

# Athena: the Python Analytic Module

doc version 0.0.1

Zed Yang

July 12, 2016

## 1 Introduction

## 2 Class Inheritance Hierarchy

## 3 Run

### 3.1 Backtest Example: MA Cross Strategy

Athena has a built-in simple strategy example for backtest. Before he runs any Python scripts, one should always make sure a *Redis* instance is ready for connections.

- (Step.1) Go to `*/Athena/athena_redis-x64` folder and turn on redis server, i.e. execute `run.bat`. As an optional step, one could open `redis-cli.exe` and type `flushall` command to clean up all cached data in this redis instance.

The backtesting program is based upon multiple concurrent Python processes, representing different objects interacting with the public event engine implemented by redis cache. As for this naive MA Cross strategy, one would configure and run **six** different Python scripts in `Athena/examples` folder.

**Channel** is one of the core concepts in the context of event queuing model using Redis. Each item in Athena interacts with the public cache by subscribing and publishing to some particular channels. For example, the moving average signal subscribes to bar signal channel and publishes to its own signal channel; while portfolio class subscribes to market data channels and buy/sell signals, publishes to “portfolio” channel. The table below lists the subscribes and publishes of all relevant objects to backtest this strategy.

Table 1: Pubsub Table		
Object	Subscribes	Publishes
Backtest Driver	market data batch from SQL	['Au(T+D)', 'au1706', 'ag1607', 'ag1611']
1-min Bar Signal	'Au(T+D)'	'signal:bar_1m_Au(T+D)'
Short MA Signal	'signal:bar_1m_Au(T+D)'	'signal:ma_36_bar_1m_Au(T+D)'
Long MA Signal	'signal:bar_1m_Au(T+D)'	'signal:ma_48_bar_1m_Au(T+D)'
MA Strategy	['signal:bar_1m_Au(T+D)', 'signal:ma_36_bar_1m_Au(T+D)', 'signal:ma_48_bar_1m_Au(T+D)']	'strategy:ma_cross'
Portfolio	['Au(T+D)', 'strategy:ma_cross']	'portfolio'

- (Step.2) Execute these six Python processes.
  - `signal_sub.py` is the 1-min Bar object.

- `signal_ma_long_sub.py` and `signal_ma_short_sub.py` are the derived moving average signals based upon bars.
- `strategy_sub.py` is the MA cross strategy object.
- `portfolio_sub.py` is the portfolio.

One should execute these five scripts first, the order does not matter. When they are all ready and in listening state, one can execute `backtest_pub.py` to drive the backtest. The console will prompt out some text, when the backtest terminates, `backtest_pub.py` will exit with code 0 and print the time it takes for the main loop.

- (*Step.3*) When the backtest is over, execute `test_equity_curve.py` script in `Athena/analyzer` directory to plot an equity curve. Note that the database should not be flushed between step 2 and 3, otherwise, the portfolio log will be lost.