

黄金交易数据交换协议

（GTP，Gold Trade Protocol）

V1.01

上海黄金交易所

2024年5月

文档修订历史

| **版本号** | **日期** | **修订说明** |
| --- | --- | --- |
| V1.0 | 2023-11 | 形成基线版 |
| V1.01 | 2024-5 | 修订附录中GTP协议白皮书 |

目录

[1 范围 6](#_Toc25556)

[2 规范性引用文件 6](#_Toc859)

[3 术语和定义 6](#_Toc11880)

[4 应用环境 6](#_Toc8907)

[5 协议层次结构 7](#_Toc28082)

[6 体系结构 8](#_Toc15033)

[6.1 通讯模式 8](#_Toc6301)

[6.1.1 对话模式 9](#_Toc1405)

[6.1.2 私有模式 10](#_Toc27368)

[6.1.3 广播模式 10](#_Toc14197)

[6.2 数据流 10](#_Toc6293)

[6.2.1 数据流划分 10](#_Toc10574)

[7 会话机制 12](#_Toc391)

[7.1 连接断开机制 12](#_Toc9637)

[7.1.1 连接 12](#_Toc3871)

[7.1.2 登录认证 12](#_Toc31907)

[7.1.3 心跳机制 13](#_Toc29677)

[7.1.4 退出/断开 13](#_Toc21888)

[7.2 消息序列类别号和消息序号 13](#_Toc1445)

[7.2.1 消息序列类别号 14](#_Toc13582)

[7.2.2 消息序号 14](#_Toc27638)

[7.3 消息连续性 14](#_Toc10690)

[7.4 消息持久化 15](#_Toc29441)

[7.5 消息断点 15](#_Toc20688)

[7.6 消息恢复机制 15](#_Toc1649)

[7.6.1 消息重传方式 15](#_Toc17746)

[7.6.2 可靠消息恢复 16](#_Toc17041)

[7.6.3 不可靠消息处理 16](#_Toc20892)

[7.6.4 应用示例 17](#_Toc13009)

[7.7 消息安全机制 18](#_Toc15394)

[7.8 数据完整性校验 18](#_Toc20878)

[8 消息格式 18](#_Toc5278)

[8.1 消息语法 18](#_Toc15141)

[8.2 域定义 18](#_Toc19287)

[8.2.1 域的使用 18](#_Toc22169)

[8.2.2 域汉字编码 19](#_Toc27580)

[8.3 数据类型 19](#_Toc2123)

[8.3.1 基本数据类型 19](#_Toc23723)

[8.3.2 整数（int） 19](#_Toc26519)

[8.3.3 十进制小数（Decimal） 19](#_Toc23580)

[8.3.4 扩展数据类型 21](#_Toc30304)

[9 消息定义 21](#_Toc22799)

[9.1 消息结构 21](#_Toc4543)

[9.2 会话报头 22](#_Toc9702)

[9.3 特征信息 22](#_Toc19834)

[9.3.1 基本报头 23](#_Toc3732)

[9.3.2 扩充报头 23](#_Toc14321)

[9.4 会话消息 24](#_Toc31051)

[9.5 应用消息头 25](#_Toc12954)

[9.6 应用消息类型 26](#_Toc11659)

[9.6.1 交易前消息 26](#_Toc11082)

[9.6.2 交易中消息 27](#_Toc31942)

[9.6.3 交易后消息 28](#_Toc9436)

[9.6.4 管理消息 29](#_Toc8427)

[10 应用层消息重发机制 29](#_Toc12789)

[10.1.1 消息重发机制 29](#_Toc2209)

[10.1.2 应用示例：报单/撤单消息恢复 31](#_Toc23281)

[10.1.3 应用示例：ETF交易消息恢复 32](#_Toc17908)

[11 附录 32](#_Toc28740)

# 范围

本标准制定了上海黄金交易与市场参与者之间进行黄金交易所需的数据交换协议（Gold Trade exchange Protocol，简称GTP），规定了消息类型、报文语法、域定义、会话机制、扩展方式等内容。

本标准适应于上海黄金交易所交易系统与会员二级系统、红马甲及其他外部对等单位之间进行业务数据交换。

# 规范性引用文件

GBK 汉字内码扩展规范

GB/T 2659 世界各国和地区名称代码

GB/T 12406 [表示货币和资金的代码](http://www.baidu.com/link?url=JY5cPnB_Xp8OS-YFCyf9dZGqq8HF6uz1_iWOZjZvFw5HC3uXr1CBCiQHsd9AasnDVCYoIOUS_1z5S6sWChpCna)

