**综合交易平台**

**多活交易中心介绍**

**（培训材料）**

**目 录**

[1 文档介绍 1](#_Toc346776653)

[1.1 文档目的 1](#_Toc346776654)

[1.2 读者对象 1](#_Toc346776655)

[1.3 参考文献 1](#_Toc346776656)

[1.4 术语与缩写解释 1](#_Toc346776657)

[2 概述 2](#_Toc346776658)

[2.1 背景 2](#_Toc346776659)

[2.2 方案 3](#_Toc346776660)

[2.3 系统总体架构 3](#_Toc346776661)

[2.4 多活架构的数据同步实现 6](#_Toc346776662)

[2.4.1 远程同步组件-DRMT 7](#_Toc346776663)

[2.4.2 系统启动数据同步-Sync\_merge脚本 10](#_Toc346776664)

[2.4.3 回写数据-Oprdispatcher 11](#_Toc346776665)

[2.4.4 风控系统-riskdrmt 12](#_Toc346776666)

[2.5 多活架构的数据同步流程 14](#_Toc346776667)

[2.5.1 日常盘后数据同步操作流程 14](#_Toc346776668)

[2.5.2 日常盘中数据同步操作流程 16](#_Toc346776669)

[2.6 综合交易平台FENS机制的原理 17](#_Toc346776670)

[2.6.1 综合交易平台FENS跟交易API配合工作 17](#_Toc346776671)

[2.6.2 正常登录时工作流程 19](#_Toc346776672)

[2.6.3 投资者切换交易中心时工作流程 20](#_Toc346776673)

[2.7 多活交易中心的启动 20](#_Toc346776674)

[2.8 投资者、操作员与交易中心的对应关系 22](#_Toc346776675)

[2.9 多活切换流程 22](#_Toc346776676)

[2.9.1 场景一（所有交易中心工作正常，部分投资者在主中心交易和转账） 24](#_Toc346776677)

[2.9.2 场景二（所有交易中心工作正常，部分投资者在备中心交易） 25](#_Toc346776678)

[2.9.3 场景三（所有交易中心工作正常，从主中心切换部分投资者到备中心） 27](#_Toc346776679)

[2.9.4 场景四（所有交易中心工作正常，从备中心切换部分投资者到主中心） 28](#_Toc346776680)

[2.9.5 场景五（主中心正常，某个备交易中心宕掉，主中心接管该备中心） 29](#_Toc346776681)

[2.9.6 场景六（主中心宕掉，备中心接管主中心所有投资者，备中心变为主中心） 32](#_Toc346776682)

[2.9.7 场景七（某个分交易中心出现故障，备交易中心接管出现故障的分交易中心） 35](#_Toc346776683)

[2.9.8 场景八（主中心接管某个分中心，备中心又接管主中心） 37](#_Toc346776684)

[2.9.9 场景九（主中心接管某个分中心，备中心接管另一个分中心，备中心又接管主中心） 40](#_Toc346776685)

[2.9.10 场景十（备中心接管某个非CTP系统） 43](#_Toc346776686)

# 

# 文档介绍

## 文档目的

本册培训教材的编写目的是使综合交易平台系统技术人员对综合交易平台多活交易中心系统的实现和维护有一个比较清晰的了解，并进而掌握。

## 读者对象

综合交易平台系统相关技术人员。

## 参考文献

## 术语与缩写解释

|  |  |
| --- | --- |
| 缩写、术语 | 解 释 |
| 主用系统 | 即俗称的主席系统 |
| 次用系统 | 即俗称的二席系统 |
| 主交易中心 | 在灾难没有发生之前，负责主要交易功能；负责所有基础数据维护；负责开户、结算、报送。 |
| 备交易中心 | 在灾难没有发生前，负责少数投资者的交易功能；在灾难发生后，备中心接替主中心，负责主要交易功能。 |
| 分交易中心 | 只负责少数投资者的交易功能，可以不承担接替主交易中心的功能。 |
| FENS | Front End Name Server：前端名字服务程序（简称FENS） |

# 

# 概述

## 背景

我国期货市场自创建起就一直使用全电子化方式交易，交易系统一直是交易所和期货公司最为核心的技术系统。近年来期货市场的品种不断创新，交易量不断创新高，整个行业对于电子化交易的依赖程度也是越来越高。

对于期货公司而言，集中的电子化交易一方面在降低扩张成本、降低业务风险、整合公司资源上起到了积极作用，另一方面也把系统风险集中到了一起。在大交易量、多交易客户的情况下，期货公司的电子化交易系统如果出现故障，导致交易暂停将带来重大的经济损失，甚至公司形象也将因此受损。电子化交易系统是必要的、重要的，但是同时又是脆弱的，这是计算机系统的普遍问题。不可预测的自然灾害（地震、台风、水灾、雷电、火灾等）；基础设施的损坏（CPU、硬盘损坏、通讯故障、建筑物倒塌、电源中断等）；操作失误（误操作、人为蓄意破坏等）；病毒攻击等这些复杂的因素都导致了计算机系统本身的脆弱。

目前国内外各行各业重大机构纷纷建立起来各自的灾备中心，如上海证券交易所的同城容灾备份中心、深圳证券交易所的“基于主机平台的两地三中心灾备”、上海期货交易所的异地灾备中心、郑州商品交易所的同城灾备中心以及大连商品交易所的同城灾备中心等。

2009年7月出台的《期货公司信息技术管理指引》（以下简称《指引》）提出期货公司也应在条件成熟时为核心交易系统建设异地灾备系统。而即将出台的《证券期货业信息系统备份能力标准》（以下简称《标准》），更是明确规定金融行业包括期货公司核心交易系统的系统备份能力建设等级标准。

上海期货信息技术有限公司（以下简称上期技术）在监管层提出的各类交易系统异地备份要求基础之上，前瞻性的提出期货公司核心交易系统应利用物理距离超过600公里以上的多地机房，以及上期技术研发的综合交易平台（以下简称CTP系统）构建多活交易中心，除了当主交易中心出现故障或者灾难时，备交易中心迅速接管主交易中心业务，另外，最重要的一点，备交易中心在日常情况下也能提供交易，是真正“活着”的备中心。CTP系统技术上的独特与先进性使得能够实现“一键运维”和“一键切换”。

