A Transaction-based Portfolio Management System in Dyalog-APL using .NET

Claus Madsen, PhD, Associate Professer DTU,
Founder FinE Analytics, Co-Founder and Partner ISave
10. September 2019

Agenda

- Who am I?
- What is a transaction based Portfolio Management System
- The big picture
- Some design features
- Example
- End Remarks

Who am I?

- My profession is Finance
  - Ph.D in Financial Engineering
- Has and am working mainly with Risk (Credit, Market, Counterparty) and Valuation from a Quantitative perspective
- Selfemployed since 2001, mainly developing the financial toolkit: FinE Function Library (www.fineanalytics.com)
  - Before that…Quant (and Risk) for +14 years in different Banks

Who am I?....continued

- I have been working with Dyalog APL since version 6...that means around early 90s

- Which is the only language i really know....on a good day I can read C#.....

What is a transaction based Portfolio Management System?

- Is used by Asset Managers, which we can define as:
- Asset Manager is commonly used in the financial sector to describe people and companies who manage investments on behalf of others
  - Those include, for example, investment managers that manage the assets of a pension fund, Insurance Companies, Private entities etc

The Big Picture

- Build as a series of .NET classes – at this stage 26 classes
- In a number of groups:
  - Assets
  - AT
  - CT
  - Portfolio Object
  - Benchmark Object
  - A range of data containers
  - One Shared Class
    - More on this later

Design features – I

- Data storage and cross class communication
  - Example (ScriniumBond)
- Debugging
  - Ride
  - Debug-Files
    - (ScriniumCE)
- The purporse of the Class group structure – as seen in the previous slide
- Night Batch
- On-Boarding
  - Array based methods

Design features – II

- Calculation performed:
  - Return Measures as: SR, Annual Return, TWR
  - Risk: MAD, Std and Sharpe
  - VaR: CVaR and VaR
  - Relative Risk: Alpha, Beta, Standard Error
  - Instrument Specific calculations like for example:
    - Duration, Delta, Gamma etc

Design features – III

- For Assets we have: ScriniumAsset, ScriniumBond, ScriniumCFD, ScriniumCurrencyForward, ScriniumETN, ScriniumFRA, ScriniumFloatingMBB, ScriniumGenricForward, ScriniumMoneyMarket, ScriniumOption
  - Instrument specific calculations are performed on Asset-Class level

Design features – IV

- Commen settings for all Assets are:
  - ASSETID
  - ASSETCLASSID
  - SETTLEDAYS
  - TRADINGUNIT
  - CURRENCYCODE

Design features – V

- Transaction Data classes:
  - ScriniumAT
  - ScriniumCT
- Market Data classes:
  - ScriniumCurrencyRates
  - ScriniumPrices
- Special purpose Class:
  - ScriniumBenchmark
  - ScriniumPortfolio
    - Sub–Groups
- The Shared Class: ScriniumCE
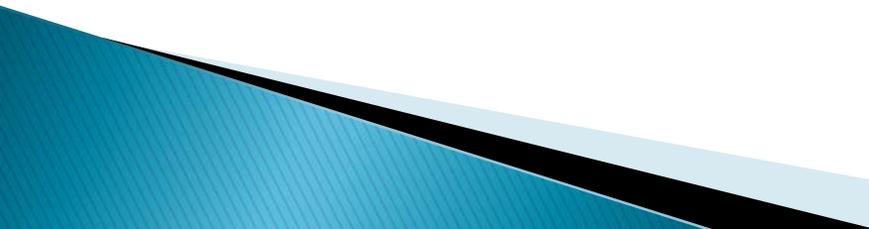
An example

- TEST_DYALOG ☺

End Remarks

▸ The tool set is extremely flexible and versatile

▸ Has no interaction with DB or UI – makes it no dependent on any particulary way this has been designed

▸ The data structure designed for CE is only what is needed to perform calculations – no more no less