

金鸿期源（FMAPI 期货行情接口）用户使用手册

文档版本：V2026.08

适配程序：FMAPI 期货行情动态库

运行平台：Windows x64

文档用途：指导 FMAPI 部署、接口调用、行情落地、数据转换

文档版权：永轩软件 www.yx10000.com

一、产品概述

1.1 主要功能简介

FMAPI 基于 CTP 期货官方行情接口轻量化封装，完整提供三大类能力：原始 Tick 行情回调接收、Tick / 各级周期 K 线本地二进制文件落地存储、多套 K 线数据转换工具函数。

产品特色：

- 全自动行情链路运维：程序启动自动连接行情前置；行情断线自动重连、重订阅全部合约；后台进程全程自动启停维护；
- 内置定点闭合 K 线、结算 Tick 回溯、多线程异步 IO 三大核心能力；
- Tick 落地两种存储模式可选：全量逐笔 Tick 存储、仅成交量变动精简存储；
- 缓存与刷新参数可动态调整：自定义 Tick 缓存 IO 落盘间隔、各周期 K 线内存刷新周期；

核心能力：

- 实时接收 CTP 原始 Tick 行情回调，支持二进制、JSON、XML、制表符多模态数据输出；
- 专业 / 全能版支持 Tick、5 秒 K、1 分钟 K、5 分钟 K、日 K 本地二进制文件自动落地持久化；
- 内置全套数据转换工具函数：Tick 批量合成周期 K、1 分钟 K 向上对齐聚合多分钟 K、通达信 K 线文件解析转标准结构体；
- 划分四级授权版本，不同版本开放行情落地、转换工具等差异化功能权限。

1.2 适用场景

1. 个人 / 机构量化交易、自动化程序化交易策略开发;
2. 本地长期期货行情二进制仓库搭建、历史行情回测;
3. 期货行情网站、量化平台后端稳定行情源对接;
4. 通达信存量历史行情批量迁移, 补齐本地历史数据集;
5. 全自动行情运维, 无需手动处理断线、合约重订阅逻辑;
6. 多格式数据输出适配各类开发语言、数据分析工具。

1.3 开发 & 运行环境

- 操作系统: Windows 10 / Windows 11 / Windows Server x64
- 依赖组件: VC++ x64 运行库、GraphsObject.dll
- 磁盘建议: 全部 Tick、K 线二进制数据存储场景均推荐 SSD 固态硬盘, 大幅降低多线程 IO 阻塞、提升读写性能。

1.4 版本分级与权限说明

软件分为**试用版**、**基础版**、**专业版**、**全能版**四个授权版本, 各版本开放权限划分如下:

■ 试用版

- ✧ 限制: Tick 接收采用随机丢弃数据包限流机制; **无任何文件存储能力**;
- ✧ 开放功能: 自动连接、断线自动重连重订阅、多模态行情回调输出;
- ✧ 限制功能: 二进制落地、全部 K 线转换工具、结算 Tick 回溯、定点闭合、多线程 IO 全部锁定, 无仿真 / 实盘数据隔离。

■ 基础版

- ✧ 限制: **无任何文件存储能力**; 无 Tick 随机丢弃限流;
- ✧ 开放功能: 全自动行情运维、多模态行情回调输出;
- ✧ 限制功能: 二进制落地、全部 K 线转换工具、结算 Tick 回溯、定点闭合、多线程 IO 全部锁定, 无仿真 / 实盘数据隔离。

■ 专业版

- ✧ 定位: 自建本地二进制行情仓库、实时数据落地工具开发;
- ✧ 开放功能: 基础版全部能力 + Tick / 各级 K 线本地二进制落地、双 Tick 存储模式、动态 IO / 刷新参数、多线程 IO、定点闭合 K 线、结算 Tick 回溯;
- ✧ 限制功能: 全套 K 线批量转换工具函数锁定。

■ 全能版

- ✧ 定位：全场景实时存储 + 历史数据转换一体化使用；
- ✧ 开放功能：专业版全部能力 + 全套 K 线数据转换工具函数；
- ✧ 无流量、功能限制，开放文档全部接口。

1.5 开发 & 使用约束

仅支持本地 DLL 函数调用交互。

二、详细功能介绍

2.1 全自动行情链路运维功能

1. 自动连接：程序启动自动连接配置的实盘 / 仿真行情前置；
2. 断线自愈机制：行情链路断开后后台持续轮询重连，连接成功后自动批量重新订阅全部已配置合约；
3. 自动启停：行情服务后台常驻运行，进程启停自动清理缓存、归档当日未落地数据；
4. 内置仿真与生产目录隔离，区分环境逻辑由调用方自行实现。

2.2 原始 Tick 行情回调与多模态输出功能

- 透传 CTP 原始深度 Tick 快照，完整推送盘口、价格、成交、持仓全部字段；
- 试用版 Tick 随机丢弃数据包限流；基础 / 专业 / 全能版无 Tick 丢弃；
- 四种输出格式可选：二进制结构体、JSON 文本、XML 文本、制表符分隔文本，适配各类程序与数据分析工具；
- 回调数据使用全局内存池统一管理，使用者无需手动申请 / 释放内存，避免内存泄漏；
- Tick 缓存 IO 间隔、K 线内存刷新周期支持运行时动态修改，适配高低配机器。