## 方案

每个交易中心都有独立席位连接国内四家期货交易所，每个交易中心都可以供客户连接交易，主结算在主交易中心的主用系统进行，结算后数据统一分发给备交易中心主用系统以及主交易中心的次用系统，盘中通过同步组件实现资金、权限等所有上场数据在主交易中心的主用系统与备交易中心、各个分交易中心之间的同步，保证每个交易中心除了投资者/交易员的交易权限，其余数据保持一致，从而可以做到：日常时，投资者/交易员可以选择在任一交易中心交易，当某一交易中心发生灾难时，别的交易中心可以立刻接管灾难发生交易中心投资者/交易员的交易。

## 系统总体架构

CTP多活交易中心系统采用多套交易系统实时热备方案，系统总体架构如下图：



如上图所示：综合交易平台部署主交易中心1和备交易中心2，还可以继续扩展备交易中心3，备交易中心4等等，每个交易中心可以都部署完整的综合交易平台系统，包括交易、风控、结算等子系统，其中主交易中心和备中心都可以部署银期模块，但因为银行的白名单制度，只能一套系统在生产时间接银行的银期转账。为了在发生灾备切换的时候备中心能够接管主中心的银期转账业务，需要从备中心向主中心所在的当地分行拉一条专线，该专线在主中心运行正常或者主中心到当地分行的线路正常的时候可以作为备用线路。当主中心宕掉或者主中心到当地分行的线路故障时，该线路可以启用。

每个交易中心都有独立席位连接四家期货交易所，加上综合交易平台是一套交易和结算完全分离的系统，交易可以完全脱离物理数据库的系统，因此每个交易中心都可以供客户连接交易，盘后只有主交易中心做结算。

首先大致介绍一下多交易中心日常运行的数据流：

（1）：前一交易日主交易中心结算后，主交易中心结算数据库通过数据库增量备份方式同步给几个备交易中心，保证下一日初始数据的一致性。**当然，如果用户采用专业的第三方数据库实时同步工具，盘后就不必增量备份，大大节省了运维时间**。同时，备交易中心在盘中会将部分数据，例如交易用户事件、投资者修改交易密码等信息回写到备中心物理数据库，为保证主中心数据库数据的完整性，这些信息需要在盘后导回主交易中心，保证这些交易特征明显的数据（同步之前主交易中心并未掌握）的一致性。

（2）：交易系统初始化：其中主交易中心的交易系统通过交易初始化组件实现数据库启动初始化（这是由综合交易平台的体系架构决定的，交易初始化数据来源于保存在物理数据库中的上一交易日结算后的数据），备交易中心则使用前一交易日在主交易中心生成的初始化文件来初始化。初始化指令由DRMT同步到各个分交易中心，各个分交易中心接到指令后也会立即完成初始化操作。

（3）：当各交易中心的交易系统开始交易时，从交易所收取相同的报单回报，成交回报。主交易中心的交易系统，通过tmdb以及dbmt组件仍然可以实现与数据库的互动，但是备交易中心的交易系统，在盘中只使用tmdb组件将数据回写给备交易中心的交易数据库，其交易所需的资金等信息则全部由主交易中心同步过来。

（4）：主交易中心的客户做银期转账，手工出入金，开户等操作，主交易中心通过drmt组件将这些数据同步给几个分交易中心的前置，再分别上场给各备交易中心的交易核心，确保几个交易中心盘中数据一致。 备交易中心的客户做银期转账、密码修改等业务时，可以连接到主交易中心的前置进行操作的，出入金、密码等信息实时同步到备中心。

（5）：正常情况下此流程不会进行，只有当主交易中心因为不可抗力导致不能正常运行后，启动灾备切换进程，备交易中心的结算数据库通过dbmt组件向交易核心发出切换指令，备交易中心交易系统接管所有主交易中心交易用户。

## 多活架构的数据同步实现

综合交易平台的多活的架构的建立，最关键的是数据同步的实现，综合交易平台提供了相应的组件来实现数据的同步：DRMT、RiskDrmt、Sync\_merge、OperationDispatcher。

对于数据库级的实时同步，建议采用专业的第三方数据库实时同步工具，用户可以根据实际情况自行选择。



### 远程同步组件-DRMT

1. 同步的数据

交易数据包括了投资者的委托和成交，投资者费率信息，资金信息，投资者和操作员的权限信息等。其中投资者的委托和成交通过交易所的回报同步到各个中心中，其他交易数据通过DRMT同步给各个交易中心。上述的“其他交易数据”如下表显示：

|  |  |
| --- | --- |
| **实时同步上场数据** | |
| **名称** | **操作** |
| 操作员 | 新增操作员 |
| 授权 |
| 修改操作员角色对应关系 |
| 修改密码 |
| 重置操作员密码 |
| 注销操作员 |
| 操作员IP限制 |
| 新增超级操作员 |
| 超级操作员授权 |
| 角色管理 | 新增角色，设置角色权限对应关系，并授权于某操作员 |
| 修改已有角色权限对应关系 |
| 经纪公司基本参数设置 | 可提资金是否包含平仓盈利 |
| 可用资金是否包含平仓盈利 |
| 本日无仓且无成交客户出金是否受可提比例限制 |
| 银期可提资金算法 |
| 银期资金提取比例 |
| 是否启用用户事件 |
| 投资者 | 开户 （基本信息：投资者号/投资者名称/证件类型/证件号/通讯地址/投资者组/电话、资金/交易密码） |
| 修改交易/资金密码 |
| 重置交易/资金投资者密码 |
| 修改投资者基本信息 |
| 投资者合约权限管理 |
| 投资者交易权限设置（条件单设置不实时上场） |
| 投资者交易编码设置 |
| 注销投资者 |
| 变更投资者属于的投资者组 |
| 新增投资者与动态令牌的对应关系 |
| 注销投资者与动态令牌的对应关系 |
| 激活投资者与动态令牌的对应关系 |
| 手动同步用户动态令牌 |
| 资金管理 | 投资者出/入金、冲销 |
| 银期转账出入金、冲销 |
| 投资者质入/质出 |
| 资金冻结 |
| 银期转账 | 投资者可提资金比例设置 |
| 转账限额设置 |
| 费率属性设置 | 投资者保证金率基础新增、修改、删除 |
| 投资者保证金率调整新增、修改、删除 |
| 当前费率设置 | 投资者保证金率基础新增、修改、删除 |
| 投资者保证金率调整新增、修改、删除 |
| 手续费率设置 | 投资者手续费率模板新增、修改、删除 |
| 投资者手续费率新增、修改、删除 |
| 投资者手续费率模板对应关系设置 |
| 通知设置 | 设置通知书语句 |
| 灾备权限设置 | 删除投资者/操作员用户权限 |
| 修改投资者/操作员用户权限 |
| 新增投资者/操作员用户权限 |
| 灾备批量权限分发 |

