**联网结算模块实现设计**

# 一、目标

**阶段1目标**

实现从海岭Oracle数据库向联网发送数据:

**阶段2目标**

实现从Mysql数据库向联网发送数据；数据库字段表一致。

# 二、功能实现

## 2.1 参照海岭现有代码实现功能

开4个线程:

线程1： 从车道消费数据表中一次查找100条未打包的数据，(5分钟或100条)，按照联网交互的格式要求生成XML并按照7z方式打包；存入指定的文件夹中；并更新数据库打包标记；注意使用事务更新包号，打包时间。

线程2：从打包文件夹中定期扫描文件，若有则读取，按照联网规范发送至联网的数据端口中；发送成功后将文件转移至另外文件夹；

线程3：服务监听端口，接收联网发送过来的数据包，并存储在指定文件夹；

线程4：定期扫描接收联网的接收数据的文件夹，解压后分析包。更新应答包状态、争议包状态、记账包状态、清分包状态、退费包状态；

## 2.2 数据核对结果

1、每日发送的数据包，哪些没有回记账包；没有回的在运维过程中重发功能；

2、清分包的数据总数是否与发送的数据一致，数据库记录差异；

## 2.3 异常处理及日志

更新数据库异常、打包异常、发送异常的错误处理机制及日志记录；

# 三、其他

## 3.1 文档完善

将更新的字段及内容含义在文档中更新，设计文档更新。

## 3.2 时间要求

本月底之前完成开发及测试；

# 四、接口规范

# 1、数据库

mysql数据库；用golang xorm操作mysql数据库；

# 2、线程

## 2.1 线程1

从原始交易结算数据表中一次查找100条未打包的记录数据，(5分钟或100条)，未打包数据的数据有标记；  
 按照联网交互的格式要求生成XML文件存于generatexml文件夹。 并按照 LZ77 方式打包；存入指定的文件夹中sendzipxml；

并更新数据库打包标记；注意使用事务更新包号，打包时间。

### 具体步骤：

（1）查询原始交易数据（在数据层）

包编号的规则：配置文件记录

按照卡网络号打包

取数据流程

逻辑参数

更新原始库的逻辑

（2）准备数据（在数据层）

（3）Xml数据生成Xml文件、压缩，存文件

（4）Md5计算

（5）7z压缩（Cgo解决）

（6）插入原始交易消息包记录

（7）更新原始交易数据的状态

更新的流程

### Xml数据格式校准：

一个原始交易消息包type Message struct {

XMLName xml.Name `xml:"Message"`

Header Header `xml:"Header"`

Body Body `xml:"Body"`

}

type Header struct {

XMLName xml.Name `xml:"Header"`

Version string  *//统一 00010000 Hex(8) Header* MessageClass int32  *//消息传输的机制*

MessageType int32  *//消息的应用类型* SenderId string  *// Hex(16位，不足补零) 发送方Id* ReceiverId string  *//Hex(16位，不足补零) 接收方Id* MessageId int64  *//消息序号，从1开始，逐1递增 ，8字节*}  
  
type Body struct {  
 XMLName xml.Name `xml:"Body"`  
 ContentType int `xml:",attr"` *//始终为1* ClearTargetDate time.Time *//日期 如：2017-06-05 清分目标日期* ServiceProviderId string *//通行宝中心系统Id，表示消息包中的交易是由收费方产生的* IssuerId string *//发行服务机构Id， 表示产生交易记录的发行服务机构。* MessageId int64 *//交易消息包Id。* Count int *//本消息包含的记录数量* Amount int *//交易总金额(分)* Transaction []Transaction *//交易原始数据 `xml:"Message>Body>Transaction"`*}  
type Transaction struct {  
 XMLName xml.Name `xml:"Transaction"`  
 TransId int  *// 包内顺序Id，从1开始递增 ，包内唯一的交易记录* Time time.Time  *//交易的发生时间，需要加TAC计算* Fee int  *//交易的发生金额(分)* Service Service  *//服务信息* ICCard ICCard  *//IC卡信息* Validation Validation  *//与校验相关的信息* OBU OBU  *//参加交易的电子标签信息* CustomizedData string  *//特定发行方与通行宝收费方之间 约定格式的交易信息【 】*

Id string *//停车场消费交易编号(停车场编号+交易发生的时间+流水号 )*Name string `xml:"name"` *//停车场名称(不超过150个字符)*ParkTime int `xml:"parkTime"` *//停放时长(单位：分)*VehicleType int `xml:"vehicleType"` *//收费车型*AlgorithmIdentifier int `xml:"algorithmIdentifier"` *//算法标识 1-3DEX 2-SM4*