# 术语和定义

**GTP协议**：全称Gold Trade exchange Protocol，黄金交易数据交换协议。

**交易参与方**：参与上海黄金交易所交易的会员或其他机构。

**一级系统**：是指上海黄金交易所的交易系统。

**接入方系统**：接入方为交易参与方，接入方系统通常指二级系统，也包括基金公司系统、交易客户端(红马甲)等其他接入方使用的系统。

**二级系统**：是交易参与方单位根据交易所接口自主研发或者从第三方系统开发商购买的满足交易所系统协议接口标准的二级交易系统。

# 应用环境

GTP协议应用于交易所交易系统（一级系统）与接入方系统之间。



在接口提供方式，接入方系统可通过交易所提供的API程序包，调用接口功能与交易所系统建立连接，实现数据交互。待条件成熟后，交易所还将提供GTP协议和相关接口规范文档，交易参与方可自行选择API程序包或接口协议文档进行开发。

# 协议层次结构

GTP协议底层基于TCP/IP基础协议栈。

GTP协议定义的标准和规范主要集中在会话层、表示层和应用层。其中：



1）在会话层，GTP协议定义了链路管理、数据封装、数据加解密、数据压缩、数据校验、数据恢复等标准规范。

2）在表示层，GTP协议定义了黄金交易指令及相应交互数据的数据类型、语法格式、消息结构等标准规范。

3）在应用层，GTP协议针对具体业务场景设计了相应的消息类型，外部交接入方通过这些类型的消息接口调用交易所提供的服务。

# 体系结构

在GTP协议中，对于通讯双方消息交互，采取了**分类处理**的原则。分类处理后的消息被定义为不同的数据流来表示。

对于数据流的协同工作方式和管理原则，则通过通讯模式来定义。每种数据流均对应一个通讯模式，一个通讯模式下可建立多种数据流。

## 通讯模式

通讯模式定义了通讯双方之间的**协同工作的方式**。不同通讯模式有着不同的**通讯流管理原则**。



### 对话模式



由交易参与方主动发起的通讯请求，交易所端接收和处理请求，并给予异步响应。如：报单，查询等。

在对话模式下，一个数据流对应于一个连接，仅在这个连接内保障消息的完整性和有序性。连接断开后将创新构建新的数据流，这个新的数据流和断开之前的原数据流没有直接关系。

### 私有模式



交易所端主动向指定交易参与方发出信息。如：报单回报、成交回报等。

在私有模式下，一个数据流可对应于一个交易日内的一个或多个连接。在交易日内连接断开后，会从上次断开连接的地方继续接收下去，也可强制指定从头开始。通常每个交易参与方都有自己私有模式的数据流，来接收私有消息。

### 广播模式



交易所端主动向市场中的所有交易参与方发送相同的信息。如：行情信息、公告等。

在广播模式下，数据流管理原则同私有模式。通常交易所每个市场都会有一个广播模式的数据流，来向全市场所有交易参与方广播消息。

## 数据流

数据流表示一个**单向或双向、连续、无重复和遗漏**的数据报文的序列，它可以完成特定的功能[[1]](#footnote-1)。

### 数据流划分

考虑到并不是所有的消息都需要可靠送达，也并非所有消息都可以公开。按照**通讯模式、消息发送方向、可靠送达要求**（如：消息漏传后是否要求重发）、**公开程度、指令类型**等维度，可将消息归类提炼为多个数据流。

一种典型的数据流划分方式如下表：

| **通讯模式** | **数据流** | **发送方向** | **可靠性** | **可见范围** | **消息覆盖范畴** |
| --- | --- | --- | --- | --- | --- |
| 对话模式 | **交易流** | 接入方🡨🡪交易所 | **否** | 该接入方 | 管理消息  交易指令 |
| **查询流** | 接入方🡨🡪交易所 | **否** | 该接入方 | 查询指令 |
| 私有模式 | **回报流** | 交易所 🡪 多个接入方(通常为指定席位下所有登录的交易员) | 是 | 多个接入方 | 回报消息 |
| 广播模式 | **公共流** | 交易所 🡪 所有接入方 | 是 | 全市场 | 市场状态变化消息、公告信息、行情消息等 |
| **行情流** | 交易所 🡪 所有接入方 | 是 | 全市场 | 行情消息 |

#### 对话模式数据流

在对话通讯模式下，划分出**交易流和查询流**。交易流负责处理竞价交易指令（如报单、撤单）、ETF交易指令、保证金仓储交易指令，查询流单独处理查询指令（如：报单查询、成交单查询）。

交易流和查询流均为**不可靠**的数据流。交易系统不维护交易流的状态，当网络故障时，交易数据流会重置，通讯途中的数据可能会丢失。当出现消息丢失时，由应用层进行处理。详见“应用层消息重发机制”章节。