1. DRMT工作原理和产生的流水

DRMT的输入流水即是中心系统排队机的输出流水tseries。

DRMT的输出流水是过滤后，只保留上述表格数据的FTDseries

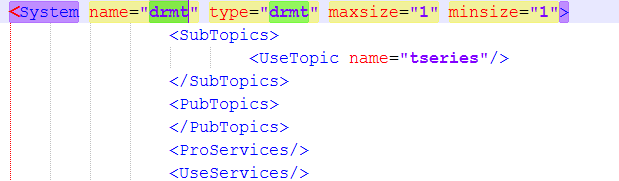
DRMT作为订阅者也接入主中心的FIB，接收tseries；然后根据配置的用户名从主中心中拿到用户名对应的密码，通过前置登录其他中心，将FTDseries中的数据发布到其他中心的FIB中。

DRMT工作时候，syslog的输出信息如图：

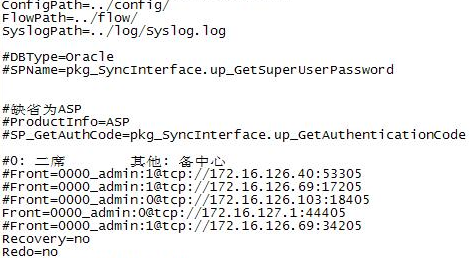


1. 全局配置文件的Systemconfig.xml的变化

增加了drmt的配置



1. DRMT的ini中的重要配置项



其中，Recovery的yes表示灾后启动备中心，no表示正常启动主中心；Redo的作用是当灾后启动备中心的drmt的时候，是否从头处理已存在的series流水。

1. DRMT的重启

drmt启动一段时间后，因某种原因，需要停掉或进程已经被终止，则可以直接带流重启，此时已有的流水不会处理，这个是根据drmt自己记录的发给各前置的报文数来决定的

### 系统启动数据同步-Sync\_merge脚本

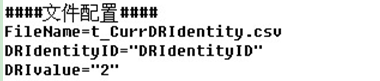
Sync\_merge脚本作用在主中心，导出sync schema下的数据，分发给其他中心。它的ini中重要的配置如下：



表明了是否只抽取隶属于目标中心的投资者的信息。



抽取哪个应用单元的数据



修改目标中心号



是否让其他中心启用用户事件



如果其他中心有库，则在导出的过程中修改其他中心的数据库的状态为交易完成初始化。

### 回写数据-Oprdispatcher

工具Oprdispatcher有两种配置方式：从数据库抽取数据同步到主中心、从流文件抽取数据同步到主中心；

**方式一：从数据库抽取数据同步到主中心。**

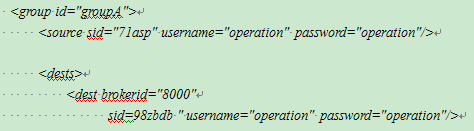
工具oprdispatcher从非主中心的交易数据库盘后把operation下面的5张表插入到对应的主交易中心对应的主席系统operation下面的5张表。

抽取的非主中心的Operation下面的5张表为：t\_BrokerUserEvent、t\_BrokerUserOtpParam、t\_BrokerUserPassword、t\_SettlementInfoConfirm、t\_TradingAccountPassword。

导入的主中心的Operation下面的5张表为：t\_ImpBrokerUserEvent、t\_BrokerUserOtpParam、t\_BrokerUserPassword、t\_SettlementInfoConfirm、t\_TradingAccountPassword。

最后主中心的historysettlement schema中的t\_brokeruserevent包含的数据是主中心的operation.t\_brokeruserevent和operation.t\_impbrokeruserevent。

数据源和目的的配置如下：



**方式二：从流文件抽取数据同步到主中心。**

目前新版本的Oprdispatcher支持从流文件里面导出CSV。同步到主中心的表中除了以前的五张表外，加了t\_tmdborder和t\_tmdbtrade以及t\_loginforbiddenuser。值得注意的是，从其他中心往主中心导入数据的动作需要在切结算之前完成。

需要注意的是，用户事件从交易中剥离开来，所以需要到风控中的流水读取用户事件，借助oprdispatcher写到主中心的数据库中（historysettlement.t\_brokeruserevent）。因为这个动作直接将用户事件导入了historysettlement schema中，所以不需要在切结算状态之前做。同时，非主中心的的系统用户事件，也需要部署相应的风控系统才能看到。如果没有独立的数据库，风控系统也可以CSV方式拉起。

Riskgate –-csv，注意这里的启动方式需要两个“-”

风控系统初始化数据可用unldr工具导出，类似于tinit导出perf文件的过程，只不过导出的是RISK schema下的数据，相应需要调用unldr的使用Risk.ctl。

### 风控系统-riskdrmt

（1）风控的riskdrmt同步的风控数据包括：

1. 自定义强平标准（设置-> 选项-> 投资者资金信息 -> 使用风险度自定义强平）；

2． 业务通知模板设置；

3． 业务通知；

4． 风险通知模板设置，包括自动短信发送时间，风险通知警示值，是否允许重复发送选项；

5． 风险通知；

6． 净持仓保证金指标

7． 风控用户操作事件记录，目前只有用户导出数据操作

8． 强平留痕数据

交易数据已经由交易系统同步，riskgate转发给风控系统。

（2）风控riskdrmt的工作原理和产生的流水

1. riskdrmt连接drmtsvr，发送登录请求，请求中包含目标drmtsvr所处中心号，及所有其它中心号。
2. riskfront处理登录请求，检查请求中的目标drmtsvr所处中心号是否和自己的中心号匹配，并发送登录应答到riskdrmt。

