



IS120 上海证券交易所行情网关

STEP 数据接口规范

0.6059 版



上海证券交易所

二〇二六五年一十月

修订记录

2018-03-09, 0.10 版, 文档创建。

2018-03-25, 0.20 版, 根据原有文件接口进行字段及内容调整。

2018-07-11, 0.30 版, 根据反馈意见调整部分说明、调整价格精度、增加成交笔数及期权虚拟匹配数量。

2018-07-25, TradingPhaseCode 闭市集合竞价相关调整。

2019-01-10, 0.31 版, 增加债券回购延长对市场状态消息字段的说明。

2019-01-25, 0.32 版, 增加盘后固定价格交易的行情接口说明, 调整国债预发行接口字段取值。

2019-03-04, 调整盘后固定价格行情的产品状态取值。

2019-12-05, 0.40 版, 原内容移入第二章, 增加章节描述通过行情网关接收的文件及外部转发数据。

2020-05-20, 0.41 版, 变更市场状态消息、行情快照消息的消息头中 SendingTime 为交易所时间。

2021-01-19, 0.42 版, 行情快照消息增加公募 REITs 相关描述。

2021-03-05, 0.50 版, 调整债券产品为单独证券类型。

2021-03-24, 0.51 版, 调整股票(含指数)全市场行情状态中债券质押回购行情结束标志的相关描述。

2021-08-06, 0.52 版, 调整行情接口中债券相关的行情数据说明。

2022-01-11, 0.53 版, 调整内容如下:

1. 调整行情接口中债券指数、债券和回购的数量单位;

2. 调整 STEP 行情接口中“成交金额”字段长度，与 BINARY 接口保持一致；
3. 移除市场行情状态（SecurityType = 01）中债券质押回购行情结束标志信息。

2022-01-17, 0.54 版，调整内容如下：

1. 调整市场状态消息中股票、基金、指数及债券分销（SecurityType = 01）和债券市场行情（SecurityType = 12）描述；

2022-03-21, 0.55 版，增加 B 转 H 行情文件接收相关描述。（本文档无调整）

2022-04-13, 0.56 版，增加期权基础信息第二版文件接收相关描述。（本文档无调整）

2023-08-14, 0.57 版，调整内容如下：

1. 明确 PreCloseIOPV（基金 T-1 日收盘时刻 IOPV）、IOPV（基金 IOPV）字段适用范围以及和 ETF 公告文件的关联关系。
2. 增加基金通行情接收相关描述。（本文档无调整）

2025-08-01, 0.58 版，在市场状态消息（MsgType=h）、行情快照消息（MsgType=W）中增加独立 IOPV 行情描述。

2025-10-17, 0.59 版，支持 ETF 公告文件 xml 版，调整 2.4.2 集中竞价类行情快照扩展字段章节 IOPV 行情与 ETF 公告文件关联关系描述。

2026-01-22, 0.60 版，市场状态消息（MsgType=M101），增加在综合业务市场无支持开盘集合竞价的产品、无发行期内国债预发行产品的情况下，字段无意义说明。

目 录

1	引言	1
1.1	名词释义	1
2	STEP 实时行情	2
2.1	会话机制	2
2.1.1	消息序号	3
2.1.2	会话安全	3
2.1.3	建立行情会话	3
2.1.4	行情数据发布	3
2.1.5	关闭行情会话	4
2.1.6	心跳	4
2.1.7	行情网关主动关闭行情会话的情况	4
2.2	协议介绍	5
2.2.1	字段说明	5
2.2.2	STEP 消息头	5
2.2.3	STEP 消息尾	6
2.2.4	STEP 消息完整性	6
2.3	会话消息	7
2.3.1	登录消息 ($MsgType=1$)	7
2.3.2	注销消息 ($MsgType=5$)	8

2.3.3	心跳消息 (<i>MsgType=0</i>)	9
2.3.4	测试请求消息 (<i>MsgType=1</i>)	9
2.3.5	重发请求消息 (<i>MsgType=2</i>)	10
2.3.6	会话拒绝消息 (<i>MsgType=3</i>)	10
2.3.7	序号重设消息 (<i>MsgType=4</i>)	11
2.4	应用消息	12
2.4.1	市场状态消息 (<i>MsgType=h</i>)	12
2.4.2	行情快照消息 (<i>MsgType=W</i>)	15
3	文件接收	24
4	转发行情	25
5	后记	26
	附录一 计算校验和	27

1 引言

上海证券交易所行情网关数据接口规范包括 BINARY 与 STEP 两卷，本卷主要介绍 STEP 数据接口规范（第二章），并对行情网关可接收文件（第三章）和转发行情（第四章）进行一定的说明，请市场参与者结合《上海证券交易所行情网关技术指引及接口开发指南》一并使用。