}  
*//服务信息*type Service struct {  
 XMLName xml.Name `xml:"Service"`  
 ServiceType int *//交易的服务类型【】* Description string *//对交易的文字解释【数据库中账单描述】* Detail string *//交易详细信息*}  
*//IC卡信息*type ICCard struct {  
 XMLName xml.Name `xml:"ICCard"`  
 CardType int *//卡类型，22为储值卡；23记账卡* NetNo string *//网络编码，BCD码 Hex(4) k网络号（16进制）数据库10进制* CardId string *//IC卡物理编号，BCD码 Hex(16) 卡号* License string *//0015文件中记录的车牌号* PreBalance string *//交易前余额，以元为单位 Decimal* PostBalance string *//交易后余额，以元为单位 Decimal*}  
  
*//主要用于TAC计算*type Validation struct {  
 XMLName xml.Name `xml:"Validation"`  
 TAC string *//交易时产生的TAC码，8位16进制数 Hex(8)* TransType string *//交易标识，2位16进制数，PBOC定义，如06为传统交易，09为复合交易 Hex(2)* TerminalNo string *//12位16进制数据，即PSAM号，PSAM中0016文件中的终端机编号 Hex(2)* TerminalTransNo string *//8位16进制数，PSAM卡脱机交易序号，在MAC1计算过程中得到 Hex(8)*}  
  
type OBU struct {  
 XMLName xml.Name `xml:"OBU"`  
 NetNo string *//4501 OBU网络号* OBUId string *//OBU物理编号，BCD码 4501191509252866* OBEState string *//2字节的OBU状态* License string *//OBU中记录的车牌号*}

## 2.2线程2

从文件夹sendzipxml中 定期 扫描打包文件，  
 若有则读取文件，并按照联网规范 发送至 联网中心的数据服务端口中；  
 发送成功后（收到联网中心的ok 应答） 将文件转移至另外文件夹sendsucceed；

具体步骤：

定时器定期扫描sendzipxml文件

读取文件

准备报文

发送报文

发送成功则mv消息包至sendsucceed

发送失败 触发重发机制（）

tcp通讯报文格式：

type SendStru struct {  
 Massageid [20]byte *//消息报文序号 20字节Char型，不足左补0* Xml\_length [6]byte *//压缩后XML消息包长度* Md5\_str [32]byte *//32字节MD5校验串 MD5值由<Message>* Xml\_msg []byte *//二进制压缩后的XML 消息包 .xml.lz77*}  
  
tcp通讯报文应答格式：type ReplyStru struct {  
 Massageid [20]byte *//消息报文序号【20字节】* result byte *//【1字节】 1 成功接收 0 接收超时、长度不符*}

## 2.3线程3

服务端监听联网中心的端口，接收联网发送过来的数据包，并存储在指定文件夹receive；  
【 服务端监听端口，接收各种数据包，把数据写成文件，存放入指定文件夹 receive 】

具体步骤：

监听联网中心端口

接收数据包

存储数据包至receive文件夹

## 2.4线程4

定期扫描 接收联网的接收数据的文件夹 receive，如果有文件就解压， 解压后分析包。  
 1、更新包状态，更新应答包状态、争议包状态、记账包状态、清分包状态、退费包状态；

2、处理包信息【】

具体步骤：

定时器定期扫描receive文件夹

读取文件

解析文件

应答数据包

记账数据包

争议数据包

清分数据包

退费数据包

处理文件【待解决】

# 3、数据层处理的表

## 3.1结算数据表

CREATE TABLE `b\_js\_jiessj` (

`F\_VC\_JIAOYJLID` varchar(128) NOT NULL COMMENT '交易记录ID',

`F\_VC\_TINGCCBH` varchar(32) NOT NULL COMMENT '停车场编号',

`F\_VC\_CHEDID` varchar(32) NOT NULL COMMENT '车道ID',

`F\_VC\_GONGSJTID` varchar(32) NOT NULL COMMENT '公司/集团ID',

`F\_NB\_TINGCCLX` int(11) NOT NULL DEFAULT '1' COMMENT '停车场类型 1：单点；2：总对总；',

`F\_NB\_YUANSJYBXH` bigint(20) DEFAULT NULL COMMENT '原始交易包序号 交易状态为“已打包”时有效',

`F\_NB\_JIAOYBNXH` int(11) DEFAULT NULL COMMENT '交易包内序号 交易状态为“已打包”时有效',

`F\_NB\_JIZJG` int(11) NOT NULL DEFAULT '0' COMMENT '记账结果 0：未记账、1：已记账、2：争议数据',

`F\_NB\_ZHENGYLX` int(11) NOT NULL DEFAULT '0' COMMENT '争议类型 0：不是争议，1：验证未通过（如：TAC错误）、2：重复的交易信息、3 ：用户状态变化、4 ：无效交易类型、5 ：逾期超过设定值、6：交易数据域错、7：超过最大交易限额、8：卡号不存在、9：卡状态不匹配、10：卡超过有效期、11：不允许的交易、12：卡片CSN不匹配、13：测试交易、14：卡帐不符（仅用于储值卡）、15：无效卡类型、16：车道对时错误、17：OBU号不存在、100：其它',