若登录成功，riskfront向riskdrmt发送订阅同步数据的请求。订阅请求中包括所有其它中心号。

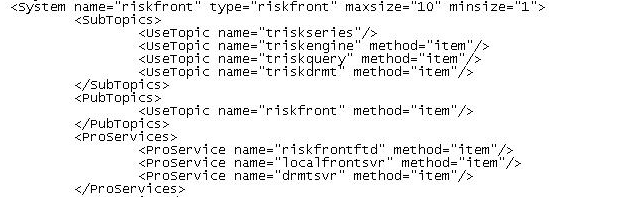
1. riskdrmt收到登录应答，若登录成功，向riskfront发送订阅同步数据请求，订阅请求中只包括一个目的中心号。
2. riskfront收到riskdrmt订阅同步数据的请求，检查请求中的目的中心号是否与自己所处中心号一致，若一致，开始向riskdrmt发送同步数据报文，报文中带有DRSysID。
3. riskdrmt收到riskfront发送来的同步数据报文，将报文根据DRSysID分类保存。
4. riskdrmt收到riskfront订阅同步数据的请求，从接收到的报文中选择DRSysID和订阅请求中DRSysID相同的报文，发送到riskfront。

7. riskfront收到riskdrmt发来的同步报文。

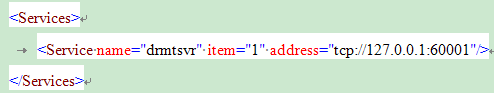
（3）全局配置文件的变化

每个中心分配唯一的风控中心号，即riskgate.ini中DRSysID。系统启动后，riskgate将DRSysID通过总线发送到riskfront，riskfront就知道自己所处的中心号了。

Systemconfig.xml的增加



DeployConfig.xml中配置riskfront提供给riskdrmt来连接的服务器地址



1. riskdrmt的ini的重要配置项



## 多活架构的数据同步流程

假定一套多活交易中心包括：

主中心（交易中心1）、备中心（交易中心2）、分交易中心3、分交易中心4、分交易中心5；中心号分别是1、2、3、4、5。

### 日常盘后数据同步操作流程

在多活交易中心系统中，开户、结算、出入金、费率调整等一系列操作均在主交易中心的主用系统进行，交易以及与交易密切相关的操作，例如交易终端登录密码修改等操作，则具体投资者在哪个交易系统进行交易，该操作则在哪个交易系统进行，盘后再统一汇总给主交易中心的主用系统。当然对于密码修改，建议终端登录到主中心进行操作，然后实时同步到备中心。

以往期货公司存在多套交易系统运行的时候，业务人员往往工作量较大，例如通常需要在多套交易系统分别调整保证金比率、手续费率等，然后分别进行结算再进行对账，综合交易平台的多活交易中心实现了上述操作在主交易中心的主用系统完成之后，系统可以方便将这些信息备份给次用中心以及备交易中心的主用系统。

具体多活交易中心日常盘后操作流程如下图所示：

**主用系统至次用系统**

当主交易中心的主用系统结算完成后，期货公司业务人员可在次用系统的管理平台（即结算平台）进行操作，将主用系统中所有在次用中心交易投资者的相关信息抓取至次用中心，这些信息主要是主用系统结算后的数据，可以方便复制到次用系统，次用系不用再结算。

**主交易中心的主用系统至备交易中心的主用系统（主要指结算数据库）**

主交易中心结算系统结算完成后，业务人员先通过结算管理平台将系统状态切换到不活跃状态后，技术人员可将主中心结算系统数据库数据进行增量备份，然后在备中心进行增量恢复。用于交易中心发生灾难时用于发起切换申请动作以及必要时接管结算用。如果采用了专业的第三方数据库实时同步工具，则不必盘后备份到备中心。

注意备交易中心交易系统的初始化采用文件方式，而文件是由主交易中心结算系统所生成。

**备交易中心的主用系统至主交易中心的主用系统（主要指交易数据库）**

备交易中心在盘中会通过tmdb组件回写Operation信息，这些信息主要包括交易用户事件、投资者登录密码修改等与交易行为密切相关的信息，这些信息一般由投资者通过交易终端的操作来更新，而不是经纪公司业务人员操作或者是在交易所有留痕的信息，所以往往都是负责交易的交易系统掌握。因此，这些信息在交易结束之后，由运维人员执行Operation Dispatcher程序，将备交易中心的这些数据导回到主交易中心主用系统中。

### 日常盘中数据同步操作流程



如上图所示：

日常盘中操作均在主交易中心的主用系统完成，例如开销户、出入金等。

盘中如果有需要实时上场（即通知交易核心）的内容，则在主交易中心的主用系统进行数据操作，主用系统交易系统中的drmt进程会负责向本主用系统下属所有次用系统交易系统以及备中心对应的主用系统交易系统的Front进程转发TradeSeries（即上图所示的TSeries），该TradeSeries取至主中心的主用系统交易系统中，这样保证一个公司下的所有主用系统、次用系统、备份主用系统交易系统数据一致。

备中心交易系统的tmdb组件平时启动，负责回写Operation数据到备中心对应的交易数据库中。当然，主交易中心的主用系统平时也会启动tmdb进程，用于回写交易数据至各自交易数据库。

备中心交易系统的dbmt组件启动是当灾难发生时，需要数据库通知交易核心接管交易。

## 综合交易平台FENS机制的原理

FENS即是Front End Name Server：前端名字服务程序。FENS为交易API提供前置服务地址列表。

### 综合交易平台FENS跟交易API配合工作

终端连接FENS有两种模式:银期模式和交易模式。终端以银期方式连接时，FENS返回银期所在中心的地址组（如果fens.xml中的spectransferid有指定，返回该指定中心的地址组；如没有指定，交易核心根据银期报盘的心跳，判断银期报盘所在中心）；终端以交易方式连接时，FENS返回投资者或操作员交易权限所在中心的地址组。这样就要求银期报盘和DRMT必须在同一个中心，如果灾备发生切换也需要一起切换到备中心。

