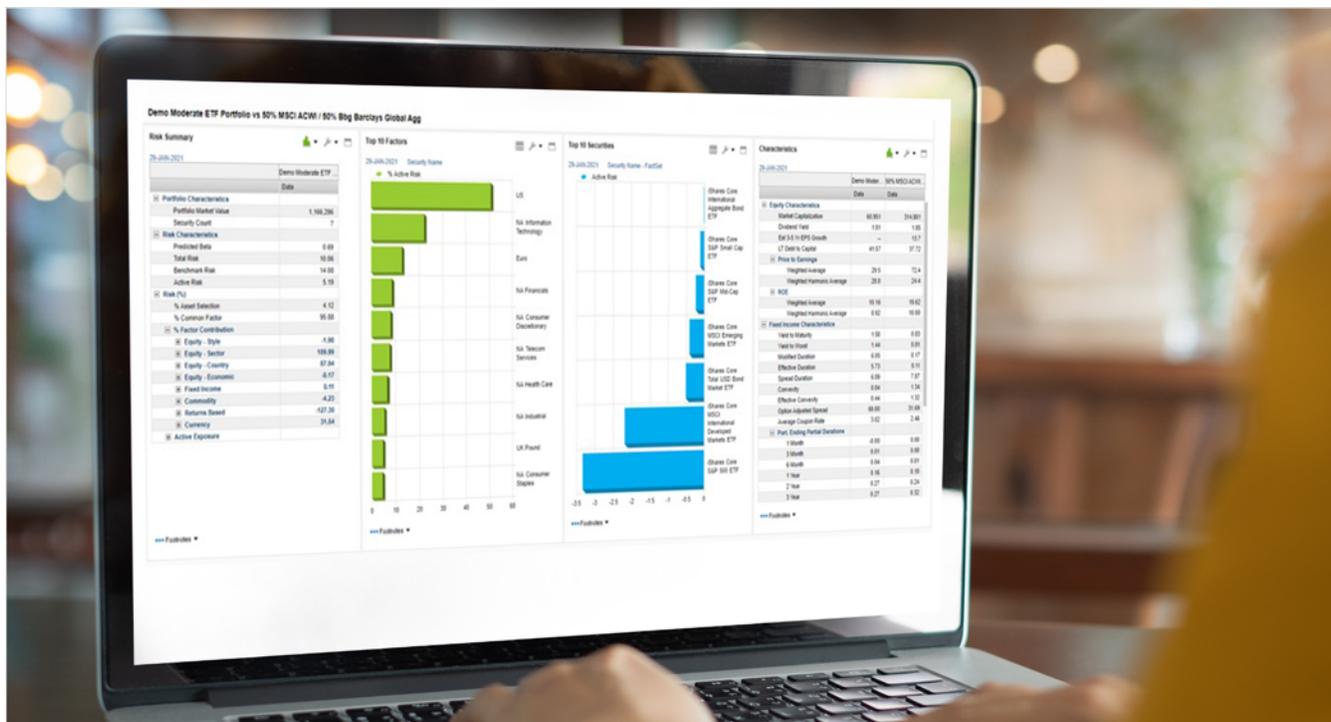


FactSet Portfolio Optimizer

Construct optimal, multi-asset class portfolios according to custom investment objectives while simultaneously adhering to specified constraints and costs.



CONSTRUCT A TRUE MULTI-ASSET CLASS PORTFOLIO

Optimize your portfolio across multiple asset classes at once. While most optimizers divide portfolios into separate, standalone optimization exercises based on asset classes, FactSet's Portfolio Optimizer allows for true cross-asset-class optimizations to better reflect investment intentions and intuitions. Support your equity-only, fixed-income-only, or fund-level-only investment strategies, as well as your multi-asset class portfolio workflows.