2.3 特色增强功能

- ✓ 定点闭合 K 线：严格按周期时间点自动切割闭合 K 线，保证时序窗口对齐；
- ✓ 结算 Tick 回溯：收盘后自动回溯当日完整 Tick 流，重算当日全部周期 K 线；
- ✓ 多线程异步 IO：读写磁盘采用独立 IO 线程池，行情主线不阻塞，提升并发存储性能；

- ✓ Tick 存储双模式：全量 Tick 逐笔记录、仅成交量变动精简存储，自由切换平衡 IO 压力。

2.4 Tick 与周期 K 本地二进制落地功能

- ✧ 库内置多线程 IO 异步落地，分层二进制文件持久存储：
 - Tick 二进制：支持全量 Tick、仅成交变动 Tick 两种落地模式，按交易日独立文件存储；
 - Sec5（5 秒 K）：Tick 实时聚合，定点窗口批量写入对应交易日文件；
 - Min1（1 分钟 K）、Min5（5 分钟 K）：Tick 驱动实时合成，单合约环形跨交易日文件；
 - DayK（日 K）：每日收盘借助结算 Tick 回溯统一合成写入单合约全历史文件；
 - 自动递归创建输出根目录、分类子目录，无需上层手动创建文件夹；
 - 落地开关可独立控制，按需开启 / 关闭对应周期存储，降低磁盘 IO 负载。

2.5 K 线数据转换工具函数

提供三套独立批量转换接口，输入原始行情数组，输出堆分配标准 KlineData 内存数组：

- ✓ Tick 批量聚合周期 K：输入 Tick 数组，生成 5 秒 / 1 分钟 / 5 分钟闭合 K；
- ✓ 1 分钟 K 向上对齐多分钟聚合：目标分钟自动向下取就近 5 的倍数（5~55 分钟），窗口向上对齐；
对齐规则示例：5 分钟周期，09:01、09:04、09:05 划归 09:05 窗口，09:06 划归 09:10 窗口，末尾不完整窗口正常输出；
- ✓ 通达信 TDX 文件解析转换：读取 .day/.lc1/.lc5 期货 K 线，自动修正夜盘时间戳，仅填充开高低收、成交量、时间戳，成交额 / 持仓变化 / 成交次数置 0。

三、基础使用配置指南

3.1 授权激活步骤

1. 调用接口读取本机硬件授权识别码；
2. 前往官网 www.yx10000.com 购买对应版本授权；
3. 程序启动注入授权码完成校验。

3.2 核心可配置项

- ✓ CTP 行情前置实盘 / 仿真 IP、端口自定义配置；
- ✓ 二进制行情文件输出根目录自定义（专业 / 全能版生效）；
- ✓ 各周期落地存储独立开关（仅专业 / 全能版生效）；
- ✓ Tick 落地模式切换：全量 Tick / 仅成交量变动 Tick；
- ✓ Tick 缓存 IO 落盘时间间隔、各周期 K 线内存刷新周期（支持运行时动态修改）；
- ✓ 本地运行日志开启 / 关闭开关；
- ✓ 全局订阅合约列表配置，断线重连后自动读取列表批量重订；
- ✓ 行情回调输出格式配置：二进制 / JSON/XML / 制表符文本任选。

四、数据维护说明

4.1 数据生命周期管理

1. 行情回调数据由全局内存池统一管理，无需手动释放；
2. 专业 / 全能版开启落地后，行情自动通过多线程 IO 写入二进制磁盘文件；
3. 二进制文件按合约、周期、交易日分层存储，支持结算 Tick 回溯重算 K 线；
4. 转换工具输出堆内存数组，调用方手动 `delete []` 释放，不会自动落地磁盘。

4.2 异常处理规则

1. 试用版 Tick 随机丢弃数据包，日志输出限流提示；
2. 低版本调用落地 / 转换 / 特色工具接口：函数直接返回 0 / 空指针，仅写入日志，无弹窗；
3. 磁盘空间不足 / 目录无写入权限：落地日志输出警告，停止写入文件；
4. TDX 文件路径错误、文件损坏：转换接口返回 0 条，输出错误日志；
5. 行情断线自动重试连接，重连成功自动批量恢复所有合约订阅；
6. 收盘结算 Tick 回溯失败会单独输出错误日志，不影响实时行情接收。

五、核心数据结构体定义

5.1 TickData 原始 Tick 结构体

```

cpp
struct TickData {
    int64_t timestampSec;    /// Unix 秒时间戳
    uint16_t millisec;       /// 毫秒 0~999
    double lastPrice;        /// 最新价
    double openPrice;        /// 开盘价
    double highestPrice;     /// 最高价
    double lowestPrice;      /// 最低价
    double PreClosePrice;    /// 昨收价
    double averagePrice;     /// 当日均价
    uint32_t volume;         /// 现量
    uint64_t openInterest;   /// 当前总持仓
    uint64_t totalVolume;    /// 累计总成交量
    double totalTurnover;    /// 累计总成交额
    double bidPrice1;        /// 买一价
    int32_t bidVolume1;      /// 买一量
    double askPrice1;        /// 卖一价
    int32_t askVolume1;      /// 卖一量
};