文档采用的术语及消息内容与 STEP 数据接口规范具有对应关系，可以互为参考。

1.1 名词释义

名词	含义
MDGW	Market Data GateWay 行情网关
VSS	Vendor Supplied System 用户行情系统
STEP	Securities Trading Exchange Protocol 证券交易数据交换协议
FIX	Financial Information Exchange 金融信息交换协议

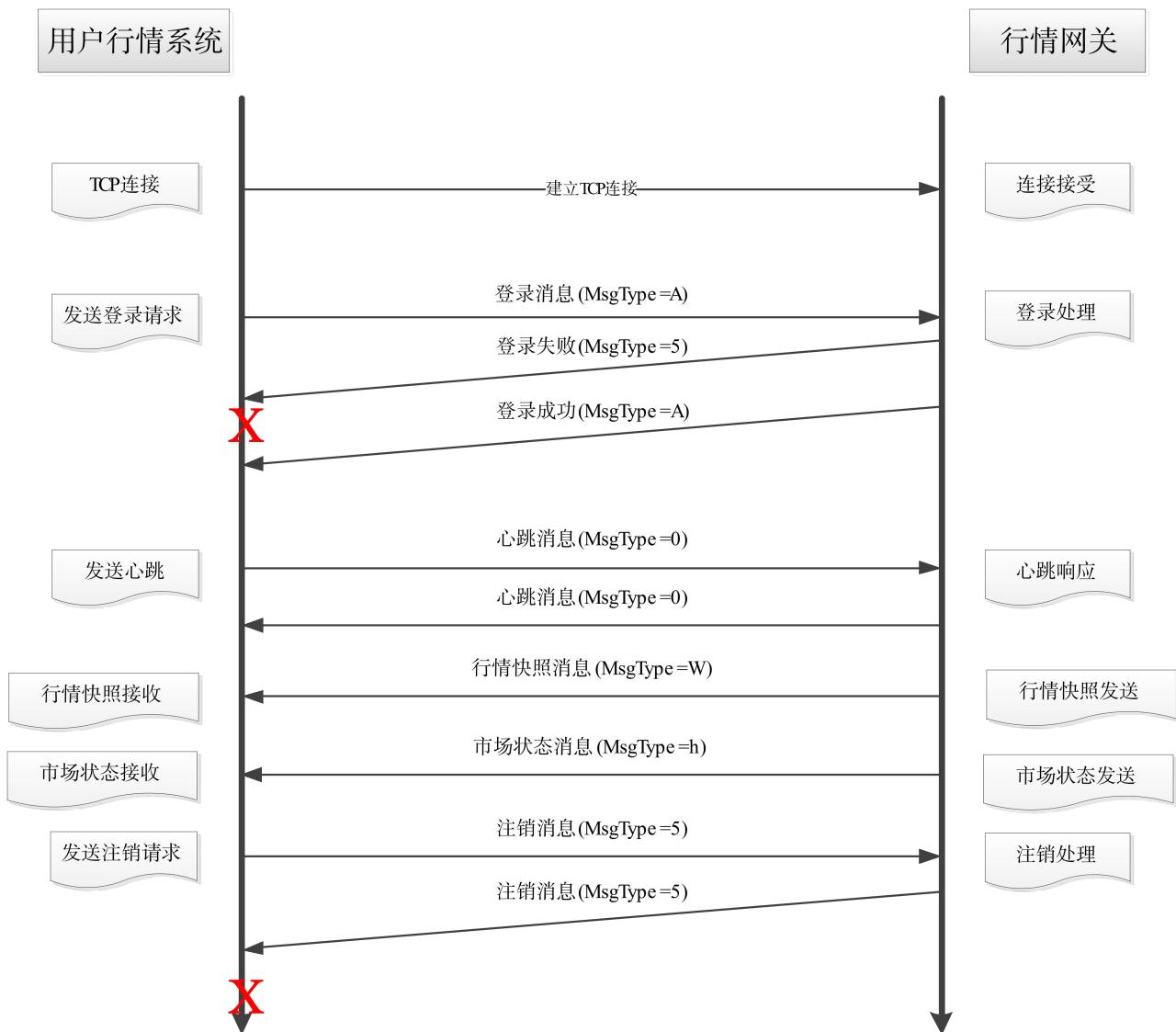
2 STEP 实时行情

本章描述市场参与者与信息服务商等用户行情系统,通过 STEP 协议接入上海证券交易所行情网关 MDGW 进行行情数据传输的机制和相关数据交换格式。

2.1 会话机制

用户行情系统 (VSS) 通过 TCP 方式接收行情网关 (MDGW) 发布的流式行情, 会话机制遵循《轻量级 STEP 会话层接口规范》。

交互示意图如下:



STEP 协议交互图

2.1.1 消息序号

会话双方收发的每条消息都被分配有一个消息序号 MsgSeqNum 来唯一标识。参与通信的每一端都需要维护一对序号 (NxtIn, NxtOut)，NxtIn 表示下一个期望的入向消息序号，NxtOut 表示下一个出向消息序号。消息序号一般在每次会话过程中从 1 开始，在整个会话过程中连续递增，直到该会话过程全部结束。

2.1.2 会话安全

MDGW 与 VSS 之间传输的数据是非加密的，数据传输安全由接入用户网络保证。

2.1.3 建立行情会话

每个行情会话都是完全独立的，同一 VSS 在一个交易日内的多次登录被视为不同的行情会话。

建立行情会话包含两个步骤：建立 TCP 连接、登录。具体步骤如下：

1、TCP 连接

VSS 与 MDGW 建立 TCP 通讯连接，其中 MDGW 是服务端，VSS 负责发起连接。

2、登录

VSS 发送的第一个消息必须是登录消息。如果 VSS 登录成功，则 MDGW 发送一个登录消息作为应答。如果 VSS 登录失败，MDGW 则在发送一个含失败说明的注销消息后由 VSS 主动关闭连接。VSS 必须在收到 MDGW 的登录消息之后才允许发送其他消息。

2.1.4 行情数据发布

在完成建立行情会话之后，MDGW 将向 VSS 发送行情数据消息。行情数据消息格式将在

第 5 章“应用消息”中详细叙述。MDGW 目前发布的行情应用消息不支持重传。

2.1.5 关闭行情会话

行情会话的正常关闭是通过连接双方互相发送注销消息完成的。MDGW 和 VSS 均可以主动发送注销消息，接收方需要回传注销消息作为应答。如果超过预定时间（一般为 5 秒）没有收到对方回传的注销消息，任何一方均有权主动关闭连接。

2.1.6 心跳

连接双方在数据发送的空间期间应主动发送心跳消息，通过心跳消息可以监控行情会话的状态。心跳最小间隔由登录消息中的 HeartBtInt 域确定。

连接双方在发送任何消息后，应立即重新设置心跳间隔计时器。如果 VSS 超过 2 个 HeartBtInt 指定周期没有收到 MDGW 发送的任何消息，则行情会话被视为可能存在异常，VSS 需要重新建立行情会话。

2.1.7 行情网关主动关闭行情会话的情况

在以下几种场景（包括但不限于）下，MDGW 会主动关闭行情会话：

- 1、VSS 与 MDGW 建立 TCP 连接之后，超过预定时间（一般为 5 秒）未发送消息完成登录；
- 2、VSS 在登录失败之后，长时间不关闭 TCP 连接；
- 3、VSS 在注销之后，长时间不关闭 TCP 连接；
- 4、MDGW 与交易所行情主机连接异常；
- 5、VSS 如未及时处理 MDGW 发送的数据，导致 MDGW 内积压的待发送消息超过特定阈值。

2.2 协议介绍

每条 STEP 消息由消息头、消息体和消息尾组成，消息最大长度为 8K 字节。

2.2.1 字段说明

字段类型说明如下：

类型	说明
CX	代表该字段内容为文本，CX 代表该字符串的最大长度，其中 X 为大于零的数字，例如 C5 代表最大长度为 5 的文本；当最大长度大于实际长度时可以不补空格。字符串使用 GBK 编码。
NX NX(Y)	代表该字段内容为数值，NX 代表该字符串为整数，X 为该整数的最大长度；NX(Y) 代表该字符串为浮点数，X 代表该字符串的最大长度，Y 代表小数位数，X 包括一位小数点，此时整数部分最多不超过 X-Y-1 位，小数部分最多不超过 Y 位。
Boolean	C1，代表该字符串内容为布尔值，有效取值是 Y 或者 N；‘Y’ 表示 Yes/True，‘N’ 表示 No/False

注 1：字段类型中如有*注释则为固定长度，否则为最大长度。

注 2：除非特别说明，消息格式不指定字段在 STEP 消息中的前后位置。

2.2.2 STEP 消息头

每一个会话或应用消息都有一个消息头，该消息头指明消息类型、消息体长度、消息序号、发送方、接收方和发送时间等信息。

消息头格式如下：

Tag	域名	必需	说明	类型

8	BeginString	Y	起始串，固定为 FIXT.1.1(消息的第一个域)	C16
9	BodyLength	Y	消息体长度(消息的第二个域)	N9
35	MsgType	Y	消息类型(消息的第三个域)	C16
49	SenderCompID	Y	发送方代码	C32
56	TargetCompID	Y	接收方代码	C32
34	MsgSeqNum	Y	消息序号	N18
43	PossDupFlag	N	会话层可能重传标志	Boolean
97	PossResend	N	应用层可能重发标志	Boolean
52	SendingTime	Y	发送时间，格式：YYYYMMDD-HH:mm:SSsss	*C21
347	MessageEncoding	N	字符编码类型方式，固定为 GBK	C16