`F\_NB\_JIZBXH` int(11) DEFAULT NULL COMMENT '记账包序号 记账结果为非0时有效',

`F\_NB\_ZHENGYCLBXH` bigint(20) DEFAULT NULL COMMENT '争议处理包序号 记账结果为争议放过、坏账时',

`F\_NB\_QINGFBXH` bigint(20) DEFAULT NULL COMMENT '清分包序号',

`F\_VC\_XIAOFJLBH` varchar(128) DEFAULT NULL COMMENT '消费记录编号 单点、总对总出口数据表中的ID号',

`F\_VC\_JIAMKH` varchar(32) NOT NULL COMMENT '加密卡号',

`F\_VC\_KAJMJYXH` varchar(32) NOT NULL COMMENT '加密卡交易序号',

`F\_VC\_OBUID` varchar(32) NOT NULL COMMENT 'Obuid',

`F\_VC\_OBUFXF` varchar(32) NOT NULL COMMENT 'obu发行方',

`F\_VC\_OBUCP` varchar(32) NOT NULL COMMENT 'obu内车牌',

`F\_VC\_OBUCPYS` varchar(32) NOT NULL COMMENT 'obu车牌颜色',

`F\_VC\_KAH` varchar(32) NOT NULL COMMENT '卡号',

`F\_VC\_KAWLH` varchar(32) NOT NULL COMMENT '卡网络号',

`F\_VC\_KAJYXH` varchar(32) NOT NULL COMMENT '卡交易序号',

`F\_VC\_KAFXF` varchar(32) NOT NULL COMMENT '卡发行方',

`F\_NB\_KALX` int(11) DEFAULT NULL COMMENT '卡类型',储值卡22，23 *记账卡*

`F\_NB\_JIAOYQYE` bigint(20) NOT NULL COMMENT '交易前余额',分

`F\_NB\_JIAOYHYE` bigint(20) NOT NULL COMMENT '交易后余额',分

`F\_NB\_JINE` int(11) NOT NULL COMMENT '金额',

`F\_VC\_TACM` varchar(32) NOT NULL COMMENT 'TAC码',

`F\_DT\_JIAOYSJ` datetime NOT NULL COMMENT '交易时间',

`F\_DT\_JIAOYLX` varchar(32) NOT NULL COMMENT '交易类型',

`F\_VC\_CHEX` varchar(32) NOT NULL COMMENT '车型',

`F\_VC\_OBUZT` varchar(32) NOT NULL COMMENT 'OBu状态',

`F\_VC\_SUANFBS` varchar(32) NOT NULL COMMENT '算法标识',

`F\_DT\_YONGHRKSJ` datetime DEFAULT NULL COMMENT '用户入口时间',

`F\_NB\_YONGHTCSC` int(11) DEFAULT NULL COMMENT '用户停车时长(分)',

`F\_VC\_ZHANGDMS` varchar(32) NOT NULL COMMENT '账单描述（给用户通知的信息）',

`F\_VC\_MIYBBH` varchar(32) DEFAULT NULL COMMENT '密钥版本号',

`F\_VC\_OBUYYXLH` varchar(32) NOT NULL COMMENT 'obu应用序列号',

`F\_VC\_CHEDJYXH` varchar(32) NOT NULL COMMENT '车道交易序号',

`F\_NB\_QINGFJG` int(11) DEFAULT NULL COMMENT '清分结果 0：未清分、1：已清分',

`F\_VC\_CHEPH` varchar(32) NOT NULL COMMENT '卡内车牌号',

`F\_NB\_DABZT` int(11) NOT NULL DEFAULT '0' COMMENT '打包状态 0：初始、1：打包中、2：已打包',

`F\_NB\_ZHENGYCLJG` int(11) DEFAULT '0' COMMENT '争议处理结果 0:未处理、1：争议放过、2：坏账',

PRIMARY KEY (`F\_VC\_JIAOYJLID`),

KEY `IDX\_GONGS` (`F\_VC\_GONGSJTID`),

KEY `IDX\_TINGCC` (`F\_VC\_TINGCCBH`),

KEY `IDX\_CHED` (`F\_VC\_CHEDID`),

KEY `IDX\_JIZJG` (`F\_NB\_JIZJG`),

KEY `IDX\_ZHENGYLX` (`F\_NB\_ZHENGYLX`),

KEY `IDX\_DABZT` (`F\_NB\_DABZT`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='结算数据 记录交易处理结果（向上结算）';