当投资者或操作员的交易权限发生变化时，FENS根据变化后的权限中心返回地址。

在交易API的头文件中，有如下定义：

/////////////////////////////////////////////////////////////////////////

///TFtdcLoginModeType是一个登录模式类型

/////////////////////////////////////////////////////////////////////////

///交易

#define THOST\_FTDC\_LM\_Trade '0'

///转账

#define THOST\_FTDC\_LM\_Transfer '1'

typedef char TThostFtdcLoginModeType;

#endif

///Fens用户信息

struct CThostFtdcFensUserInfoField

{

///经纪公司代码

TThostFtdcBrokerIDType BrokerID;

///用户代码

TThostFtdcUserIDType UserID;

///登录模式

TThostFtdcLoginModeType LoginMode;

};

///注册名字服务器用户信息

///@param pFensUserInfo：用户信息。

virtual void RegisterFensUserInfo(CThostFtdcFensUserInfoField \* pFensUserInfo) = 0;

交易终端开发者在调用“注册名字服务器用户信息时”，可以指定登录模式为“交易”或者“银期”，FENS会返回相应的地址组。

FENS配置文件示例如下：

**Fens.xml**

<fens spectransferid=""> <!—spectransferid为指定银期所在中心号-->

<frontgroup name="trade-tcp" listenIP="0.0.0.0" listenport="5901" channeltype="tcp">

<frontlist name="PD\_1" match="172.16.0.0" mask="255.0.0.0">

<systeminfo name="center\_1" sysid="1"> <!—sysid为中心号（identidyid）-->

<frontend address="172.16.188.1" port="1"/>

<frontend address="172.16.127.1" port="44405"/>

</systeminfo>

<systeminfo name="center\_2" sysid="2">

<frontend address="172.16.188.3" port="1"/>

<frontend address="172.16.126.68" port="16105"/>

</systeminfo>

<systeminfo name="center\_3" sysid="3">

<frontend address="172.16.188.2" port="1"/>

<frontend address="172.16.126.50" port="16205"/>

</systeminfo>

</frontlist>

<frontlist name="All"> <!—缺省的地址组-->

<systeminfo name="center\_1" sysid="1">

<frontend address="172.16.188.3" port="6"/>

</systeminfo>

<systeminfo name="center\_2" sysid="2">

<frontend address="172.16.188.4" port="7"/>

</systeminfo>

<systeminfo name="center\_3" sysid="3">

<frontend address="172.16.188.4" port="8"/>

</systeminfo>

</frontlist>

</frontgroup>

</fens>

### 正常登录时工作流程

投资者或操作员与交易中心的对应关系是在系统初始化的时候同步到交易系统，各个交易中心分别使用本地的全量CSV初始化本地交易中心。所有的FENS服务器均侦听本地排队机的输出流水，于是所有的FENS服务器均知道投资者或操作员与所属交易中心的对应关系。

**强调一下：由于FENS侦听本地排队机输出流，为了保证所有的FENS服务器均知道所有的投资者和操作员与交易中心的对应关系，FENS需要部署在使用全量CSV文件初始化的交易中心。同一个交易中心可以部署多台FESN服务器。**

当投资者登录FENS服务器的时候，FENS通过交易API向投资者返回投资者所属交易中心的地址组。然后投资者登录进去相应的交易中心进行报单和查询操作。



### 投资者切换交易中心时工作流程

当投资者或操作员与交易中心的对应关系发生变化时，即业务人员在主交易中心的FLEX柜台界面上进行投资者或操作员与交易中心的对应关系变更时，该信息实时上场，主中心的FENS从排队机的输出流水中侦听到该信息，该信息同时被实时同步到各个分交易中心，各个分交易中心的FENS侦听分交易中心的排队机输出流水，进而得知变化后的交易中心。投资者终端登录FENS时，FENS根据变化后的权限中心返回地址。

## 多活交易中心的启动

综合交易平台是基于全内存的交易系统，目前线上版本已经支持7\*24小时连续交易。

7\*24小时全内存连续交易极大方便了用户运维人员，不必每日启停系统。这样的连续交易系统，在切换交易日时，运维人员仅需点击“重新初始化”，将当前内存中的数据重新初始化，即可完成下一交易日的所有交易准备工作，极大简化运维操作流程，节省运维时间，减少人为误操作。

 **7\*24小时连续交易系统运行图示**

该特性使得建有CTP多活交易中心的用户，其运维人员可以采用“一键运维”来实现初始化主中心，同时该初始化指令也会通过远程同步组件实时同步到各个分交易中心，各个分交易中心接到指令后也会立即完成初始化操作。

 “**一键运维”多中心图示**

## 投资者、操作员与交易中心的对应关系

投资者的合约权限和登录权限都是从主中心同步到各个分中心的（各个分中心初始化文件都是从主中心导过去的），所以投资者的合约权限和登录权限在所有的交易中心都是一样的。

## 多活切换流程

先简单描述一下正常启动情况：

主中心（交易中心1）数据库启动；

备中心（交易中心2）和另外3个分交易中心使用csv启动；

备中心、交易中心3使用全量csv，交易中心4和交易中心5使用过滤后的csv；

主中心上启用drmt，实时转发报文至备中心和另外3个分交易中心；



主中心drmt配置如下：

|  |
| --- |
| 上面公共配置省略。  #0: 二席 其他: 备中心  #这里的二席和备中心含义和平时理解的不太一样，二席是指公司brokerid和主席不一致，盘后只导入结算结果到二席，而备中心都是通过增量备份恢复当日结算交易风控等全部数据。在自运维多交易中心系统中，这里的二席已不会出现。所以下列配置中通常都填1。  Front=0000\_admin:1@tcp://172.16.126.68:15205 #假设备中心（交易中心2）前置  Front=0000\_admin:1@tcp://172.16.126.69:16405 #假设交易中心3前置  Front=0000\_admin:1@tcp://172.16.126.70:17205 #假设交易中心4前置  Front=0000\_admin:1@tcp://172.16.126.71:18205 #假设交易中心5前置  #启动方式：yes表示灾后启动备中心，no表示正常启动主中心  Recovery=no  #带流启动方式：yes表示流中报文需处理，no表示流中报文不处理，且在Recovery=yes时有效  Redo=yes |