说明：

- 当消息为市场状态消息或行情快照消息时， SendingTime 为交易所时间。

2.2.3 STEP 消息尾

每一个会话或应用消息有一个消息尾，并以此终止。消息尾可用于分隔多个消息，包含有 3 位数的校验和值。

消息尾格式如下：

Tag	域名	必需	说明	类型
10	CheckSum	Y	校验和，消息的最末域	*C3

2.2.4 STEP 消息完整性

STEP 消息完整性通过两个方法保证：消息体长度和校验和的验证。

消息长度是以 BodyLength 域来表示，可以通过清点消息之中跟在 BodyLength 域之后、直

至并包括直接先于 CheckSum 域号（‘10=’）出现的那个域界定符<SOH>之间的字符来验证。

校验和是把每个字符的二进制值从消息开头‘8=’中的‘8’开始相加，一直加到紧靠在校验和域‘10=’之前的域界定符，然后取按 256 取模得到的结果。

校验和域位于消息的最末一个。计算校验和的代码段可参考附录一“计算校验和”。

2.3 会话消息

会话消息将在以下各节中予以介绍，并定义会话消息格式。

连接双方均可生成会话消息。

2.3.1 登录消息（MsgType=A）

登录消息应是在行情会话开始时 VSS 发送的第一个消息。MDGW 只作为登陆的接受方，不会作为登陆的发起方。

HeartBtInt 域用来指定心跳消息的发送时间间隔，必须设置为大于 0 的整数。VSS 需要在登录消息中填入预期的心跳时间间隔，MDGW 在回传登录消息时返回的 HeartBtInt 域作为协商后的心跳时间间隔。

登录请求消息格式如下：

Tag	域名	必 需	说明	类型
	标准消息头	Y	<i>MsgType = A</i>	
98	EncryptMethod	Y	加密方法，固定为 0	N8
108	HeartBtInt	Y	心跳间隔，单位为秒	N8
141	ResetSeqNumFlag	N	双方序号重置为 1 的标记（请求时必填 Y）	Boolean

789	NextExpectedMsgSeqNum	N	接收方期望得到的下一条消息序号(请求时必填 1)	N18
553	Username	N	用户名 (预留)	C32
554	Password	N	密码 (预留)	C32
1137	DefaultApplVerID	Y	本次会话中使用的 FIX 消息的缺省版本。对于 本文涉及的行情发布而言，固定填为 9	C8
1407	DefaultApplExtID	N	本次会话中使用的FIX 消息[在Tag1137 基础 上]的缺省扩展包。对于本文涉及的行情发布而 言，固定填为124	N8
1408	DefaultCstmApplVerID	N	本次会话中，FIX 消息的缺省自定义应用版本。 本标签是对tag 1137 + tag 1407 的进一步约束。 填写协议版本，如STEP1.20_SH_0.30	C32
	标准消息尾	Y		

2.3.2 注销消息 (**MsgType=5**)

注销消息是发起或确认行情会话终止的消息。未经注销消息交换而断开连接，一律视为非正常的断开。

连接双方在发送注销消息之后不应发送任何消息。

注销消息格式如下：

Tag	域名	必需	说明	类型
	标准消息头	Y	<i>MsgType = 5</i>	

1409	SessionStatus	N	Logout 时的会话状态 0 正常注销 【1-999】一般情况注销，重连可以恢复 【1000-9999】严重情况注销，建议切换服务器	N4
58	Text	N	文本	C1024
	标准消息尾	Y		

2.3.3 心跳消息 (MsgType=0)

心跳消息用于监控通信连接的状况。

当连接的任何一方在心跳时间间隔（由 HeartBtInt 域指定）时间内没有接收或发送任何数据的时候，需要产生一个心跳消息并发送出去。如果接收方在 2 倍心跳时间间隔内都没有收到任何消息的时候，那么可认为行情会话出现异常，可以立即关闭 TCP 连接。

心跳消息格式如下：

Tag	域名	必需	说明	类型
	标准消息头	Y	$MsgType = 0$	
112	TestReqID	N	如是对 TestRequest 响应而发送的心跳消息，则应包含本域。本域的内容直接来自于触发本心跳消息的 TestRequest 消息的内容	C32
	标准消息尾	Y		