#### 私有模式数据流

在私有通讯模式下，为**回报流**。该流为单向数据流，通常由交易所系统发向交易参与方系统，传送的是私有消息，如回报消息、报价消息等。回报流的消息对于指定席位下的所有已登录交易员可见。

私有模式的数据流通常为**可靠**的数据流。以竞价交易系统为例，系统会为每个席位维护一个回报流。在一个交易日内，交易参与方系统断线后恢复连接时，可以请求交易系统发送指定序号之后的私有消息。

#### 广播模式数据流

在广播模式下，划分出**公共流和行情流**。这两种流传输的消息通常对全市场可见，由交易所对外下发。公共流传送的是公告、市场、合约等基础信息；行情流则专门传输行情信息。

公共流和行情流通常为**可靠**的消息流，交易所系统维护整个系统的公共流和行情流[[2]](#footnote-2)。在一个交易日内，交易参与方系统断线后恢复连接时，可以请求交易系统发送指定序号之后的广播消息。

# 会话机制

## 连接断开机制

接入方系统与交易所系统之间无论哪种通讯模式，都应先由客户端向服务器端先发起连接请求，在得到服务器端连接确认后，再由客户端发起身份验证请求，待得到服务器端身份认证通过的响应后，双方再基于以上某种通讯模式进行报文传送。当通讯双方长时间未有消息传送时，连接双方还会进行心跳监测。

### 连接

GTP协议传输层基于可靠的TCP长连接。一个连接上可以有多个数据流，提供多个数据流的信息和服务。

### 登录认证

接入方系统在与一级系统之间新建连接后，在正式交互消息之前，连接对应的**接入方需要先登录一级系统进行身份认证、生成会话密钥、上送断点信息等处理**。

1）**身份认证**：采用双因素认证方式，验证因素包括数字证书和接入方登录密码。如果验证失败，双方连接自动断开。

2）**生成会话密钥**：接入方与一级系统之间每次建立连接后协商一个会话密钥，用于后续双方加解密。如果协商密钥失败，双方连接自动断开。

3）**断点上传**：对于存在可靠数据流的一级系统，接入方可上送断点信息，用于双方通讯异常中断时，一级系统根据上送的断点信息从指定序号之后开始消息重传（详见“消息恢复机制”章节）。

### 心跳机制

心跳消息用于监控通信连接的状况。在连接已经建立的情况下，如果任何一侧在约定的时间内，没有向另一侧发送任何报文，则需要向对方发送心跳报文。

在消息交换空闲期间，按照如下规则互相发送心跳消息：

1）如果在约定时间内（发送超时时间，简称tsend）内没发送任何消息，则主动向对方发送心跳消息；

2）如果在规定时间内（接收超时时间，简称trecv）未收到对端的任何消息，则主动断开连接。

3）心跳的发送周期和检测周期，将由交易所决定，也可以考虑可以由客户端在登录时自行设置心跳时间确定（建议trecv设置成tsend的整数倍，如：trecv=120s、tsend=60s）。

### 退出/断开

接入方系统的某一连接断开前，通常需要先发起连接退出请求，待一级系统返回成功应答后，连接断开。连接断开后，其对应的接入方系统自动退出所有数据流。

## 消息序列类别号和消息序号

每条GTP消息中均带有一个消息序列类别号（SequenceSeries）和序列号（SequenceNo），对于可靠流的消息，消息收发双方可以根据消息序列类别号和序列号进行消息持久化处理和消息恢复。

### 消息序列类别号

消息序列类别号（SequenceSeries）用于唯一标识某一种数据流。生成规则如下：

1）对于细分后的不同数据流，采用不同的序列类别号。如下表所示：

|  |  |
| --- | --- |
| **数据流** | **消息序列类别号** |
| 交易流 | 1 |
| 回报流 | 2 |
| 公共流 | 4 |
| 查询流 | 5 |
| 行情流 | 6 |

2）对于数据流中的消息报文，无论上行还是下行，均采用同一个序列类别号。

### 消息序号

消息序号（SequenceNo）用于在数据流中唯一标识某一个消息。生成规则如下：

1）对于不同数据流中的消息，消息序号独立编号。

2）对于可靠数据流中的消息，如：回报、广播消息，消息序号从1开始，以“+1”方式逐渐递增。消息序号在交易日内连续编号，直到下一交易日后才重新开始编号

3）对于不可靠的数据流由于无需在本地持久化处理，也无需重传（详见“消息恢复机制”章节），序号约定取值为0。

4）会话层消息不占用消息序号。如：心跳消息。

## 消息连续性

在通讯正常前提下，对于可靠数据流，消息接收方的各数据流中的消息序号应按照“+1”方式严格递增。如果出现异常，相应处理如下：

1）如果新收到消息序号小于当前已收到的最大消息序号，则认为该消息为重复发送消息，则直接丢弃；

2）如果出现消息序号“跳号”的异常情况，则说明应用程序本身出现故障，将直接断开连接且不发送提示报文。**这主要是考虑到消息传输过程中基于TCP/IP进行可靠传输，不可能存在消息跳号的情况。**