## 3.2原始交易

### 3.2.1 原始交易消息表

CREATE TABLE `b\_js\_yuansjyxx` (

`F\_VC\_BANBH` varchar(32) DEFAULT NULL COMMENT '版本号',

`F\_NB\_XIAOXLB` int(11) DEFAULT NULL COMMENT '消息类别 MessageClass',

`F\_NB\_XIAOXLX` int(11) DEFAULT NULL COMMENT '消息类型 MessageType',

`F\_VC\_FASZID` varchar(32) DEFAULT NULL COMMENT '发送者ID',

`F\_VC\_JIESZID` varchar(32) DEFAULT NULL COMMENT '接收者ID',

`F\_NB\_XIAOXXH` bigint(20) NOT NULL COMMENT '消息序号',

`F\_DT\_DABSJ` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '打包时间',

`F\_NB\_FASZT` int(11) NOT NULL DEFAULT '0' COMMENT '发送状态 0：未发送、1：发送中、2：已发送',

`F\_DT\_FASSJ` datetime DEFAULT NULL COMMENT '发送时间',

`F\_NB\_YINGDZT` int(11) NOT NULL DEFAULT '0' COMMENT '应答状态 0：未应答、1：消息已正常接收（用于Advice Response时含已接受建议）、2：消息头错误，如MessageClass或MessageType不符合定义，SenderId不存在等、3：消息格式不正确，即XML Schema验证未通过、4：消息格式正确但内容错误，包括数量不符，内容重复等、5：消息重复、6：消息正常接收，但不接受建议（仅用于Advice Response）、7：消息版本错误',

`F\_VC\_QINGFMBR` varchar(32) DEFAULT NULL COMMENT '清分目标日 yyyy-MM-dd',

`F\_VC\_TINGCCQFFID` varchar(32) DEFAULT NULL COMMENT '停车场清分方ID',

`F\_VC\_FAXFWJGID` varchar(32) DEFAULT NULL COMMENT '发行服务机构ID',

`F\_NB\_JILSL` int(11) DEFAULT NULL COMMENT '记录数量',

`F\_NB\_ZONGJE` int(11) DEFAULT NULL COMMENT '总金额',

`F\_VC\_XIAOXWJLJ` varchar(512) DEFAULT NULL COMMENT '消息文件路径',

PRIMARY KEY (`F\_NB\_XIAOXXH`),

KEY `IDX\_FASZT` (`F\_NB\_FASZT`),

KEY `IDX\_YINGDZT` (`F\_NB\_YINGDZT`),

KEY `IDX\_DABSJ` (`F\_DT\_DABSJ`),

KEY `IDX\_FASSJ` (`F\_DT\_FASSJ`),

KEY `IDX\_QINGFMBR` (`F\_VC\_QINGFMBR`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='原始交易消息 向上结算';

### 3.2.2 原始交易应答消息表

CREATE TABLE `b\_js\_yuansjyydxx` (

`F\_VC\_BANBH` varchar(32) DEFAULT NULL COMMENT '版本号',

`F\_NB\_XIAOXLB` int(11) DEFAULT NULL COMMENT '消息类别 MessageClass',

`F\_NB\_XIAOXLX` int(11) DEFAULT NULL COMMENT '消息类型 MessageType',

`F\_VC\_FASZID` varchar(32) DEFAULT NULL COMMENT '发送者ID',

`F\_VC\_JIESZID` varchar(32) DEFAULT NULL COMMENT '接收者ID',

`F\_NB\_XIAOXXH` bigint(20) DEFAULT NULL COMMENT '消息序号',

`F\_NB\_QUERDXXXH` bigint(20) DEFAULT NULL COMMENT '确认的消息序号',

`F\_DT\_CHULSJ` datetime DEFAULT NULL COMMENT '处理时间',

`F\_NB\_ZHIXJG` int(11) DEFAULT NULL COMMENT '执行结果 1：消息已正常接收（用于Advice Response时含已接受建议）、2：消息头错误，如MessageClass或MessageType不符合定义，SenderId不存在等、3：消息格式不正确，即XML Schema验证未通过、4：消息格式正确但内容错误，包括数量不符，内容重复等、5：消息重复、6：消息正常接收，但不接受建议（仅用于Advice Response）、7：消息版本错误',

`F\_VC\_QINGFMBR` varchar(32) DEFAULT NULL COMMENT '清分目标日 yyyy-MM-dd',

`F\_VC\_XIAOXWJLJ` varchar(512) DEFAULT NULL COMMENT '消息文件路径',

KEY `IDX\_XIAOXXH` (`F\_NB\_XIAOXXH`),

KEY `IDX\_QUERDXXXH` (`F\_NB\_QUERDXXXH`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='原始交易应答消息 向上结算';

### 3.2.3 原始交易明细表

CREATE TABLE `b\_js\_yuansjymx` (

`F\_VC\_XIAOXXH` bigint(20) DEFAULT NULL COMMENT '消息序号',

`F\_NB\_BAONXH` int(11) DEFAULT NULL COMMENT '包内序号',

`F\_DT\_JIAOYSJ` datetime DEFAULT NULL COMMENT '交易时间',

`F\_NB\_JINE` int(11) DEFAULT NULL COMMENT '金额',

`F\_VC\_DINGZJYXX` varchar(512) DEFAULT NULL COMMENT '定制交易信息 CustomizedData',

`F\_VC\_JIAOYBH` varchar(128) DEFAULT NULL COMMENT '交易编号 停车场编号+交易发生时间+流水号',

`F\_VC\_TINGCCMC` varchar(256) DEFAULT NULL COMMENT '停车场名称',

`F\_NB\_TINGFSC` int(11) DEFAULT NULL COMMENT '停放时长 分',

`F\_NB\_SHOUFCX` int(11) DEFAULT NULL COMMENT '收费车型 1：车型一、2：车型二、3：车型三、4：车型四、5：车型五、6：车型六、7~10：自定义',

