尊敬的易达交易系统用户：

为了更好服务投资者，易达交易系统发布了1.108版本，关于该版本的更新内容详情如下。

# 更新内容

## ydServer

### 支持高可用部署

在高可用部署情况下，至少需要两台易达柜台服务器，一台仲裁服务器，构成一个服务集群。在一台柜台服务出现异常时，仲裁服务会判断自动启用另一台柜台服务。

ydApi客户策略程序需要配置集群中的多个柜台地址和端口，在柜台发生切换时，ydApi可以自动进行切换处理，投资者的程序无感知。

高可用功能需要额外的授权，如有需要请联系我们。

### 改进大商所组合保证金算法

新版中我们调整了大商所组合保证金算法，大部分组合保证金能做到和CTP完全一致。

但买期权（期权对锁、买入期权垂直价差 、卖出期权垂直价差、买权覆盖）相关保证金无法做到和CTP完全一致，这个问题待后续版本解决。

需要再次强调的是，易达系统默认关闭了买期权组合保证金功能，关闭情况下，买期权能正常进行组合，但是不会节约保证金；该功能开启后，将会组合减免，且平空头仓位时不会检查保证金。

### 支持大商所七期接口

易达交易系统支持大商所七期交易接口，在安装的时候请使用./install dce7 prod或者./install dce7 test进行安装和更新，原六期系统使用./install dce6 prod或者./install dce6 test。

### 支持质押资金

易达可以将质押资金并入昨权益供投资者使用，选择打开该功能后，质押资金会视为普通权益供客户开仓使用，易达目前并不控制质押资金的出金。

质押资金功能默认关闭，可以在config/SystemParam.csv中添加或更新以下内容打开该功能：UseCollateral,Account,1。

### 支持CTP异构数据

易达交易系统支持使用CTP异构数据初始化，将数据按照和sync\_merge相同的方法放到/home/trade/product\_env/data\_source/ctp\_heter目录下即可。

### 支持账户级别的风控参数和自成交检查

易达交易系统支持买卖开仓量，多空持仓量，成交量以及撤单笔数等在产品和合约级别对账户进行设置。

易达交易系统同时也支持自成交检查在账户级别控制，业务人员可以通过网页端进行设置。请特别注意，易达系统修复了FAK/FOK单纳入自成交检查设置的问题，请酌情决定是否需要关闭该功能。

### 其他新增功能和问题修复

1. 支持上期所做市商接收询价号和报价时自动填写询价号的功能。
2. 支持大商所期权对冲功能。
3. 支持交易所撤单失败在notifyFailedCancelOrder返回。
4. 支持多席位同时接受回报的优选功能。
5. 支持中金所市价单。
6. 允许撤销报价产生的衍生委托。
7. 改进了保证金和持仓盈亏算法，使得刷新更为高效和频繁。
8. 修复了期权行权回报可能会被丢失的问题。

## ydAPI

### 兼容性说明

本次更新为协议兼容升级，具体更改内容参见<https://www.hanlinit.com/docs/programming-guide-108/>，本次版本新增、修改了一些结构体和API方法，请比较新老版本的改动是否会对投资者的交易产生影响。

* ydApi用户仍然可以使用原有的客户端交易，但我们仍然强烈建议能升级本API。
* ydExtendedApi的用户必须更新并重新编译程序，否则会导致保证金计算不准确，注意保证金和持仓盈亏的调用方式发生了改变。
* 裸协议接口没有变化，投资者可以继续使用现有协议格式报撤单。

### 支持高可用

ydApi支持高可用，允许在配置文件里设置多个柜台地址和端口。在客户端切换到备节点前，ydApi会推送*YD\_AE\_ServerSwitched*事件。

具体客户端配置参照如下：

# Count of recovery site. Used to achieve high availablity at the expen

se of a little performance of order notification.

# 0 for no recovery, 1 for recovery always use primary site, 2 for recovery use primary and secondary sites

RecoverySiteCount=0

# IP address of primary trading server

TradingServerIP=127.0.0.1

# TCP port of primary trading server, other ports will be delivered after logged in

TradingServerPort=51000

# IP address of secondary trading server.

# Valid only when RecoverySiteCount equals to 2.

TradingServerIP2=127.0.0.2

# TCP port of secondary trading server, other ports will be delivered after logged in.

# Valid only when RecoverySiteCount equals to 2.

TradingServerPort2=51000

### 全新设计的资金计算

新版中我们重新设计了保证金和持仓盈亏的计算方式，新的方式更加高效，建议投资者都能使用“auto”方式来计算保证金和持仓盈亏。

Config.txt的具体配置如下：

# Work mode for recalculation of margin and position profit. Valid when

using ydExtendedApi.