## 消息持久化

对于消息发送方对外发出的消息，尤其是可靠的数据流，应在本地进行数据持久化处理，通过流文件[[3]](#footnote-3)等方式顺序逐笔记录其对外发送的消息。

通过数据持久化处理，既便于消息恢复时进行有序发送，又可以减少对数据库的访问量，提高程序运行效率。

## 消息断点

对于可靠的数据流，消息接收方应记录从数据流中接收到的最后一条消息的消息序列信息，在GTP协议中也称之为断点信息。断点由消息序列类别号和消息序号组成。对于不同的数据流，消息接收方通常会记录多个断点信息。如：对于公共流和行情流，需要接入系统分别记录各流的断点信息。

在消息收发双方断开重连之后，消息接收方通过将断点信息发给消息发送方，消息发送方可根据断点信息快速定位到缺失的消息，并有针对性的重传消息，保证可靠数据流中消息的**完整性**。

## 消息恢复机制

### 消息重传方式

在GTP协议中，消息恢复机制支持按照如下三种方式：

* **重传（RESTART）**：从本交易日开始重传；
* **续传（RESUME）**：从上次收到的最大消息序号之后开始重传；
* **快照（QUICK）**：只传输最新的消息报文。

### 可靠消息恢复

对于可靠数据流，为确保消息的有序、完整传输，当因通讯异常等原因导致存在漏传消息情况时，消息接收方可**在接入方重新登录时**，将断点信息发给消息发送方，消息发送方根据断点信息，从本地流数据中有序读取出可靠流消息，逐一补传给消息接收方。

消息重传处理如下：

1）在通讯恢复后接入系统重新登录交易所前置，告知交易所各可靠数据流的断点信息。根据数据流恢复方式的不同，断点中上送的消息序号也不同：

* + 当为重传方式时，消息序号为0；
  + 当为续传方式时，消息序号为数据流中接收到的最大消息序号；
  + 当为快照方式时，消息序号为-1。

2）交易所对接入方进行登录验证后，根据各数据流断点信息中上送的消息序列类别号和消息序号，先通过消息序列类别号定位需要消息恢复的数据流，然后从流文件中有序读取消息序号之后的所有消息，逐一重传给二级系统。其中：

* + 当断点中消息序号为0时，重发当天所有对外发送的消息；
  + 当断点中消息序号为-1时，对于行情流，先传送当前行情快照，再传送登录后市场行情的内容；对于其他流，只发送登录后的内容。

### 不可靠消息处理

对于不可靠的数据流，消息不可恢复。根据数据流的不同，处理方式也不同。处理方式如下：

1）对于交易流，考虑到重传交易指令的时间已不同于原提交时间，此刻交易环境已发生变化，重新提交指令已不能满足交易发起方的初始业务需要，通常由让交易发起方决定是否需要再次重新发起交易请求。详见“应用消息重发机制”章节。

2）对于查询流，如果查询指令发送/应答超时，则由查询发起方直接再次发起查询指令即可。

### 应用示例



## 消息安全机制

为了保证链路安全和数据安全，避免双方纠纷，接入方系统和交易所前置之间的具体安全流程参见相关安全接入流程文档。

## 数据完整性校验

数据完整性主要通过消息体长度（字节数）来校验。

接收方收到消息后，如果发现按照接收消息的实际长度与消息中上送的值不同，则可认为消息不完整，直接断开连接。

# 消息格式

## 消息语法

任何消息都严格由多个“域号=值”的基本结构组成。这些基本结构之间用可见的域界定符‘,’分割。



在遵循以下规则的前提下，“域号=值”基本结构可以是任意的次序：

1）除在数组类型中外，其他情况下域不允许重复出现。

2）对于消息中的特殊字符（如：’=’、’{‘、’}’、’[‘、’]’、’,’等），需要做转义处理。

## 域定义

域号采用“**1位字母+2位数字**”的3位字符编号规则。

### 域的使用

域的使用有三种方式：M-必须的；O-可选的；C-条件限制选择（根据其他相关域的存在与否或取值来决定）。

### 域汉字编码

域取值为汉字时，统一采用GBK汉字编码标准。

在传输之前，当采用GBK编码后的字符中包含有’=’、’,’、’{‘、’}’、’[‘、’]’等特殊字符时，应先进行转义处理。

## 数据类型

### 基本数据类型

在“域号=值”的基本结构中，值的数据类型可以是整数int、十进制小数Decimal、字符串String、二进制数据类型Binary、哈希类型Hash和数组类型Array中的任意一种类型。