2.3.4 测试请求消息 (MsgType=1)

测试请求消息能强制对方发出心跳消息。测试请求消息的作用是检查对方消息序号和检查通

信线路的状况。对方用带有测试请求标识符（TestReqID）的心跳作应答。MDGW 不会主动发送此消息，但会遵循 FIX 标准引擎规则而响应 VSS 的测试请求消息。

测试请求消息格式如下：

Tag	域名	必需	说明	类型
	标准消息头	Y	<i>MsgType = 1</i>	
112	TestReqID	N	测试请求标识符	C32
	标准消息尾	Y		

2.3.5 重发请求消息（MsgType=2）

MDGW 不会主动发出会话层的重传请求，只为兼容 FIX 标准引擎提供对该请求的响应。

MDGW 接收到重发请求消息，只会通过序号重设消息（4.7）响应。

重发请求消息格式如下：

Tag	域名	必需	说明	类型
	标准消息头	Y	<i>MsgType = 2</i>	
7	BeginSeqNo	Y	起始消息序号	N18
16	EndSeqNo	Y	结束消息序号	N18
	标准消息尾	Y		

2.3.6 会话拒绝消息（MsgType=3）

当接收方收到一条消息，由于违反了会话层规则而不能适当地处理该消息时，应该发出会话拒绝消息。MDGW 不会主动发出会话拒绝消息。

会话拒绝消息格式如下：

Tag	域名	必需	说明	类型

	标准消息头	Y	<i>MsgType</i> = 3	
45	RefSeqNum	Y	关联消息的序号，即被拒绝消息的序号	N18
371	RefTagID	N	相关错误消息中，出现错误的 FIX 域号	N6
372	RefMsgType	N	相关错误消息的 <i>MsgType</i>	C16
373	SessionRejectReason	N	会话拒绝原因编号	N5
58	Text	N	文本，拒绝的原因描述	C1024
	标准消息尾	Y		

2.3.7 序号重设消息 (*MsgType*=4)

序号重设消息用于告知接收方下一个消息的消息序号。序号重设消息的 *MsgSeqNum* 按标准 FIX 协议规定可以任意填写且接收方不会检查，建议固定填写为 1。MDGW 不会主动发出序号重设消息，只为兼容 FIX 标准引擎。当 MDGW 收到用户的重传请求时，以序号重设消息予以响应。当 MDGW 收到用户序号重设消息，则重置入向消息序号 *NxtIn* = *NewSeqNo*。

序号重设消息格式如下：

Tag	域名	必需	说明	类型
	标准消息头	Y	<i>MsgType</i> = 4	
123	GapFillFlag	N	缺口填补标志	Boolean
36	NewSeqNo	Y	新消息序号	N18
	标准消息尾	Y		

2.4 应用消息

2.4.1 市场状态消息 (MsgType=h)

市场状态消息用于交易所发布市场状态和产品数量等信息，会周期性发布以及在市场状态变化时发布。

市场状态消息的格式如下：

Tag	域名	必需	说明	类型
	标准消息头	Y	<i>MsgType=h</i>	
167	SecurityType	Y	<p>证券类型</p> <p>01 股票、基金、指数及债券分销</p> <p>02 衍生品</p> <p>03 综合业务</p> <p>12 债券</p> <p>14 外部源行情</p>	*C2
339	TradSesMode	Y	<p>交易盘交易模式</p> <p>1 = 系统测试</p> <p>2 = 模拟交易</p> <p>3 = 产品（正常交易）</p>	*N1

336	TradingSessionID	Y	全市场行情状态： 该字段为 8 位字符串，左起每位表示特定的含义，无定义则填空格。	*C8
393	TotNoRelatedSym	Y	最大产品数目（包括指数）	N8
	标准消息尾	Y		

说明：

1. 衍生品指期权业务；
2. 综合业务指国债预发行、盘后固定价格交易业务；
3. 全市场行情状态(TradingSessionID)为 8 位字符串，左起每位表示特定的含义，无定义则填空格。该字段具体含义在不同证券类型时说明如下：

SecurityType = 01 ¹	
第 1 位	‘S’表示全市场启动期间（开市前）， ‘T’表示全市场处于交易期间（含中间休市） ‘E’表示全市场处于闭市期间
第 2 位	‘1’表示开盘集合竞价结束标志，未结束取‘0’
第 3 位	‘1’表示全市场行情结束标志，未结束取‘0’
第 4 位	‘1’表示上海市场行情结束标志，未结束取‘0’
SecurityType = 02	

