

AssetBook maximized their development resources by letting XLeratorDB deal with the headaches of the XIRR function.

AssetBook (<http://assetbook.com/>) had a major headache. AssetBook is the developer of Radar which is portfolio management software for registered investment advisers and they were spending too much valuable development time de-bugging and supporting their own version of XIRR, an important calculation in measuring money-weighted rate of return (more on the challenges of coding XIRR below). AssetBook found their 'aspirin' in **XLeratorDB from Westclintech** which produced **rock solid numbers** quickly and was easier to support. Plus Westclintech had two separate XLeratorDB products which offered AssetBook the flexibility of either running XIRR natively on SQL Server or as a .NET DLL. Once AssetBook implemented XLeratorDB's XIRR function they could **focus valuable development time** on growing and supporting their product. This case study provides further details on AssetBook's decision to purchase XLeratorDB.

AssetBook's Radar product is portfolio management software designed for the demanding and competitive world of registered investment advisors. The **Radar platform includes personalized dashboards** to help the advisor's customers visualize their data. The flexibility of the Radar platform demands efficient processing, including on-the-fly calculations of client's internal rate of return. Radar runs exclusively as a SaaS portfolio management solution on servers hosted in the Microsoft Azure cloud.

Challenges of coding XIRR

You can face many cascading challenges when coding XIRR to calculate **internal rate of return** or **annualized yield for a schedule of cash flows** occurring at irregular intervals. There is no closed-form solution for calculating an XIRR, so you must implement some root-finding technique. If you choose bisection, for example, you might find that it is too slow and it can hop over a solution. Generally the Newton-Raphson method is a reliable technique though you need to handle its limitations. The Newton-Raphson solution is directly affected by its starting point and there is really no good way to come up with a rule of thumb for the initial guess. Based on the guess, you might actually iterate away from a solution towards infinity or negative infinity. Or, because of the limitations of floating point math you could oscillate around a solution without ever resolving to a solution within the error tolerances.

It's also possible that based on the starting point the resolution can be quite slow (perhaps thousands of iterations). Further, it's also impossible to make any assumptions about the signs of the cash flows (for example, you cannot require that the first cash flow be negative) or the sign of the slope of the underlying function (XNPV) in the region in which you are searching for a solution.



These are just some of the challenges in coming up with an XIRR function that's not only accurate but capable of processing millions of cash flows in just a few seconds. **Westclintech has done the hard work** and its XLeratorDB product is easy to implement, produces rock-solid results, and can process millions of rows of cash flows in just seconds.

Maximize development resources

AssetBook's development team was spending lots of time debugging and supporting their own version of the XIRR function which ran on SQL Server. The problems were compounded because AssetBook was using Excel's version of XIRR to verify their own results and Excel's XIRR function had known issues they needed to work around. In addition, due to the evolution of the application architecture, AssetBook was looking for a product that had a common calculation engine whether running as a **native CLR on SQL Server** or running as a **DLL on a .NET application server**. XLeratorDB met all their needs: XLeratorDB does the hard work around XIRR and it fit easily into the Radar system because it is offered either as a CLR function on the database or as a .NET DLL.

Rock solid XIRR results

Per Michael Petersheim of the AssetBook development team, XLeratorDB's results for XIRR were "rock solid". Now instead of spending limited development time debugging their own version of XIRR or stepping customers through the calculations, they could concentrate on other development priorities, including enhancing the product to meet their customers' evolving needs.

Easy to implement and integrate

It took AssetBook **less than three weeks** to do its due diligence testing and confirm that XLeratorDB's XIRR results were good. From there, it was only another 3-4 weeks until they had deployed XLeratorDB in production.

AssetBook's development team also appreciated the flexibility of Westclintech **offering two distinct versions of the XLeratorDB** product which uses the same calculation engine.

AssetBook could use it either as a CLR function running on SQL Server or as a .NET DLL running on a Windows application server. Although many times it is more efficient to run the XIRR calculations natively on the database server, AssetBook had some unique architectural requirements including doing lots of asynchronous / parallel processing of the data on the application server. This was necessary to support the inherent flexible nature of its Radar portfolio management system. AssetBook was **happy with the flexibility** offered by the XLeratorDB product versions.



AssetBook also found it very easy to use XLeratorDB as they migrated from a smaller, regional hosting provider to virtual SQL Server machines running in the Microsoft Azure cloud.

Even though XLeratorDB contains many other financial functions, including FV, NPV, and PMT, AssetBook determined it saved them so much time that it was worthwhile to purchase the product only for the XIRR functionality.

Great support

AssetBook had a few questions when they were initially using XLeratorDB and Michael said **the support from Westclintech was “superb”**. Westclintech’s XLeratorDB team has 50+ years of experience building, implementing, and supporting financial services software. Michael said the Westclintech spoke their language and understood their challenges all the while providing quick responses to their questions.

AssetBook wanted to focus its development resources on enhancing the Radar system to meet its customers’ evolving needs and didn’t want to spend time debugging / supporting its own CLR version of the XIRR function. AssetBook was very happy they found the XLeratorDB from Westclintech whose XIRR function was easy to implement (either as a CLR function or as a DLL), produced rock solid results, and was supported by a knowledgeable and responsive support team.