### 整数（int）

无逗号和小数位，由ASCII 码字符‘-’，‘0’至‘9’组成。除非特别声明，整数类型均有正负。

在GTP协议中，通常用Nx表示十进制整数，x代表整数最大位数（不包括正负号）。

### 十进制小数（Decimal）

含有可选的小数部分，由ASCII 码字符‘-’，‘0’至‘9’和‘.’组成。

在GTP协议中，通常用N(x,y)表示，x代表总位数，不包括小数点和正负号，y代表小数点后的最长位数。

#### 字符串（String）

区分字母大小写，自定义长度。

在GTP协议中，通常用C1表示单个字符。通常用Cx表示字符串，x表示字符串的最大长度，除非特殊声明，字符串均可包含大小写字母，不允许出现不可见字符。

对于二进制数据的处理，传输前需先做BASE64转换成可见字符。

#### 哈希类型（Hash）

哈希类型是“域号=值”基本域块的集合，多个基本域块之间用’,’分隔。哈希以’{‘（左大括号）开始，’}’（右大括号）结束。

示例如下：

…DisseminationField={SeqSrsNo=1,SeqNo=101},…

**注释**：DisseminationField为断点信息，接入系统在登录时上送。示例中的域名在应用时会以域号代替。

#### 数组类型（Array）

数组类型是一组值的集合，多个值之间用’,’分隔。值（value）可以是字符串（string）、整数(int)、十进制小数（Decima）、哈希(hash)类型，也可以是数组（array）类型。数组以’[‘（左中括号）开始，’]’（右中括号）结束。支持多重嵌套。

示例如下：

…IPAddressListField=[

172.168.1.1,

172.168.1.2,

172.168.1.3

],…

**注释**：IPAddressListField为前置IP地址列表，由前置给接入系统下发。

…DisseminationData=[

{SeqSrsNo=1,SeqNo=101},

{SeqSrsNo=2,SeqNo=102},

{SeqSrsNo=3,SeqNo=103}

],…

**注释**：DisseminationField为断点信息，在登录时上送，通常上送有多个可靠数据流的断点信息,通过数组方式上送。示例中的域名在应用时会以域号代替。

### 扩展数据类型

**扩展数据类型在基础数据类型上定义而来。在应用过程中可根据实际情况灵活定义，自由扩展或裁剪。**

扩展定义的数据类型参考如下：

| **扩展类型** | | **单位** | **精度** | **类型** | **说明** |
| --- | --- | --- | --- | --- | --- |
| Amount | 金额 | 分 | 分 | N18 | 100表示1元 |
| Price | 价格 | 元/克  元/千克 | 0.000001元 | N(12,6) | 黄金/铂金/钯金：元/克  白银：元/千克 |
| Weight | 重量 | 千克 | 毫克 | N(12,6) | 0.000001表示1毫克 |
| Rate | 比率 | - | 百万分之一 | N(16,6) | 1表示100% |
| Quantity | 数量 | - | - | N12 |  |
| OKFlag | 是否标志 | - | - | C1 | 1-是，0-否 |
| Date | 日期 | - | - | C8 | YYYYMMDD |
| Time | 时间 | - | - | C8 | HH:MM:SS |

# 消息定义

## 消息结构

GTP应用消息由消息头和消息体构成。在消息传输之前，会先经GTP会话层进行数据封装。如下图所示：



其中：

1）对于封装后的GTP消息，有专门的会话报头。

2）应用消息头和消息体整体封装在报文体中传输。

3）应用层消息头和会话层报头均可以进行横向扩展。

## 会话报头

会话报头用来标识GTP消息的封装信息，如编码格式、消息长度、压缩信息、加解密信息等。

同时，通过对会话报头进行扩展，还可用来传输会话层的链路管理消息，如心跳、会话状态、时钟等。

## 特征信息

为便于GTP协议报文的解析和验证以及问题的排查，会话报头前引入如下特征信息：

·魔数（**magic number）**： 4个字节的整数，用于在会话层读取报文时进行验证，当读入流发生错误时，可在底层及时识别。

·特征值（**logID）**：4个字节的整数，便于日志的筛选和排查问题。一种典型用法如下：

1）对于请求发送方，每个请求消息发送前随机生成一个logID，该logID随同消息正文一起发给接收方。接收方对于与该请求对应的后续处理，通过logID进行识别。以报单请求：二级系统上送报单请求时，会随机生成一个logID发给交易所，交易所后续与该报单请求相对应后继消息（报单应答、报单回报、成交回报）都会带上该特征值。