`F\_NB\_SUANFBS` int(11) DEFAULT NULL COMMENT '算法标识 1：3DEX、2：SM4',

`F\_NB\_FUWLX` int(11) DEFAULT NULL COMMENT '服务类型',

`F\_VC\_ZHANGDSM` varchar(256) DEFAULT NULL COMMENT '账单说明',

`F\_VC\_JIAOYXXXX` varchar(512) DEFAULT NULL COMMENT '交易详细信息',

`F\_NB\_KALX` int(11) DEFAULT NULL COMMENT '卡类型 22：储值卡、23：记账卡',

`F\_VC\_WANGLBM` varchar(32) DEFAULT NULL COMMENT '网络编码 BCD码',

`F\_VC\_KAWLBH` varchar(32) DEFAULT NULL COMMENT '卡物理编号 BCD码',

`F\_VC\_KANCPH` varchar(32) DEFAULT NULL COMMENT '卡内车牌号 0015文件中记录的车牌号',

`F\_VC\_KAJYXH` varchar(32) DEFAULT NULL COMMENT '卡交易序号',

`F\_NB\_JIAOYQYE` bigint(20) DEFAULT NULL COMMENT '交易前余额',

`F\_NB\_JIAOYHYE` bigint(20) DEFAULT NULL COMMENT '交易后余额',

`F\_VC\_TACM` varchar(32) DEFAULT NULL COMMENT 'TAC码',

`F\_VC\_JIAOYBS` varchar(32) DEFAULT NULL COMMENT '交易标识 2位16进制数，PBOC定义，如06为传统交易，09为复合交易',

`F\_VC\_ZONGDJH` varchar(32) DEFAULT NULL COMMENT '终端机号 即PSAM号，PSAM中0016文件中的终端机编号',

`F\_VC\_ZONGDJYXH` varchar(32) DEFAULT NULL COMMENT '终端交易序号 PSAM卡脱机交易序号',

`F\_VC\_OBUWLBH` varchar(32) DEFAULT NULL COMMENT 'OBU物理编号',

`F\_VC\_OBUZT` varchar(32) DEFAULT NULL COMMENT 'OBU状态',

`F\_VC\_OBUNCPH` varchar(32) DEFAULT NULL COMMENT 'OBU内车牌号',

KEY `IDX\_XIAOXXH` (`F\_VC\_XIAOXXH`),

KEY `IDX\_BAONXH` (`F\_NB\_BAONXH`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='原始交易明细 向上结算';

## 3.3记账处理

### 3.3.1 记账处理消息

CREATE TABLE `b\_js\_jizclxx` (

`F\_VC\_BANBH` varchar(32) DEFAULT NULL COMMENT '版本号',

`F\_NB\_XIAOXLB` int(11) DEFAULT NULL COMMENT '消息类别 MessageClass',

`F\_NB\_XIAOXLX` int(11) DEFAULT NULL COMMENT '消息类型 MessageType',

`F\_VC\_FASZID` varchar(32) DEFAULT NULL COMMENT '发送者ID',

`F\_VC\_JIESZID` varchar(32) DEFAULT NULL COMMENT '接收者ID',

`F\_NB\_XIAOXXH` bigint(20) DEFAULT NULL COMMENT '消息序号',

`F\_DT\_JIESSJ` datetime DEFAULT NULL COMMENT '接收时间 接收到消息的时间',

`F\_NB\_YUANSJYXXXH` bigint(20) DEFAULT NULL COMMENT '原始交易消息序号',

`F\_NB\_JILSL` int(11) DEFAULT NULL COMMENT '记录数量',

`F\_NB\_ZONGJE` int(11) DEFAULT NULL COMMENT '总金额',

`F\_NB\_ZHENGYSL` int(11) DEFAULT NULL COMMENT '争议数量',

`F\_NB\_ZHIXJG` int(11) DEFAULT NULL COMMENT '执行结果 1：消息已正常接收（用于Advice Response时含已接受建议）、2：消息头错误，如MessageClass或MessageType不符合定义，SenderId不存在等、3：消息格式不正确，即XML Schema验证未通过、4：消息格式正确但内容错误，包括数量不符，内容重复等、5：消息重复、6：消息正常接收，但不接受建议（仅用于Advice Response）、7：消息版本错误',

`F\_DT\_CHULSJ` datetime DEFAULT NULL COMMENT '处理时间',

`F\_VC\_XIAOXWJLJ` varchar(512) DEFAULT NULL COMMENT '消息文件路径',

KEY `IDX\_XIAOXXXH` (`F\_NB\_XIAOXXH`),

KEY `IDX\_YUANSJYXXXH` (`F\_NB\_YUANSJYXXXH`),

KEY `IDX\_ZHIXJG` (`F\_NB\_ZHIXJG`),

KEY `IDX\_CHULSJ` (`F\_DT\_CHULSJ`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='记账处理消息 向上结算';