¹ SecurityType = 01 时，上海市场包括股票、基金、债券分销。全市场包括上海市场及指数。

第 1 位	‘S’表示期权市场启动期间（开市前） ‘T’表示期权市场处于交易期间（含中间休市） ‘E’表示期权市场处于闭市期间
第 2 位	‘1’表示期权市场开盘集合竞价结束标志，未结束取‘0’
第 3 位	‘1’表示期权市场行情闭市标志，未闭市取‘0’
SecurityType = 03	
第 1 位	‘S’表示综合业务市场启动期间（开市前） ‘T’表示综合业务市场处于交易期间（含中间休市） ‘E’表示综合业务市场处于闭市期间
第 2 位	‘1’表示综合业务市场开盘集合竞价结束标志，未结束取‘0’。 <u>当综合业务市场无支持开盘集合竞价的产品时，无意义</u>
第 3 位	‘1’表示综合业务市场收盘集合竞价结束标志，未结束取‘0’， <u>暂不启用</u>
第 4 位	‘1’表示国债预发行市场行情闭市标志，未闭市取‘0’。 <u>当无发行期内国债预发行产品时，无意义</u>
第 5 位	‘1’表示盘后固定价格交易行情闭市标志，未闭市取‘0’
SecurityType = 12²	
第 1 位	‘S’表示债券市场启动期间（开市前） ‘T’表示债券市场处于交易期间（含中间休市） ‘E’表示债券市场处于闭市期间

² SecurityType = 12 时，债券市场包含债券现券及质押式回购。

第 2 位	‘1’表示债券市场开盘集合竞价结束标志，未结束取‘0’
第 3 位	‘1’表示债券市场行情结束标志，未结束取‘0’
第 4 位	‘1’表示债券现券（可转债及新标准券）行情结束标志，未结束取‘0’
第 5 位	‘1’表示债券质押回购、债券现券（除可转债及新标准券）行情结束标志，未结束取‘0’。
SecurityType = 14	
第 1 位-第 8 位	全空格

2.4.2 行情快照消息 (MsgType=W)

行情快照消息用于发布证券产品行情，交易所会周期性发布全量快照消息以及在行情变化时发送增量快照消息。多条快照消息之间无数据依赖性，VSS 无需区分全量/增量消息，可直接替换当前数据。

消息格式如下：

Tag	域名	必须	说明	类型
	标准消息头	Y	<i>MsgType=W</i>	

167	SecurityType	Y	证券类型 01 股票、基金、指数及债券 分销 02 衍生品 03 综合业务 12 债券 14 外部源行情	*C2
339	TradSesMode	Y	交易盘交易模式 1 = 系统测试 2 = 模拟交易 3 = 产品（正常交易）	*N1
75	TradeDate	Y	交易日期 YYYYMMDD	*N8
779	LastUpdateTime	N	最新更新时间 HHMMSSsss	*N9
1500	MDStreamID	Y	行情类别	*C5
48	SecurityID	Y	产品代码	C8
55	Symbol	N	产品简称	C8
140	PrevClosePx	N	昨收盘	N14(5)
387	TotalVolumeTraded	N	成交量	N16
8503	NumTrades	N	成交笔数	N16
8504	TotalValueTraded	N	成交金额	N17(2)
268	NoMDEntries	Y	行情条目个数	N5

→	269	MDEntryType	Y	行情条目类别 0 = 买入价 (270, 271, 290) 1 = 卖出价 (270, 271, 290) 2 = 成交价 (270) 3 = 指数(270) 4 = 今开盘价 (270) 5 = 今收盘价 (270) 6 = 今结算价 (270) 7 = 当日最高成交价 (270) 8 = 当日最低成交价 (270) v = IOPV (270) w=前一日 IOPV (270) x = 动态参考价格及虚拟匹 配数量 (270、271) z1=昨日结算价 (270) z2=总持仓量 (271)	C2
→	270	MDEntryPx	N	价格	N14(5)
→	271	MDEntrySize	N	数量	N12
→	290	MDEntryPositi onNo	N	档位	N2
8538	TradingPhaseCode		N	产品状态	*C8

	标准消息尾	Y		
--	-------	---	--	--

说明：

1. 字段无意义或无该字段行情数据时，字符填空格，数值填 0；数值字段取值超过约定格式最大值时，取最大值，如 N10 取 9999999999；
2. 行情类别(MDStreamID)及其扩展字段如下：

SecurityType	MDStreamID	说明
01	MD001	指数类产品
	MD002	股票（A、B 股）类产品
	MD003	债券分销类产品
	MD004	基金类产品（含公募 REITs）
02	MD301	期权类产品
03	MD101	国债预发行产品
	MD102	盘后固定价格
12	MD201	债券类产品 ³
14	MDE01	IOPV