2）对于心跳报文，log\_id的值为100000000；对于其他报文log\_id的值为9位数 范围为[100000002,999999999]。

·报文版本号：2个字节，便于快速排查因API版本号不一致导致的异常。

### 基本报头

GTP数据封装基本报头由6个字节组成，包含如下字段，定义如下：



其中：

* **报文类型**：标识报文体的类型，占1个字节，取值范围如下：

|  |  |  |
| --- | --- | --- |
| **报文类型** | **数值** | **描述** |
| GTPTypeNone | 0 | 此报文不具有任何意义，一般用于心跳 |
| GTPTypeString | 1 | 报文体是正常的域数据内容 |
| GTPTypeBinary | 2 | 报文体是二进制数据 |
| GTPTypeCompressed | 3 | 报文体是压缩后的数据，需要解压缩后再处理 |
| GTPTypeEncrypt | 4 | 报文体是加密后的数据，需要解密后再处理 |

* **扩充长度**：表示扩充报头的长度，占1个字节。如果取值为0，代表没有扩充报头，紧跟在报头后面的是封装报文体。
* **信息正文长度**：表示封装报文体的长度，占4个字节。如果取值为0，代表没有报文体，属于会话层的链路管理报文，如心跳消息。

**在基本报头，交易所系统可对报文类型进行扩展定义。当接收端收到非上述报文类型的报文时，可以将其丢弃，不做任何处理。**

### 扩充报头

扩充报头最长127个字节，包含如下字段：



其中：

* **标记类型、标记数据**：表示扩展报头的报文类型，标记数据根据标记类型的定义取相应的值。取值范围如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **标记类型** | **数值** | **长度** | **描述** |
| GTPTagNone | 0 | 0 | 丢弃不处理此标记 |
| GTPTagCompressMethod | 2 | 1 | 信息正文压缩方法 |
| GTPTagKeepAlive | 5 | 0 | 发送端发送心跳信息 |
| GTPTagDatetime | 1 | 4 | 时间戳。Unix格式时间，网络序 |
| GTPTagSessionState | 4 | 1 | 发送端状态 |
| GTPTagTradedate | 6 | 4 | 交易所当前交易日期。Unix格式时间，网络序 |
| GTPTagTarget | 3 | 2 | 说明报文需要发送的目标。如：告知交易所是要发到正式系统还是测试系统。 |
| GTPTagSetHeartbeatTimeout | 7 | 2 | 心跳超时时间设置 |

* **标记长度**：表示扩展报头的标记数据长度，最多125个字节。

**在扩充报头，交易所系统可以进行扩展定义，增加新的标记类型。接收端如果收到无法识别的标记，只需将其简单丢弃，不做其他处理。**

## 会话消息

会话消息由基础报头和扩充报头组成，没有报文体。

**对于不同类型的会话消息，通过扩充报头中的“标记类型”和“标记数据”进行扩展和自定义。**

GTP协议中定义的常见几种会话消息如下：

| **会话消息** | **标记类型** | **标记长度** | **标记数据** |
| --- | --- | --- | --- |
| 心跳 | GTPTagKeepAlive | **0** | 发送端发送心跳信息 |
| 心跳超时时间设置 | GTPTagSetHeartbeatTimeout | 2 | 发送端发送心跳超时时间 |
| 时间戳 | GTPTagDatetime | 4 | Unix格式时间，网络序 |
| 发送端状态 | GTPTagSessionState | 1 | 取值范围如下：  0-未知；1-未登录；2-已登录  （可根据需要进行扩展） |
| 发送交易日期 | GTPTagTradedate | 4 | Unix格式时间，网络序 |

## 应用消息头

应用消息头可用来表示数据流标识、报文链标识、消息长度、消息发送方信息等。定义如下表：

| **域名** | **数据类型** | **必需** | **说明** |
| --- | --- | --- | --- |
| BeginString | C8 | Y | 标识协议版本号，固定取值为GTP1.0 |
| ContentLength | Length | N | 除消息头之外，各field长度和，以字节为单位 |
| MsgType | C4 | Y | 消息类型，不同类型消息类型标识符不同 |
| SequenceSeriesNo | C1 | Y | 序列类别号，代表数据流的标号 |
| SequenceNo | N10 | Y | 消息序列号，基于数据流的编号 |
| ChainFlag | C1 | Y | 消息的连续标志，取值[[4]](#footnote-4)参考如下：  ‘S’-单个报文  ‘F’-报文链的第一个报文  ‘C’-报文链的中间报文  ‘L’-报文链的最后一个报文，无后续报文 |
| RootID | C8 | N | 作为消息来源标志，如有值，交易所原值返回 |
| SenderID | C8 | N | 发送方标识符 |
| ReceiverID | C8 | N | 接收方标识符 |