### 3.3.2 记账处理应答消息

CREATE TABLE `b\_js\_jizclydxx` (

`F\_VC\_BANBH` varchar(32) DEFAULT NULL COMMENT '版本号',

`F\_NB\_XIAOXLB` int(11) DEFAULT NULL COMMENT '消息类别 MessageClass',

`F\_NB\_XIAOXLX` int(11) DEFAULT NULL COMMENT '消息类型 MessageType',

`F\_VC\_FASZID` varchar(32) DEFAULT NULL COMMENT '发送者ID',

`F\_VC\_JIESZID` varchar(32) DEFAULT NULL COMMENT '接收者ID',

`F\_NB\_XIAOXXH` bigint(20) DEFAULT NULL COMMENT '消息序号',

`F\_NB\_QUERDXXXH` bigint(20) DEFAULT NULL COMMENT '确认的消息序号',

`F\_DT\_CHULSJ` datetime DEFAULT NULL COMMENT '处理时间',

`F\_NB\_ZHIXJG` int(11) DEFAULT NULL COMMENT '执行结果 1：消息已正常接收（用于Advice Response时含已接受建议）、2：消息头错误，如MessageClass或MessageType不符合定义，SenderId不存在等、3：消息格式不正确，即XML Schema验证未通过、4：消息格式正确但内容错误，包括数量不符，内容重复等、5：消息重复、6：消息正常接收，但不接受建议（仅用于Advice Response）、7：消息版本错误',

`F\_VC\_XIAOXWJLJ` varchar(512) DEFAULT NULL COMMENT '消息文件路径',

KEY `IDX\_XIAOXXH` (`F\_NB\_XIAOXXH`),

KEY `IDX\_QUERDXXXH` (`F\_NB\_QUERDXXXH`),

KEY `IDX\_CHULSJ` (`F\_DT\_CHULSJ`),

KEY `IDX\_ZHIXJG` (`F\_NB\_ZHIXJG`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='记账处理应答消息 向上结算';

### 3.3.3 记账处理明细

CREATE TABLE `b\_js\_jizclmx` (

`F\_NB\_YUANSJYXXXH` bigint(20) DEFAULT NULL COMMENT '原始交易消息序号',

`F\_NB\_BAONXH` int(11) DEFAULT NULL COMMENT '包内序号',

`F\_NB\_CHULJG` int(11) DEFAULT NULL COMMENT '处理结果 1：验证未通过（如：TAC错误）、2：重复的交易信息、3：用户状态变化、4：无效交易类型、5：逾期超过设定值、6：交易数据域错、7：超过最大交易限额、8：卡号不存在、9：卡状态不匹配、10：卡超过有效期、11：不允许的交易、12：卡片CSN不匹配、13：测试交易、14：卡帐不符（仅用于储值卡）、15：无效卡类型、16：车道对时错误、17：OBU号不存在、100：其它',

KEY `IDX\_YUANSJYXXXH` (`F\_NB\_YUANSJYXXXH`),

KEY `IDX\_BAONXH` (`F\_NB\_BAONXH`),

KEY `IDX\_CHULJG` (`F\_NB\_CHULJG`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='记账处理明细 向上结算';

## 3.4争议处理

### 3.4.1 争议交易处理消息

CREATE TABLE `b\_js\_zhengyjyclxx` (

`F\_VC\_BANBH` varchar(32) DEFAULT NULL COMMENT '版本号',

`F\_NB\_XIAOXLB` int(11) DEFAULT NULL COMMENT '消息类别 MessageClass',

`F\_NB\_XIAOXLX` int(11) DEFAULT NULL COMMENT '消息类型 MessageType',

`F\_VC\_FASZID` varchar(32) DEFAULT NULL COMMENT '发送者ID',

`F\_VC\_JIESZID` varchar(32) DEFAULT NULL COMMENT '接收者ID',

`F\_NB\_XIAOXXH` bigint(20) DEFAULT NULL COMMENT '消息序号',

`F\_DT\_JIESSJ` datetime DEFAULT NULL COMMENT '接收时间 接收到消息的时间',

`F\_VC\_QINGFFID` varchar(32) DEFAULT NULL COMMENT '清分方ID',

`F\_VC\_LIANWZXID` varchar(32) DEFAULT NULL COMMENT '联网中心ID',

`F\_VC\_FAXFID` varchar(32) DEFAULT NULL COMMENT '发行方ID',

`F\_VC\_ZHENGYJGWJID` int(11) DEFAULT NULL COMMENT '争议结果文件ID',

`F\_DT\_ZHENGYCLSJ` datetime DEFAULT NULL COMMENT '争议处理时间',

`F\_NB\_ZHENGYSL` int(11) DEFAULT NULL COMMENT '争议数量',

`F\_NB\_QUERXYJZDZJE` int(11) DEFAULT NULL COMMENT '确认需要记账的总金额',

`F\_VC\_XIAOXWJLJ` varchar(512) DEFAULT NULL COMMENT '消息文件路径',

`F\_NB\_ZHIXJG` int(11) DEFAULT NULL COMMENT '执行结果 1：消息已正常接收（用于Advice Response时含已接受建议）、2：消息头错误，如MessageClass或MessageType不符合定义，SenderId不存在等、3：消息格式不正确，即XML Schema验证未通过、4：消息格式正确但内容错误，包括数量不符，内容重复等、5：消息重复、6：消息正常接收，但不接受建议（仅用于Advice Response）、7：消息版本错误',