```

5.2 KlineData 标准周期 K 结构体

```

cpp
struct KlineData {
    uint64_t timestamp;      /// K 线结束 Unix 秒时间戳
    double openPrice;        /// 开盘价
    double highestPrice;     /// 最高价
    double lowestPrice;      /// 最低价
    double closePrice;       /// 收盘价
    uint64_t volume;         /// 周期成交量
    double turnover;         /// 周期成交额
    int positionChange;      /// 周期持仓变化
    int tradeCount;          /// 周期成交次数
};

```

5.3 FixedQuoteData 五档行情快照结构体

```

cpp
struct FixedQuoteData {

```

```

char InstrumentID[81];          ////// 合约代码
char ExchangeID[9];            ////// 交易所代码
char TradingDay[9];            ////// 交易日 YYYYMMDD
char ActionDay[9];             ////// 业务日期
char UpdateTime[9];            ////// 更新时间 HH:MM:SS
int UpdateMillisec;            ////// 更新毫秒

double LastPrice;              ////// 最新价
double AveragePrice;           ////// 当日均价
double PreClosePrice;          ////// 昨收盘
double OpenPrice;              ////// 今开盘
double HighestPrice;           ////// 最高价
double LowestPrice;            ////// 最低价
double ClosePrice;             ////// 盘中临时收盘价
int Volume;                    ////// 成交量
double Turnover;               ////// 成交金额
long long OpenInterest;        ////// 当前持仓量
long long PreOpenInterest;      ////// 昨持仓量
double PreSettlementPrice;      ////// 上次结算价
double SettlementPrice;         ////// 当日结算价
double UpperLimitPrice;         ////// 涨停价
double LowerLimitPrice;         ////// 跌停价

// 五档买卖盘
double BidPrice1; int BidVolume1;
double AskPrice1; int AskVolume1;
double BidPrice2; int BidVolume2;
double AskPrice2; int AskVolume2;
double BidPrice3; int BidVolume3;
double AskPrice3; int AskVolume3;
double BidPrice4; int BidVolume4;
double AskPrice4; int AskVolume4;
double BidPrice5; int BidVolume5;
double AskPrice5; int AskVolume5;
};

```

六、注意事项

1. 行情回调输出数据依托全局内存池自动管理，开发者无需手动申请、释放内存，无内存泄漏风险；

2. 支持二进制、JSON、XML、制表符文本四种回调输出格式，适配各类编程语言、数据分析脚本；
3. 全部 Tick、K 线二进制存储场景必须使用 SSD 固态硬盘，机械盘会产生严重 IO 延迟、行情阻塞；
4. 行情断线后自动重连并批量恢复所有合约订阅，无需开发者手动编写重订阅逻辑；
5. 本接口仅供单位、个人或团体量化学习、非面向商用开发，商业盈利场景需单独申请商用授权。

七、版权与免责声明

7.1 版权声明

1. 金鸿期源 FMAPI 期货行情接口所有程序、代码、文档、接口规范著作权归属永轩软件独家所有，官网：www.yx10000.com；
2. 未经官方书面授权，禁止逆向、反编译、篡改、分发、二次封装本软件及配套动态库；
3. 软件分试用版、基础版、专业版、全能版，各版本功能权限由官方授权体系管控，严禁绕过授权校验；
4. 任何私自破解、篡改分发行为，著作权方保留追究民事、刑事责任的全部权利。

7.2 免责声明

1. 用户下载、启动、调用本软件即代表完整阅读、理解并同意本全部免责条款；如不认可以下全部内容，请立即停止一切使用并完整卸载本软件及相关文件；
2. FMAPI 仅对 CTP 原始行情数据做转发、缓存、格式转换处理，不保证行情数据实时、连续、准确、完整，不承诺数据源稳定不间断；
3. 本软件适用于量化交易、自动化程序化交易、本地期货行情仓库、期货行情站点行情源等各类场景，但所有基于接口数据产生的交易决策、策略回测、线上服务运营行为，全部风险由使用者自行独立承担；
4. 因行情延迟、数据缺失、数据错误、网络中断、程序异常、硬件故障、第三方 CTP 前置限制等造成的账户亏损、资金损失、业务停摆、经济赔偿等一切后果，开发者永轩软件不承担任何赔偿、担保、连带责任；
5. 使用者需自行遵守期货市场监管规则、交易所交易规范、网络安全相关法律法规，若利用本软件开展违规、违法操作，全部法律责任由使用者自行承担；

6. 仅官方渠道发布的标准程序包经过安全校验，第三方修改、破解版本可能存在数据劫持、崩溃、病毒风险，由此引发的所有问题开发者概不负责；
7. 无论何种情形，对于用户因使用本软件产生的直接、间接、预期收益损失，开发者均无需承担任何补偿与赔偿责任；
8. 本免责条款、版本权限规则、版权规则可随版本迭代更新，更新后无需单独通知用户，持续使用即视为接受最新条款。