### 场景一（所有交易中心工作正常，部分投资者在主中心交易和转账）



这是最常见的工作场景，所有交易中心正常工作，在主中心进行交易的投资者，交易终端连接到FENS服务器后，获取对应的交易中心前置列表（即主中心前置地址列表），也从FENS获知银期转账系统处于主中心，交易终端的交易报单和转账均发向主中心的前置。

### 场景二（所有交易中心工作正常，部分投资者在备中心交易）



这种情况下，也就是说这些投资者仅在备中心具有交易权限。当这些投资者登录到FENS时，获知有交易权限的交易中心号（即备中心）对应的前置地址列表，然后登录到备中心的前置进行报单等交易。同时从FENS获知银期转账系统所在的交易中心号（即主中心），当需要进行银期转账时，该客户端向主中心的前置发送转账请求。

也就是说，客户端以交易接入模式从FENS获知对应的交易中心前置地址；以银期接入模式从FENS获知对应的银期系统所在的交易中心前置地址，进而得以实现在备中心进行交易，而从主中心进行银期转账的目的。在主中心进行银期转账后，出入金信息实时同步到备中心。

对于投资者来说，不管是交易还是银期转账，都是在同一个终端上进行，并不知道背后的操作，银期转账后马上可以在终端看到资金的变化。

### 场景三（所有交易中心工作正常，从主中心切换部分投资者到备中心）

 盘中部分投资者有从主中心切换到某个备中心进行交易的需求，在CTP多活模式下，业务人员从柜台进行操作，只需将投资者对应的交易中心从主中心变更为备中心即可。该对应关系上场后，FENS从排队机的输出流中侦听到该信息。同时该信息也被DRMT同步到各个分中心，各个分中心的FENS也获得这些投资者与新的交易中心的对应关系。

投资者的交易终端重新登录即可自动连接到对应的备中心的前置。也可以通过FENS连接到主中心的前置进行银期转账，资金信息被实时同步到各个分中心。

### 场景四（所有交易中心工作正常，从备中心切换部分投资者到主中心）

 盘中部分投资者有从某个备中心切换到主中心进行交易的需求，在CTP多活模式下，业务人员从柜台进行操作，只需将投资者对应的交易中心从备中心变更为主中心即可。该对应关系上场后，FENS从排队机的输出流中侦听到该信息。同时该信息也被DRMT同步到各个分中心，各个分中心的FENS也获得这些投资者与新的交易中心的对应关系。

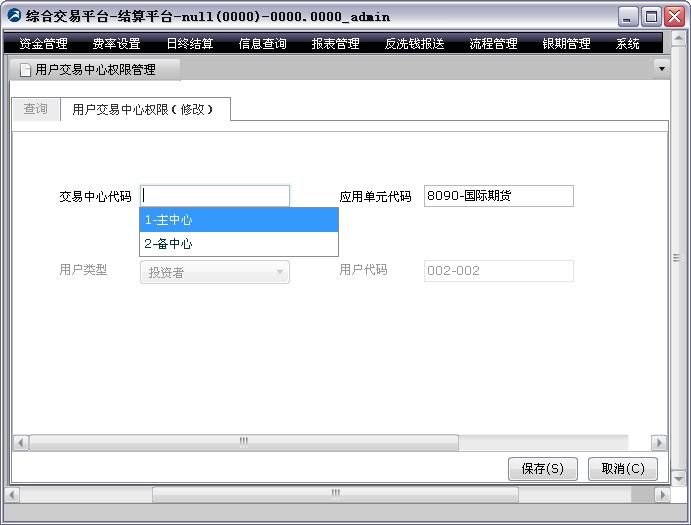
投资者的交易终端重新登录即可自动连接到对应的中心（即主中心）的前置。同时通过FENS连接到主中心的前置进行银期转账，资金信息被实时同步到各个分中心。

### 场景五（主中心正常，某个备交易中心宕掉，主中心接管该备中心）



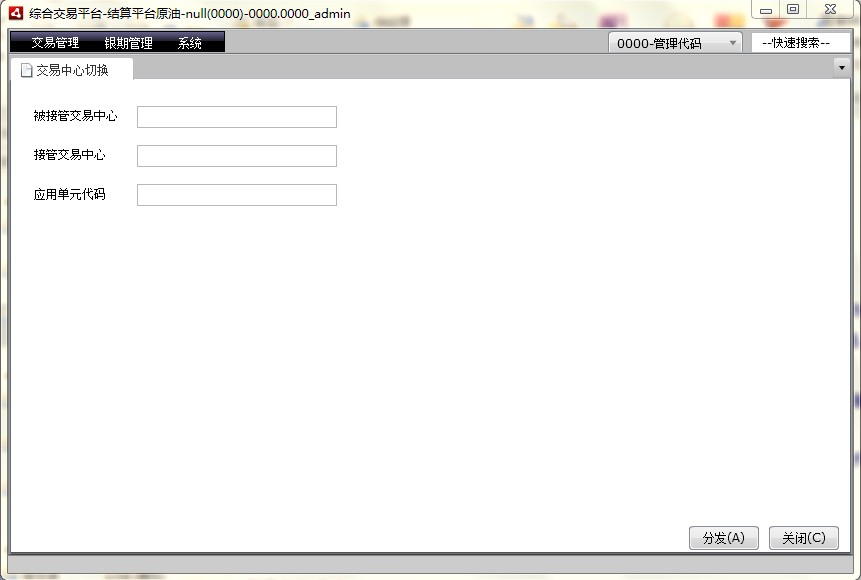
某个分交易中心（除主中心外的其他交易中心）出现故障，主中心接管该分交易中心。此时可以将该分交易中心的交易权限切换至主中心。