KEY `IDX\_XIAOXXH` (`F\_NB\_XIAOXXH`),

KEY `IDX\_JIESSJ` (`F\_DT\_JIESSJ`),

KEY `IDX\_ZHENGYCLSJ` (`F\_DT\_ZHENGYCLSJ`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='争议交易处理消息 向上结算';

### 3.4.2 争议交易处理应答消息

CREATE TABLE `b\_js\_zhengyjyclydxx` (

`F\_VC\_BANBH` varchar(32) DEFAULT NULL COMMENT '版本号',

`F\_NB\_XIAOXLB` int(11) DEFAULT NULL COMMENT '消息类别 MessageClass',

`F\_NB\_XIAOXLX` int(11) DEFAULT NULL COMMENT '消息类型 MessageType',

`F\_VC\_FASZID` varchar(32) DEFAULT NULL COMMENT '发送者ID',

`F\_VC\_JIESZID` varchar(32) DEFAULT NULL COMMENT '接收者ID',

`F\_NB\_XIAOXXH` bigint(20) DEFAULT NULL COMMENT '消息序号',

`F\_NB\_QUERDXXXH` bigint(20) DEFAULT NULL COMMENT '确认的消息序号',

`F\_DT\_CHULSJ` datetime DEFAULT NULL COMMENT '处理时间',

`F\_NB\_ZHIXJG` int(11) DEFAULT NULL COMMENT '执行结果 1：消息已正常接收（用于Advice Response时含已接受建议）、2：消息头错误，如MessageClass或MessageType不符合定义，SenderId不存在等、3：消息格式不正确，即XML Schema验证未通过、4：消息格式正确但内容错误，包括数量不符，内容重复等、5：消息重复、6：消息正常接收，但不接受建议（仅用于Advice Response）、7：消息版本错误',

FVcXiaoxwjlj string *//F\_VC\_XIAOXWJLJ 消息文件路径 VARCHAR(512)*

KEY `IDX\_XIAOXXH` (`F\_NB\_XIAOXXH`),

KEY `IDX\_QUERDXXXH` (`F\_NB\_QUERDXXXH`),

KEY `IDX\_CHULSJ` (`F\_DT\_CHULSJ`),

KEY `IDX\_ZHIXJG` (`F\_NB\_ZHIXJG`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='争议交易处理应答消息 向上结算';

### 3.4.3 争议交易处理明细

CREATE TABLE `b\_js\_zhengyjyclmx` (

`F\_NB\_ZHENGYJYCLXXXH` bigint(20) DEFAULT NULL COMMENT '争议交易处理消息序号',

`F\_NB\_FENZXH` int(11) DEFAULT NULL COMMENT '分组序号 入库者自行生成，可取数组下标',

`F\_NB\_YUANSJYXXXH` bigint(20) DEFAULT NULL COMMENT '原始交易消息序号',

`F\_NB\_ZUNJLSL` int(11) DEFAULT NULL COMMENT '组内记录数量',

`F\_NB\_ZUNJEZH` int(11) DEFAULT NULL COMMENT '组内金额总和',

`F\_NB\_YUANSBNXH` int(11) DEFAULT NULL COMMENT '原始包内序号',

`F\_NB\_CHULJG` int(11) DEFAULT NULL COMMENT '处理结果 0：表示正常支付、1：表示此交易作坏账处理。',

KEY `IDX\_ZHENGYJYCLXXXH` (`F\_NB\_ZHENGYJYCLXXXH`),

KEY `IDX\_FENZXH` (`F\_NB\_FENZXH`),

KEY `IDX\_YUANSJYXXXH` (`F\_NB\_YUANSJYXXXH`),

KEY `IDX\_CHULJG` (`F\_NB\_CHULJG`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='争议交易处理明细 向上结算';

