

PythonGo 实盘运维说明

1. 常用操作

1.1 策略加载

每个策略文件，都可以创建多个策略实例（实例名称不能相同）。PythonGo 在原则上支持同时运行的实例个数没有上限；但由于电脑硬件性能和网络带宽的影响，过多的策略同时运行可能引起 tick-to-trade 延时过高，影响交易。

PythonGo 目前不支持策略文件的实时编辑功能，需用户自行编写后，将策略文件复制到指定文件夹下。具体的方法请参考《PythonGo 安装使用文档》2.2.1 章节“存放策略文件”中的内容。以电脑用户名为“Administrator”的用户为例，策略文件的存放路径为：

```
C:\Users\Administrator\AppData\Roaming\InfiniTrader_Simulation\pyStrategy\strategy
```

1.2 策略配置文件

策略的参数配置文件采用 JSON 文件格式保存，文件命名规则为“策略实例名称.json”，其中策略实例名称就是在策略文件创建实例时由用户输入的实例名称。例如运行名为“dma1”的策略实例，其参数配置文件名为“dma1.json”。下次相同策略相同实例名的策略实例被创建时，该配置文件的参数会被自动读取，用户无需重新输入参数。

配置文件中的具体参数，由具体策略文件中的 paramList 自行定义，会动态生成保存。配置文件中的具体数据实际上是一个 Python 的字典，key 对应的是参数的名称，value 值对应的参数的数值。配置文件的存放地址为本地无限易数据目录下的“\pyStrategy\json”目录（具体位置参照 1.1 中的策略文件存放路径）。

1.3 历史数据

PythonGo 中历史数据由公司其他部门提供的接口获取。用户可以通过在策略文件中加载 loadDay 和 loadBar 函数, 分别获取历史日线数据和历史分钟线数据。

- loadDay(years, symbol, exchange, func): 载入历史日线数据, 支持近 2 年的数据; 默认只需要输入载入年数, 即 loadDay(1)来载入数据。
- loadBar(days, symbol, exchange, func): 载入历史分钟线数据, 支持近 3 天的数据; 默认只需要输入载入天数, 即 loadBar(3)来载入数据。

历史数据的具体字段请参考《PythonGo 策略开发文档》的第 2 章节“数据结构详情”中的 VtTickData 和 VtBarData 的内容。

2. 功能限制

2.1 运行实例

每台交易终端可以运行多个策略实例, PythonGo 在原则上支持同时运行的实例个数没有上限; 但由于电脑硬件性能和网络带宽的影响, 过多的策略同时运行可能引起 tick-to-trade 延时过高, 影响交易。

策略文件更新后, 需要在策略管理中移除策略, 再重新加载修改后的策略文件才可以生效

2.2 行情订阅

无限制, 由上联的系统控制。

2.3 委托流量

无限制, 由上联的系统控制。

2.4 撤单限制

无限制, 由上联的系统控制。