当 SecurityType =14 且 MDStreamID= MDE01 时，PrevClosePx 昨收盘、TotalVolumeTraded 成交数量、NumTrades 成交笔数、TotalValueTraded 成交金额字段取值为 0, TradingPhaseCode 产品状态取值为空。

3. 对数量单位说明如下：

- a) 产品价格、金额单位，除 B 股为美元外，其他为人民币元；
- b) 指数的成交数量(TradeVolume)为参与计算相应指数的交易数量，股票指数交易数量单位是 100 股，基金指数的交易数量单位是 100 份，债券指数的交易数量单位是千元面额；指数成交金额(TotalValueTraded)为参与计算相应指数的成交金额；
- c) 各类产品数量与成交数量单位，股票为股，基金、公募 REITs 为份，债券与回购为千元面额，期权合约的数量单位为张；
- d) 对于债券分销，成交金额为每笔成交的“价格*数量*10”的总和；对于债券现券，成交金额为每笔成交的“价格*数量*10”的总和；对于质押式回购，成交金额为 100*成交数量*10。

4. 实时阶段及标志(TradingPhaseCode)为 8 位字符串，左起每位表示特定的含义，无定义则填空格。该

³ 包含债券现券及质押式回购类产品。

字段具体含义在不同行情类别时说明如下：

SecurityType = 01, MDStreamID= MD001	
全为空格（预留）	
SecurityType = 01, MDStreamID= MD002, MD003, MD004	
第 1 位	‘S’表示启动（开市前）时段 ‘C’表示开盘集合竞价时段 ‘T’表示连续交易时段 ‘E’表示闭市时段 ‘P’表示产品停牌 ‘M’表示可恢复交易的熔断时段（盘中集合竞价） ‘N’表示不可恢复交易的熔断时段（暂停交易至闭市） ‘U’表示收盘集合竞价时段
第 2 位	‘0’表示此产品不可正常交易 ‘1’表示此产品可正常交易 在产品进入开盘集合竞价、连续交易、收盘集合竞价、熔断（盘中集合竞价）状态时值为‘1’，在产品进入停牌、熔断（暂停交易至闭市）状态时值为‘0’，且 闭市后保持该产品闭市前的是否可正常交易状态。
第 3 位	‘0’表示未上市 ‘1’表示已上市

第 4 位	<p>‘0’表示此产品在当前时段不接受进行新订单申报</p> <p>‘1’ 表示此产品在当前时段可接受进行新订单申报</p> <p>仅在交易时段有效，在非交易时段无效</p>
SecurityType = 02, MDStreamID= MD301	
第 1 位	<p>‘S’表示启动（开市前）时段</p> <p>‘C’表示集合竞价时段</p> <p>‘T’表示连续交易时段</p> <p>‘B’表示休市时段</p> <p>‘E’表示闭市时段</p> <p>‘V’表示波动性中断</p> <p>‘P’表示临时停牌</p> <p>‘U’表示收盘集合竞价</p> <p>‘M’表示可恢复交易的熔断（盘中集合竞价）</p> <p>‘N’表示不可恢复交易的熔断（暂停交易至闭市）</p>
第 2 位	<p>‘0’表示未连续停牌</p> <p>‘1’表示连续停牌</p> <p>（预留，暂填空格）</p>

第 3 位	‘0’表示不限制开仓 ‘1’表示限制备兑开仓 ‘2’表示限制卖出开仓 ‘3’表示限制卖出开仓、备兑开仓 ‘4’表示限制买入开仓 ‘5’表示限制买入开仓、备兑开仓 ‘6’表示限制买入开仓、卖出开仓 ‘7’表示限制买入开仓、卖出开仓、备兑开仓
第 4 位	‘0’表示此产品在当前时段不接受进行新订单申报 ‘1’ 表示此产品在当前时段可接受进行新订单申报 仅在交易时段有效，在非交易时段无效
SecurityType = 03, MDStreamID= MD101	
第 1 位	‘S’表示启动（开市前）时段 ‘C’表示 <u>开盘</u> 集合竞价时段 ‘T’表示连续交易时段 ‘E’表示闭市时段 ‘P’表示停牌
SecurityType = 03, MDStreamID= MD102	