1. 若客户数很少，比如只有5人以下，可以通过经纪公司系统管理员在结算柜台“多交易中心管理🡪用户交易中心权限管理”来切换客户交易中心权限，这里只可以对单个客户进行切换操作。如下图所示：



注：“交易中心代码”输入框因不是下拉框，所以需要先删除原有数据，才会有匹配数据显示。

1. 若客户数较多，则不适合使用“用户交易中心权限管理”来操作，则需要使用CTP系统管理员在结算柜台“多交易中心管理🡪交易中心切换”菜单操作。如下图所示：



交易中心切换指令ReqInsDRTransfer（OrigDRIdentityID，DestDRIdentityID，OrigBrokerID，DestBrokerID）。当drmt的配置Recovery=no时，表示主中心正常运行的drmt，是会同步交易中心切换指令ReqInsDRTransfer到其他交易中心的。

本场景就是交易中心2的交易权限都被切换到了主中心上，同时被drmt转发到其他各个分交易中心，这样交易中心2的投资者只能在主中心上进行交易，不能在原先的交易中心2上继续交易了。如下图所示：

 在主中心的柜台操作接管备中心的投资者，投资者与交易中心的对应关系上场后，FENS从排队机的输出流中侦听到该信息。同时该信息也被DRMT同步到各个分中心，各个分中心的FENS也获得这些投资者与新的交易中心的对应关系。

投资者的交易终端重新登录即可自动连接到对应的中心（即主中心）的前置。同时通过FENS连接到主中心的前置进行银期转账，资金信息被实时同步到各个分中心。

### 场景六（主中心宕掉，备中心接管主中心所有投资者，备中心变为主中心）

**主中心交易和结算均出现故障，需切换主中心的交易和结算到备中心**

主中心发生故障，交易和结算均不能正常进行，此时可以将主中心的所有业务切换至备中心，即进行灾备切换。



**操作步骤：**

1. 停止主中心drmt
2. 启动备中心的dbmt，（需要确认当前sync.t\_tbcommand的最大commandno，使用带commandno方式启动）
3. 启动备中心的web
4. 启动备中心的drmt，（除主中心外其他交易中心的drmt.ini配置Recovery=yes,Redo=yes，以恢复模式启动drmt）

因为备中心第一次启动drmt，原来drmt下面是没有收过tseries流水，所以，drmt启动后开始收流。由于Recovery=yes,Redo=yes，所以在收到交易中心切换指令ReqInsDRTransfer前，是不会转发任何报文出去。

1. 使用CTP系统管理员在结算柜台“多交易中心管理🡪交易中心切换”菜单操作，让备中心接管主中心。

由于Recovery=yes，Redo=yes，这时drmt收到ReqInsDRTransfer报文后，检查其DestDRIdentityID是否为本系统中心号，如果一致，则开始转发；如果不一致，则不开始转发。这里我们是在主中心上将主中心切换到备中心，所以DestDRIdentityID=2，即备中心，则备中心drmt开始转发后续报文。

1. 所有业务操作开始在备中心（此时的备中心已经升级为主中心）上操作，银期备中心冷备启用。FENS侦听到银期系统所在的当前中心，原先在主中心的投资者交易终端重新登录即可与备中心建立连接进行交易和转账操作。



### 场景七（某个分交易中心出现故障，备交易中心接管出现故障的分交易中心）



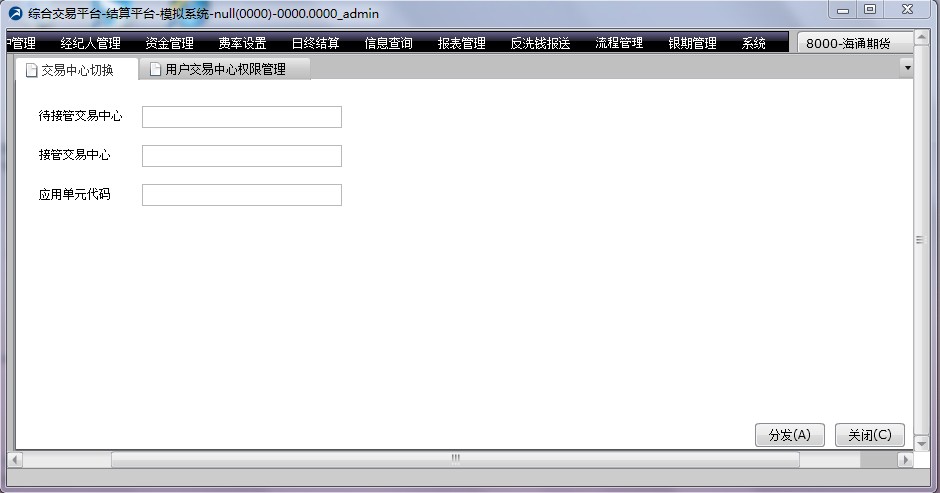
**交易中心4出现故障，备中心（交易中心2）接管交易中心4。**

在正常情况运行一段时间后，**交易中心4**发生故障，交易不能正常进行，特需时，可以将**交易中心4**的交易权限切换至备中心**（交易中心2）**，有两种方式可以操作。

1. 启动备中心的dbmt，（需要确认当前sync.t\_tbcommand的最大commandno，使用带commandno方式启动）；

启动备中心的web；

使用CTP系统管理员在结算柜台“多交易中心管理🡪交易中心切换”菜单操作，让备中心接管**交易中心4**；



1. 使用操作员用户登录CTP监控终端ThostMan2，在“应急指令🡪切换交易中心”菜单操作。如下图所示：



非主中心的交易中心切换到其他的非主中心的交易中心，在正常情况下是不建议如此操作的。但部分客户有特殊需求时，例如需要切换过来的客户都是高频客户，担心影响主系统时，可以使用手工应急方式切换到备中心或其他交易中心。但是这类切换只能自行保存手工切换日志或截图。只有进行切换操作的对应中心交易流水中有留痕。