其中：

1）**BeginString**：标识消息的协议版本号，不同版本号消息的消息头可能存在差异。

2）**ChainFlag**：当报文长度超过最大报文长度时，长报文需要分割成多个报文发送。通过Chain标识可以用来识别收到的报文是被分割成多块的长报文的哪一部分。。

3）**SequenceSeriesNo** 和 **SequenceNo**: 用于保障通讯双方信息的完整性和有序性而定义的两个字段。具体用法参见“消息恢复机制”章节。

4）**MsgType**:用于标识消息类型，具体定义参见“应用消息”章节。

5）**SenderID**和**ReceiverID**：用于标识发送方和接收方。在交易所对外发送的通知报文中可为空。

## 应用消息类型

GTP协议中的应用消息由消息头、消息尾和消息体构成。其中：

**消息体的域根据业务需要可灵活扩展：一方面可对消息类型进行扩展；另一方面也可以对某一消息类型中的消息域灵活扩展。**

在GTP协议中，主要按照黄金交易前、交易中、交易后的应用场景，设计相关消息类型。

以竞价交易为例说明如下（实际消息类型以接口规范为准）：

### 交易前消息

对于交易员，主要指信息的获取，买卖成交的意向，如：公告信息、市场信息、合约信息、行情信息、询价等；

对于交易所，除信息发布外，还包括交易状态的切换等，如：市场状态切换、合约状态切换。

涉及部分消息类型如下：

| **消息种类** | **消息细类** | **消息名称** |
| --- | --- | --- |
| 基础信息发布 | 市场信息 | 市场信息回报 |
| 交割品种代码信息 | 交割品种代码信息 |
| 合约信息 | 合约信息回报 |
| 递延费 | 递延费率查询请求及应答 |
| 递延费率修改请求及应答 |
| 发布现货延期交收补偿费率通知 |
| 公告 | 公告发布请求及应答 |
| 交易所公告查询请求及应答 |
| 公告回报 |
| 市场/合约状态切换 | 状态切换模式 | 状态模式请求及应答 |
| 市场状态切换 | 市场状态改变请求及应答 |
| 市场状态改变回报 |
| 市场状态查询请求及应答 |
| 合约状态切换 | 合约状态改变请求及应答 |
| 合约/市场状态改变回报 |
| 合约状态查询请求及应答 |
| 行情服务 | 交易行情发布 | 交易实时行情 |
| 交易分钟行情 |
| 最终行情 |
| 交易行情查询 | **实时**行情查询请求及应答 |
| **分钟**行情查询请求及应答 |
| 交割行情 | 递延交收行情 |
| 国际行情 | 发布国际行情请求&应答 |
| 国际行情回报 |
| 国际行情查询请求及应答 |

### 交易中消息

对于交易员，主要指下单委托，竞价撮合的过程，如：委托报单、撤单、报单查询、成交单查询等；

对于交易所来说，主要指为满足交易员应急需要或风险控制需要而设计的一些消息，如：应急交易、强平交易、强撤交易等。

涉及部分消息类型如下：

| **消息种类** | **消息细类** | **消息名称** |
| --- | --- | --- |
| 竞价交易 | 普通交易 | 报单请求及应答 |
| 报单回报 |
| 撤单请求及应答 |
| 撤单回报 |
| 成交回报 |
| 报单查询请求及应答 |
| 成交单查询请求及应答 |
| 本地报单号查询请求及应答 |
| 应急交易 | 应急报单请求及应答 |
| 应急报单复核请求及应答 |
| 应急撤单请求及应答 |
| 应急报单查询请求及应答 |
| 强撤交易 | 强撤请求及应答 |
| 强平交易 | 超仓查询请求及应答 |
| 资金不足查询请求及应答 |
| 计算会员资金不足强平手数请求及应答 |
| 强平报单请求及应答 |
| 强平报单/撤单请求及应答 |
| 强平成交回报 |
| 强平成交单查询请求及应答 |
| 黄金ETF交易 | ETF账户备案 | ETF客户绑定请求及应答 |
| ETF客户解绑定请求及应答 |
| 认购交易 | ETF认购请求及应答 |
| ETF认购确认请求及应答 |
| 申购交易 | ETF申购请求及应答 |
| ETF申购回报 |
| ETF申购确认 |
| 赎回交易 | ETF赎回请求及应答 |
| ETF赎回确认 |
| 信息查询 | ETF申赎清单查询请求及应答 |
| 认申赎交易本地编号查询请求及应答 |
| 账户备案业务本地编号查询请求及应答 |
| 基金公司功能接口 | 发送账户绑定结果请求及应答 |
| 发送认购确认通知请求及应答 |
| 发送申赎清单系统控制版本请求及应答 |

