TLDR: How BackTrader Executes Code

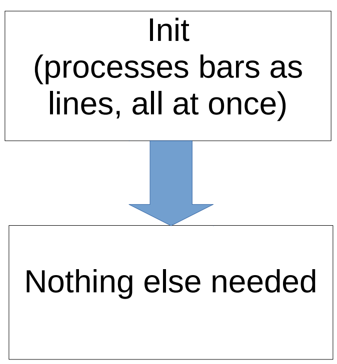
Preonce, Once, Next, Init – Why Isn't My Code Working?

Backtrader evolved from a backtesting only platform to a platform that performs real-time processing of data. It now has signal generation from strategies, optimization and trade execution. Because of this there are different ways for Backtrader to execute indicator code varying depending on whether or not the data is live or historical and preloaded.

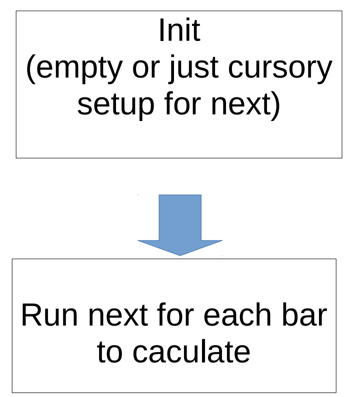
Coding for Backtesting:

This is how Backtrader processes indicator code when backtesting preloaded data:

By LINES

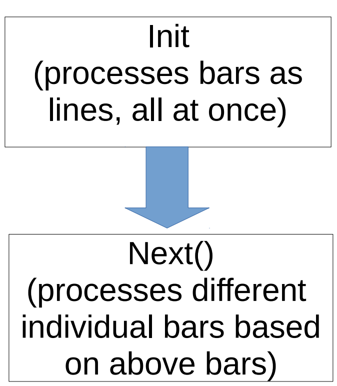


By BARS

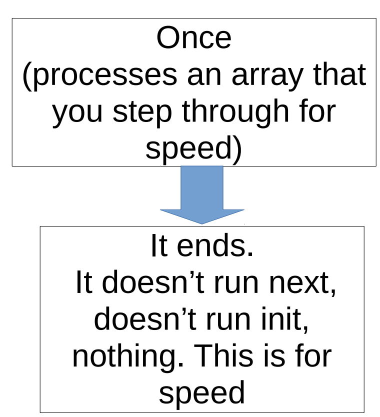


OR

By BARS



By BATCH



That's it – it doesn't execute preonce, once, unless it's running the once routine for this mode of execution. And when running once it doesn't run init or next. And when running init and/or next it is not running once(). For simplicity this diagram excludes prenext and similar.

So you have your file of stored data with timestamps and imported it with read_csv and adddata as various online examples show. You have your timestamps in UTC time and you made sure that they are going from earliest to latest so that backtrader doesn't try to start from the future. Great! You code your routine in next() so that it executes at each bar.

...and it runs!

Fantastic! But it's slow. So you add some preconfiguration in once() to help it along a bit.

And it explodes. You might think "is this platform broken?" Why isn't once running when the bars have reached the minimum period and then next running for each bar? Why? Because that's not what backtrader is designed to do. Once is only for backtesting. Isolated on its own with its brothers preonce, oncestart, etc.

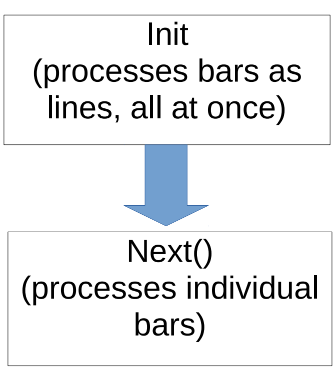
So, fine, you move all your code to the once routine but... wait... you spend hours diagnosing only to find out that init() didn't run and none of your variables are declared/setup/initialized. And where are your precalculated lines from init()? Why don't they exist?

Because that's not the code path when Backtrader runs the once routine. And backtrader will pick which routines to run depending on how the data is supplied. You can force this path by using the runonce flag in cerebro.run().

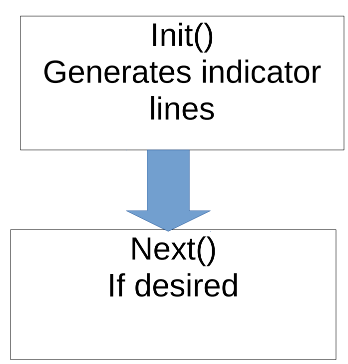
Coding for Real Time Live Data:

This is how Backtrader sees code when running against live real time data:

By BARS



BY LINES



Now you put it to work on real-time code and absolutely nothing happens. Again, once is only for preloaded data for backtesting. In real-time, Backtrader will run init and next to process bars.

Another mistake is to put lines calculations for an indicator in init() and then have additional bar-by-bar calculations for the **same** lines object in next(). Because Backtrader calculates the lines object each bar and triggers this with a timer this will not work.

You can create both once and init/next, but one will run when backtrader detects preloaded data, and one will run in live sessions. This can be useful for batch processing before live trading but you have to be certain that the code logic is duplicated and identical between once and next or you'll have different results in backtesting than live data.

What if data is real-time but backfilled with pre-stored data from a file? Then the real-time execution pathway of init/next will occur.

Written by B. Bradford June 18, 2020