Strategy (WLPR) replay demo in MATLAB

**Files:**

* .m file name:

[\\GitHub\matlabtraderdev\replay\_demo.m](file:///\\GitHub\matlabtraderdev\replay_demo.m)

[\\GitHub\matlabtraderdev\replay\_setstrat.m](file:///\\GitHub\matlabtraderdev\replay_setstrat.m)

[\\GitHub\matlabtraderdev\replay\_init.m](file:///\\GitHub\matlabtraderdev\replay_init.m)

* intraday 1m candle stick data file names: [\\GitHub\matlabdatabase\intradaybar\rb1810\rb1810\_yyyymmdd\_1m.txt](file:///\\GitHub\matlabdatabase\intradaybar\rb1810\rb1810_yyyymmdd_1m.txt)
* intraday tick data file names:

[\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810\_20180423\_tick.mat](file:///\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810_20180423_tick.mat)

[\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810\_20180424\_tick.mat](file:///\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810_20180424_tick.mat)

[\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810\_20180425\_tick.mat](file:///\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810_20180425_tick.mat)

[\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810\_20180426\_tick.mat](file:///\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810_20180426_tick.mat)

[\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810\_20180427\_tick.mat](file:///\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810_20180427_tick.mat)

[\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810\_20180502\_tick.mat](file:///\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810_20180502_tick.mat)

[\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810\_20180503\_tick.mat](file:///\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810_20180503_tick.mat)

[\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810\_20180504\_tick.mat](file:///\\GitHub\matlabtraderdev\trading\objs\@cReplayer\rb1810_20180504_tick.mat)

**Note:**

WLPR strategy generates trading signals based on the world-famous William %R technical indicator. Compared to the classical set-up, in which the number of periods parameter is set to 14, our WLPR strategy uses 144. In addition to this, WLPR of -100 indicates the asset is over-sold and hence a buy signal shall be placed, whereas 0 means the asset is over-bought and a sell signal shall be placed therefore. Last but not least, we use 15-minute candle stick in our demo test.

**Design:**

There are 3 scheduled execution jobs, i.e.

1. the job to process the market data from any data source;

\\GitHub\matlabtraderdev\trading\md\@cMDEFut

[\\GitHub\matlabtraderdev\trading\md\@cMDEOpt](\\\\GitHub\\matlabtraderdev\\trading\\md\\@cMDEOpt)

1. the job to process the orders(entrusts) placed;

\\GitHub\matlabtraderdev\trading\objs\@cOps

1. the job to process the trading signals;

\\GitHub\matlabtraderdev\trading\strat\@cStrat

required to run a strategy within the MATLAB-based AUTOTRADE framework. In the real-time trading, job1 runs every 0.5 seconds (this is the same as the CTP tick frequency); job2 runs every 1 second and we shall define the frequency of job3 according to different strategy logic, e.g. every 1m or every 15m.

classes of ‘cMDEFut’, ‘cMDEOpt’, ‘cOps’ and ‘cStrat’ are derived from base class ‘cMyTimerObj’. It is important to note that method called ***‘refresh’*** imbedded in ***‘replay\_timer\_fcn’*** of ‘cMyTimerObj’ plays a key role for defining jobs in the scheduled executions.