### 交易后消息

对于交易员，主要指账户查询、保证金存取，库存处置过程，如：持仓/保证金/库存查询、会员保证金查询、出入金、租借、质押等；

对于交易所和清/结算行，则主要指与结算银行的清结算活动，如：往来账、对账等。

涉及部分消息类型如下：

| **消息种类** | **消息细类** | **消息名称** |
| --- | --- | --- |
| 保证金账户 | 账户查询 | 会员资金查询请求及应答 |
| 往来账 | 往来账消息 |
| 资金划拨请求及应答 |
| 保证金对账 | 对账汇总请求及应答 |
| 对账明细请求及应答 |
| 对账结果汇总请求及应答 |
| 对账结果明细请求及应答 |
| 保证金查询 | 往来账查询请求及应答 |
| 资金密码维护 | 增加资金密码请求及应答 |
| 修改资金密码请求及应答 |
| 重置资金密码请求及应答 |
| 持仓账户 | 账户查询 | 现货延期交收持仓查询请求及应答 |
| 会员期货持仓查询请求及应答 |
| 会员现货T+N持仓查询请求及应答 |
| 库存账户 | 账户查询 | 客户库存查询请求及应答 |
| 提货申请 | 增加提货申请请求及应答 |
| 撤销提货申请请求及应答 |
| 查询提货申请请求及应答 |
| 质押登记申报 | 提交质押登记申报请求及应答 |
| 撤销质押登记申报请求及应答 |
| 质押登记申报查询请求及应答 |
| 质押登记注销申报 | 提交质押登记注销申报请求及应答 |
| 质押登记注销申报查询请求及应答 |
| 租借登记申报 | 提交租借登记申报请求及应答 |
| 撤销租借登记申报请求及应答 |
| 租借登记申报查询请求及应答 |
| 仓储推送通知 | 推送客户库存变化流水 |
| 推送申报状态变化流水 |
| 仓储查询 | 查询清算用库存变化流水请求及应答 |
| 查询库存变化流水请求及应答 |
| 查询申报状态变化流水请求及应答 |
| 租借还金申报 | 提交租借还金申报请求及应答 |
| 撤销租借还金申报请求及应答 |
| 租借还金申报查询请求及应答 |

### 管理消息

主要指交易员身份认证、密码修改及其他活动。如：登录登出交易、密码修改等消息。

# 应用层消息重发机制

### 消息重发机制

在采用对话模式的交易流中，当因通讯网络异常导致交易发起方无法确认交易请求消息是否已送达，也无法确认交易处理方是否已给出响应时。双方再次连接后，交易请求方可按照如下三种方式处理[[5]](#footnote-5)：

**方式一：先查询，再处理**

交易请求方根据该笔交易的本地特征编号向交易处理方查询处理状态信息。

* 如果能够查到，交易请求方在本地更新的该笔交易的处理状态。
* 如果查不到，则作废处理，此时交易请求方可根据实际情况决定是否需要重提交易请求。

**方式二：直接重新提交请求**

交易请求方使用相同的本地特征编号重复提交该笔交易请求，交易处理方正常收到该笔请求后，进行如下处理：

* 如果发现该交易请求已被接收，交易处理方提示对方重复提交；
* 如果未被接受，则视作正常的交易请求处理。

**方式三：根据交易回报信息处理**

在交易发起方重新登录后，如果交易发起方正好收到关于该笔请求的交易回报信息，则可视为该笔交易请求已被处理，直接根据回报信息，更新该笔交易的本地处理状态。

通常根据业务类型不同，需要设计不同的应用消息去进行消息恢复。以：竞价交易和黄金ETF交易为例，应用示例如下：

### 应用示例：报单/撤单消息恢复



### 应用示例：ETF交易消息恢复



# 附录



1. 参考FTD协议中的定义 [↑](#footnote-ref-1)
2. 公共流通常由交易系统维护，行情流通常由行情系统单独维护。 [↑](#footnote-ref-2)
3. 对于采用多套消息序号的消息，消息发送方可在本地分别记录在多个流文件。 [↑](#footnote-ref-3)
4. 参照FTD协议定义。在主板交易接口中主要用到C、L，对于单个报文标识L，对于连续报文，前几个标识C，最后一个报文标识L [↑](#footnote-ref-4)
5. 如果接入方系统与交易所前置间为单连接，则只能在连接恢复后发送；如果多连接，可通过另一正常连接进行处理 [↑](#footnote-ref-5)