第 1 位	<p>‘T’表示启动（开市前）时段</p> <p>‘A’表示集中撮合时段</p> <p>‘H’表示连续交易时段</p> <p>‘D’表示闭市时段</p> <p>‘F’表示停牌</p>
SecurityType = 12, MDStreamID= MD201	
第 1 位	<p>‘S’表示启动（开市前）时段</p> <p>‘C’表示开盘集合竞价时段</p> <p>‘T’表示连续交易时段</p> <p>‘E’表示闭市时段</p> <p>‘P’表示产品停牌</p>
第 2 位	<p>‘0’表示此产品不可正常交易</p> <p>‘1’表示此产品可正常交易</p> <p>在产品进入开盘集合竞价、连续交易状态时值为‘1’，在产品进入停牌状态时值为‘0’，且闭市后保持该产品闭市前的是否可正常交易状态。</p>
第 3 位	<p>‘0’表示未上市</p> <p>‘1’表示已上市</p>
第 4 位	<p>‘0’表示此产品在当前时段不接受进行新订单申报</p> <p>‘1’ 表示此产品在当前时段可接受进行新订单申报</p> <p>仅在交易时段有效，在非交易时段无效。</p>
SecurityType = 14, MDStreamID= MDE01	

第1位-第8位	全空格
---------	-----

5. 对行情条目循环体说明如下：

- a) 行情条目类别为 0 或 1 时，表示该条目为买卖盘档位，此时 MDEntryPx、MDEntrySize、MDEntryPositionNo 分别表示买卖价格、数量和档位序号，MDEntryPositionNo 从 0 开始计数；
- b) 行情条目类别为 w 时，MDEntryPx 表示基金 T-1 日收盘时刻 IOPV⁴，此条目仅当 MDStreamID=MD004 时有意义，当 MDStreamID=MDE01 时无意义；行情条目类别为 v 时，MDEntryPx 表示基金 IOPV⁵，此条目仅当 MDStreamID=MD004 或 MDE01 时有意义；
- c) 行情条目类别为 x 时，MDEntryPx 表示动态参考价格、MDEntrySize 表示虚拟匹配数量，仅当 MDStreamID=MD301 时有意义；
- d) 行情条目类别为 z2 时，MDEntrySize 表示期权合约总持仓量，单位是(张)，仅当 MDStreamID=MD301 时有意义；
- e) 其他行情条目类别，将仅通过 MDEntryPx 表示相应值，其他字段无意义；
- f) 在集合竞价时段内，当前买入价和当前卖出价中同时为虚拟开盘参考价格，即根据集合竞价算法计算得出的虚拟撮合价格。同时申买量一和申卖量一为行情发布时刻的虚拟匹配量。申买量二为行情发布时刻的买方虚拟未匹配量。申卖量二为行情发布时刻的卖方虚拟未匹配量；
- g) 对期权产品 (MDStreamID=MD301)，今日结算价通过期权收盘价格文件 (clpr03MMDD.txt) 揭示。
- h) 对盘后固定价格行情 (MDStreamID=MD102)，仅今日收盘价、买一数量、卖一数量有意义。其中，买一数量表示当前未成交的买入申报总股数，卖一数量表示当前未成交的卖出申报总股数；

⁴ 仅对 ETF 产品，此字段有效。该字段取 0 时，无意义。该字段不作除权除息等调整。

⁵ 仅对 ETF 产品，此字段有效。该字段取 0 时，无意义。当 ETF 公告文件 Publish (TXT 版) /PublishIOPVFlag (XML 版) 字段 (是否需要发布 IOPV) 取值为 1 时，此字段为对应产品的 IOPV 值；当 ETF 公告文件 Publish (TXT 版) /PublishIOPVFlag (XML 版) 字段取值为 0 时，此字段固定取 0，无意义；

3 文件接收

同《IS120 上海证券交易所行情网关 BINARY 数据接口规范》中的文件接收章节。

4 转发行情

同《IS120 上海证券交易所行情网关 BINARY 数据接口规范》中的转发行情章节。

5 后记

上海证券交易所对本文档享有知识产权，未经上海证券交易所书面许可，任何单位和个人不得将本文档用于其他商业目的。

本文档编写过程中，深受证券业界信息技术同仁讨论启发，特此致谢。对本文档有任何批评指正意见，可通过电子邮件或服务电话进行反馈。

电子邮件：tech_support@sse.com.cn

服务电话：4008888400-2

网站地址：<https://www.sse.com.cn/> - 交易技术支持专区

通讯地址：上海市浦东新区浦东南路 528 号 上交所技术公司技术开发总部

附录一 计算校验和

以下为计算校验和的代码段：

```
const char* CalcChecksum(const char* buffer, uint32 len, char bufChecksum[4])  
{  
    uint8 checksum = 0;  
  
    uint32 i = 0;  
  
    for (i = 0; i < len; i++)  
    {  
        checksum += (uint8)buffer[i];  
    }  
  
    sprintf(bufChecksum, "%03d", checksum);  
  
    return bufChecksum;  
}
```