OPTIMIZE BASED ON YOUR PREFERRED APPROACH

FactSet Portfolio Optimizer supports two optimization approaches for a more holistic analysis of risk. Optimize according to a parametric approach based upon minimization of risk according to mean variance. Or, gain a competitive advantage by employing a scenario-based approach, which considers downside risk as measured by Expected Tail Loss (ETL). This offers a particularly useful multi-asset class approach for portfolios that hold assets with non-linear payoffs and non-normal return distributions.

FACTSET > SEE THE ADVANTAGE

TARGET UNIQUE, CUSTOM OBJECTIVES

Build the most ideal asset distribution model with a broad range of unique objectives. Leverage traditional objectives to maximize factors such as expected return, minimize costs like financial risk, or introduce objectives including Sharpe Ratio, Stable Tail-Adjusted Return Ratio (STARR), or Diversification Ratio.

The screenshot shows the 'DOCUMENT:MAX UTILITY' configuration window. It includes sections for 'Objectives' and 'Constraints'. The 'Objectives' section has a table with columns for Name, Type, and Settings. The 'Constraints' section has a table with columns for Active, Name, Description, Type, Dirs., Type, Level, Min, Max, Hor., and Additional Settings. Constraints listed include 'Number of Assets', 'Holdings Turnover', 'Turnover', and 'Max Active Risk'.

The screenshot shows the 'DOCUMENT:MIN CTL' configuration window. It includes sections for 'Objectives' and 'Constraints'. The 'Objectives' section has a table with columns for Name, Type, and Settings. The 'Constraints' section has a table with columns for Active, Name, Description, Type, Dirs., Type, Level, Min, Max, Hor., and Additional Settings. Constraints listed include 'Max Asset Weight', 'Debt 80%', 'Equity 80%', and 'Max Tail Loss'.

BENEFIT FROM SEAMLESS FACTSET INTEGRATION

Gain complete flexibility through the integration between FactSet's multi-asset class Portfolio Analysis engine and FPO. Perform sophisticated analysis directly in Portfolio Analysis rather than within a standalone application. Use the Portfolio Optimizer API to conduct fully customizable quantitative research. Design and automate production processes in the FactSet Quantitative Research Environment (FactSet's secure, hosted, Jupyter Lab environment) or any environment of your choosing.

The screenshot shows two parts of the FactSet interface. On the left is a table titled 'MC Risk and Trade Simulation' with columns for Index, Fund, Asset, and various performance metrics. On the right is a 'Risk & Asset Liability Tracking Chart' showing a timeline from 2014 to 2021 with various data points and trends.

```

38     "ifacctestexists": "appenddate",
39     "ifofdbdateexists": "replacedate",
40     "excludezero": True,
41     "archivedate": "20140104"
42 }
43 }
44
45 rolling_date = date(2014, 1, 4)
46 date5 = ''
47 one_week = timedelta(weeks=1)
48 period = 1
49 end_date = date(2021, 2, 9)
50
51 while rolling_date < end_date:
52     dates = rolling_date.strftime('%Y%m%d')
53     risk_model_date = date5
54     backtest_date = date5
55     fpo_output_types['account'][ 'archivedate' ] = date5
56     #print(fpo_output_types)
57
58     if period > 1:
59         strategy_overrides['constraints'] = {
60             "17787663301-0302644b4f9f6": "enable",
61             "1778765ed62-3a26e4a7c7b66": "disable"
62         }
63
64     print(period, rolling_date.strftime('%Y%m%d'), end = ", ", flush = True)
65
66     optimalResults = fpo.optimize(fpo_strategy, fpo_output_types, fpo_acct, pa_document,
67                                 portfolio_override, benchmark_override, risk_model_id_override,
68                                 strategy_overrides,
69                                 risk_model_date, backtest_date, cash_flow)
70
71     if not 'trades' in optimalResults:
72         print(' Issue with this period', flush = True)
73     else:
74         print(len(optimalResults['optimal']), ' optimal', ' ', len(optimalResults['trades']), ' trades')
75     period += 1
76     rolling_date = rolling_date + one_week
77

```