### 场景八（主中心接管某个分中心，备中心又接管主中心）

**交易中心4（除主中心外的其他交易中心）出现故障，主中心接管交易中心4；然后又过了一段时间后，主中心交易和结算均出现故障，需切换主中心的交易和结算到备中心**

主中心接管交易中心4同场景五（主中心正常，某个备交易中心宕掉，主中心接管该备中心）。

  **主中心接管交易中心4**

备中心接管主中心同场景六（即主中心宕掉，备中心接管主中心所有投资者，备中心变为主中心），但有如下细节需要注意：

1. 主中心接管交易中心4，交易中心切换指令ReqInsDRTransfer（4,1,1200,1200）通过drmt已经同步给备中心了，即备中心的tseries流水中已有一个ReqInsDRTransfer报文。
2. 备中心需要接管主中心时，按照场景六的操作顺序，先启动备中心的drmt，由于备中心drmt是首次运行，且Recovery=yes,Redo=yes，这时drmt收到ReqInsDRTransfer报文后，检查其DestDRIdentityID是否为本系统中心号，如果一致，则开始转发，如果不一致，则不开始转发。由于备中心tseries流水中已有一个ReqInsDRTransfer（4,1,1200,1200），所以需要先判断DestDRIdentityID是否为2，即备中心，这里第一个ReqInsDRTransfer并不是切换到备中心上，所以后面的流水仍然不转发。只有到我们在备中心结算柜台上将主中心切换到备中心，drmt收到第二个ReqInsDRTransfer时，由于DestDRIdentityID=2，即备中心，备中心drmt才开始转发后续报文。



### 场景九（主中心接管某个分中心，备中心接管另一个分中心，备中心又接管主中心）

**交易中心3（除主中心外的其他交易中心）出现故障，主中心接管交易中心3；然后过了一段时间后，交易中心4出现故障，备中心接管交易中心4；再然后，主中心（交易中心1）交易和结算均出现故障，需切换主中心（交易中心1）的交易和结算到备中心（交易中心2）**

主中心接管交易中心3同场景五。

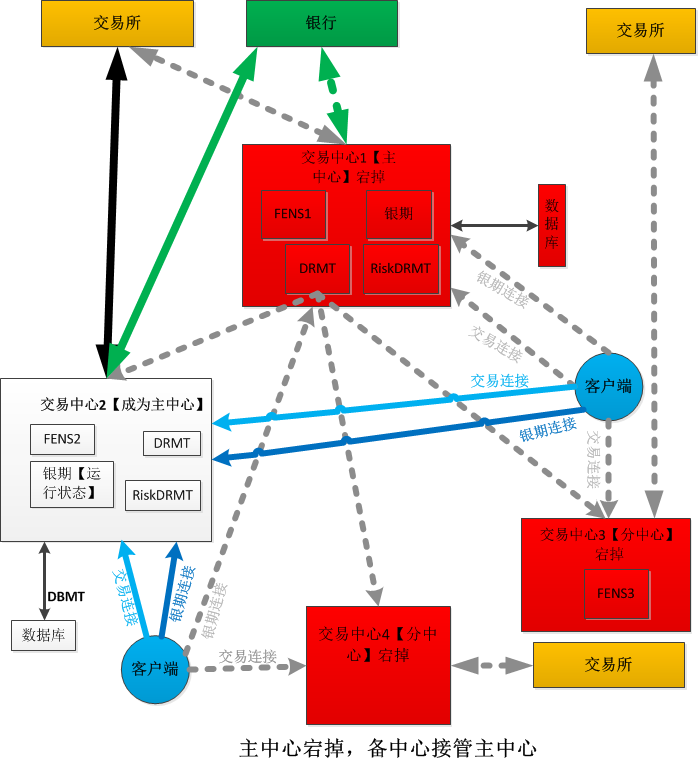
 交易中心3上的全部投资者被主中心接管。原交易中心3的投资者客户端自动重连时会选择到主中心的交易前置，银期转账也会选择到主中心的前置。在主中心上进行接管操作。同时主中心向所有分中心发送交易中心3被主中心接管的报文。

备中心接管交易中心4同场景七。



备中心接管主中心与场景六有些区别，具体如下：

1. 备中心需要接管主中心时，按照场景六的操作顺序，先启动备中心的drmt，由于备中心drmt是首次运行，且Recovery=yes,Redo=yes，这时drmt收到的第一个切换报文ReqInsDRTransfer 是主中心接管交易中心3的ReqInsDRTransfer（3,1,1200,1200），所以DestDRIdentityID不为2，后面的流水不转发。第二个切换报文ReqInsDRTransfer是备中心接管交易中心4的ReqInsDRTransfer（4,2,1200,1200），由于DestDRIdentityID=2，备中心drmt才开始转发后续报文，但其实此时主中心drmt仍然是正常运行的，在第二个ReqInsDRTransfer（4,2,1200,1200）和第三个ReqInsDRTransfer（1,2,1200,1200）之间的流水会被重复处理，包含资金类报文。



所以这里备中心接管主中心起drmt时，我们需要配置drmt的redo=no，在起drmt前先把备中心的tseries流水拷贝到备中心drmt目录下，然后以带流重启的方式restart drmt，这样drmt便不再检查已有的流水中是否有ReqInsDRTransfer报文，即不会去看前两个ReqInsDRTransfer， 直接等重启后，接收后面的ReqInsDRTransfer（1,2,1200,1200）指令。

### 场景十（备中心接管某个非CTP系统）

一般情况下，用户具有多套交易系统，比如既有CTP主用系统，又有个性化的“X盛系统”。

当“X盛系统”宕掉时，CTP也可以接管该“X盛系统”。

前提条件：平时在CTP系统中将该“X盛系统”的所有投资者设置成与某个虚拟的交易中心对应，并提前下载好相应的CTP系统交易终端。

 比如提前在CTP系统中设置虚拟的交易中心9，该交易中心9中的投资者与未来被接管的“X盛系统”的投资者对应。



当“X盛系统”宕掉时，用户立即在CTP系统的主中心进行“交易中心2接管交易中心9”的操作，然后通知“X盛系统”的所有投资者使用预先下载好的CTP系统的交易终端进行登录和下单即可。这样就实现了用CTP的备中心接管了出现故障的“X盛系统”目标。