#     auto(default): subscribe market data and automatically recalculate in proper time.

#     subscribeOnly: subscribe market data and recalcMarginAndPositionProfit should be called explicitly

#     off: never do recalculation

RecalcMode=auto

# Gap between recalculations, in milliseconds. Valid when RecalcMode is set to auto.

# It will be adjusted to 1000 if less than 1000

RecalcMarginPositionProfitGap=1000

# Delay of recalculation after market data arrives to avoid collision with input order, in milliseconds.

# Valid when RecalcMode is set to auto. Should be between 0 and 100.

RecalcFreeGap=100

原来的几个更新保证金和持仓盈亏的方法删除，

~~virtual void updatePositionProfitAndMargin(const YDInstrument \*pInstrum~~

~~ent)=0;~~

~~virtual void updatePositionProfit(const YDInstrument \*pInstrument,double positionPrice)=0;~~

~~virtual void updateMargin(const YDInstrument \*pInstrument,double marginBasePrice)=0;~~

~~virtual void updatePositionProfitAndMargin(const YDInstrument \*pInstrument,double positionPrice,double marginBasePrice)=0;~~

新的更新保证金和持仓盈亏方法为：

// recalculate margin and position profit. if "RecalcMode" is "auto", t

here is no need to call this method

virtual void recalcMarginAndPositionProfit(void)=0;

同时，为了便于ydExtendedApi的投资者更方便试算保证金，新增了两个函数，分别用于计算持仓的每手保证金和每个组合节约的保证金。

/// anyDirection can use Direction or PositionDirection

virtual double getMarginPerLot(const YDInstrument \*pInstrument,int hedgeFlag,int anyDirection,double openPrice,const YDAccount \*pAccount=NULL)=0;

virtual double getMarginPerLot(const YDExtendedPosition \*pPosition,double openPrice)=0;

virtual double getCombPositionMarginSaved(const YDExtendedCombPositionDetail \*pCombPositionDetail,double legMargins[])=0;

### 支持柜台回报

易达可以支持柜台回报，即在通过校验并发往交易所以后，向投资者通告这一状况，该回报的状态为YD\_OS\_Accepted，除了交易所回填的InsertTime和OrderSysID两个字段外，其他字段都可以正常使用。

柜台回报可以用于了解柜台是否收到了报单，也可以用于外部接收报单时与柜台内部报单进行关联，考虑到易达的本地保单号OrderLocalID是柜台全局唯一的，所以可以使用OrderLocalID进行关联。

该功能需要向期货公司申请打开，期货公司可以在网页端管理界面为客户打开“柜台回报”功能。

### 新增可用资金（扣除持仓亏损）

YDExtendedAccount中的Available并不包含持仓盈亏，因此也造成了部分投资者的误解。新版本中我们新增了两个方法：

* useable()获取用户当前的可用资金（扣除持仓亏损）
* canUse()方法判断当前可用资金（扣除持仓亏损）是否足够

### 交易风控参数移到账户层面

柜台的交易风控参数TradingConstraints已经支持在账户级别设置，相应的这个风控参数信息在ydApi里，也移到了账户级别YDAccountProductInfo和YDAccountInstrumentInfo。具体请参考API接口编程文档。

### 改进了撤单失败返回

ydApi在客户撤单失败的情况下，不管是交易所拒绝，还是被柜台拒绝都会通过notifyFailedCancelOrder返回失败信息。

### 其他新增功能和问题修复

1. getYDNanoTimeStamp()方法从ydUtil.h搬到ydApi.h
2. ydApi中增加getTradingDay()方法，获取交易日。
3. 新增WIN64位API
4. yd.so支持直接运行以打印版本号

## ydClient

### 支持多环境

ydClient可以支持同时配置多个环境并在多个环境中切换的功能，具体方法为：

* 在当前ydClient柜台下面增加config目录
* 把多个后缀为.ini文件放入config目录，文件名任意。
* 可以在ini文件里加入name变量，显示柜台名称，请尽量使用英文字符，如果要使用中文字符，请确保文件使用GBK编码。

### 支持数据导出

支持导出账户资金，持仓，委托，成交，报价，合约，费率，交易权限，方便进行数据核对。

### 支持一键撤单和撤报价

每次点击全部撤单或者全部撤销报价，易达客户端会帮助客户把所有挂单中的委托和报价撤销。

### 其他新增功能和问题修复

1. 在ydClient中增加出金校验，ydServer和ydAPI不做校验
2. 优化了ydClient的显示性能
3. ydClient界面支持筛选合约

# 更新说明

1. **由于此次更新内容较多，本版本已经在部分期货公司的生产测试环境试运行了近2个月，目前没有问题反馈。为了稳妥推进系统升级，我们将与你们协商升级计划，分批次逐步升级现存生产系统，限制全市场每周升级套数。**
2. 本次更新为**协议兼容升级**。具体更改内容参见https://www.hanlinit.com/docs/programming-guide-108/。
	1. 使用ydApi的投资者仍然可以使用原有的客户端交易，但我们仍然强烈建议能升级本API。
	2. 使用ydExtendedApi的用户必须更新，注意保证金和持仓盈亏的调用方式发生了改变。
3. 为了确保投资者能顺利过渡到新版本上，我们开放了新版本的互联网测试环境，使用方法请参见<https://www.hanlinit.com/docs/dev-environments/>。