway' application.

The new solution

Scalability

The new solution calculated the PCFs in parallel for all funds. Then it split them into the ETF-specific files in Power Automate before delivery to stakeholders via email and SFTP. To achieve this, we used the native HiPortfolio Data Retrieval Objects (DRO) to source our data. In contrast to the VBA-based solution, which sourced from custom queries, using the DROs allowed us to pass parameters for multiple funds in the SQL statements, which the custom queries wouldn't allow.

The result

Instead of executing the process over and over to calculate each PCF, it only needs to be triggered one time by one person.

Efficiency and Transparency

For our new solution, our SQL queries to HiPortfolio DROs were very basic, with just the essential clauses and conditions to return the data for our ETF portfolios.

The result

The speed at which the data was returned from HiPortfolio was far superior when using simple statements to the DROs. Testing showed that, when using more complex SQL statements to filter and transform the data at source, the process took a lot longer to execute. For us, the Power Query application trumped HiPortfolio in processing speed when transforming data.

By using these simple statements, we are also retrieving the unadulterated data, which gives the user full visibility across the transformation process in the PCF calculation, step-by-step, via Power Query's user-friendly UI.

Risk Mitigation

As well as the Power BI report used for PCF calculation, we also created a report for verifying the data in the final PCFs against the NAV data contained in HiPortfolio. This is another mitigating step on top of the functionality we added to include all PCF data adjustments before triggering the calculation process.

The result

By adding the functionality to include our adjustments – e.g., converting unlisted stock lines to cash or adjusting prices – ahead of calculation, we created exceptional flexibility and ensured there



is never a need to manually update the final PCF files before delivery, thereby keeping data integrity at the expected level.

The report checking against HiPortfolio's NAV data can be run for all funds at once in under a minute. Hence, we're verifying the accuracy of the PCF calculations and providing that extra level of comfort before final delivery, without creating any extra pressure in the busy morning period.

In summary

We created a solution that met the immediate requirement of a top-tier ETF service provider to respond to client demand and maintain excellence in their offerings.

We also envisaged several potential problems and solved them before they were even realised. The solution's components are not interdependent. As a result, it offers ultimate adaptivity and flexibility for whenever my client's environment changes. If or when they choose to migrate to a cloud-based infrastructure, or if they need to move to a new investment accounting system or engage with a new data vendor for corporate actions, etc., they're set to go, with minimal need for extra development expertise or work.