## 3.5清分统计处理

### 3.5.1 清分统计消息

CREATE TABLE `b\_js\_qingftjxx` (

`F\_VC\_BANBH` varchar(32) DEFAULT NULL COMMENT '版本号',

`F\_NB\_XIAOXLB` int(11) DEFAULT NULL COMMENT '消息类别 MessageClass',

`F\_NB\_XIAOXLX` int(11) DEFAULT NULL COMMENT '消息类型 MessageType',

`F\_VC\_FASZID` varchar(32) DEFAULT NULL COMMENT '发送者ID',

`F\_VC\_JIESZID` varchar(32) DEFAULT NULL COMMENT '接收者ID',

`F\_NB\_XIAOXXH` bigint(20) DEFAULT NULL COMMENT '消息序号',

`F\_DT\_JIESSJ` datetime DEFAULT NULL COMMENT '接收时间 接收到消息的时间',

`F\_VC\_QINGFMBR` date DEFAULT NULL COMMENT '清分目标日 yyyy-MM-dd',

`F\_NB\_QINGFZJE` int(11) DEFAULT NULL COMMENT '清分总金额',

`F\_NB\_QINGFSL` int(11) DEFAULT NULL COMMENT '清分数量',

`F\_DT\_QINGFTJCLSJ` datetime DEFAULT NULL COMMENT '清分统计处理时间',

`F\_NB\_YUANSJYSL` int(11) DEFAULT NULL COMMENT '原始包交易数量',

`F\_NB\_ZHENGYCLJGBSL` int(11) DEFAULT NULL COMMENT '争议处理结果包数量',

`F\_VC\_XIAOXWJLJ` varchar(512) DEFAULT NULL COMMENT '消息文件路径',

`F\_DT\_CHULSJ` datetime DEFAULT NULL COMMENT '处理时间',

KEY `IDX\_XIAOXXXH` (`F\_NB\_XIAOXXH`),

KEY `IDX\_JIESSJ` (`F\_DT\_JIESSJ`),

KEY `IDX\_QINGFTJCLSJ` (`F\_DT\_QINGFTJCLSJ`),

KEY `IDX\_QINGFMBR` (`F\_VC\_QINGFMBR`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='清分统计消息 向上结算';

### 3.5.2 清分统计应答消息

CREATE TABLE `b\_js\_qingftjydxx` (

`F\_VC\_BANBH` varchar(32) DEFAULT NULL COMMENT '版本号',

`F\_NB\_XIAOXLB` int(11) DEFAULT NULL COMMENT '消息类别 MessageClass',

`F\_NB\_XIAOXLX` int(11) DEFAULT NULL COMMENT '消息类型 MessageType',

`F\_VC\_FASZID` varchar(32) DEFAULT NULL COMMENT '发送者ID',

`F\_VC\_JIESZID` varchar(32) DEFAULT NULL COMMENT '接收者ID',

`F\_NB\_XIAOXXH` bigint(20) DEFAULT NULL COMMENT '消息序号',

`F\_NB\_QUERDXXXH` bigint(20) DEFAULT NULL COMMENT '确认的消息序号',

`F\_DT\_CHULSJ` datetime DEFAULT NULL COMMENT '处理时间',

`F\_NB\_ZHIXJG` int(11) DEFAULT NULL COMMENT '执行结果 1：消息已正常接收（用于Advice Response时含已接受建议）、2：消息头错误，如MessageClass或MessageType不符合定义，SenderId不存在等、3：消息格式不正确，即XML Schema验证未通过、4：消息格式正确但内容错误，包括数量不符，内容重复等、5：消息重复、6：消息正常接收，但不接受建议（仅用于Advice Response）、7：消息版本错误',

`F\_VC\_XIAOXWJLJ` varchar(512) DEFAULT NULL COMMENT '消息文件路径',

KEY `IDX\_XIAOXXH` (`F\_NB\_XIAOXXH`),

KEY `IDX\_CHULSJ` (`F\_DT\_CHULSJ`),

KEY `IDX\_QUERDXXXH` (`F\_NB\_QUERDXXXH`),

KEY `IDX\_ZHIXJG` (`F\_NB\_ZHIXJG`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='清分统计消息应答 向上结算';

### 3.5.3 清分统计明细

CREATE TABLE `b\_js\_qingftjmx` (

`F\_NB\_QINGFTJXXXH` bigint(20) DEFAULT NULL COMMENT '清分统计消息序号',

`F\_NB\_FENZXH` int(11) DEFAULT NULL COMMENT '分组序号 入库者自行生成，可取数组下标',

`F\_VC\_TONGXBZXXTID` varchar(32) DEFAULT NULL COMMENT '通行宝中心系统ID',

`F\_NB\_YUANSJYXXXH` bigint(20) DEFAULT NULL COMMENT '原始交易消息序号',

`F\_NB\_ZHENGYCLJGWJID` int(11) DEFAULT NULL COMMENT '争议处理结果文件ID',

KEY `IDX\_QINGFTJXXXH` (`F\_NB\_QINGFTJXXXH`),

KEY `IDX\_FENZXH` (`F\_NB\_FENZXH`),

KEY `IDX\_YUANSJYXH` (`F\_NB\_YUANSJYXXXH`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='清分统计明细 向上结算';

# 4、数据层逻辑

连接数据库

查询数据

插入数据

# 五、具体接口规范

## 5.1传输规则

## 传输方式

通行宝与联网中心结算数据通讯，采用TCP/IP 异步短连接方式，各方的server端和Client端 独立。

客户端Client进程负责向对方发送消息，服务端Server进程负责接收 处理对方发送的消息。

客户端发送前，主动向服务端建立tcp/ip socket连接，连接建立后，发送消息报文，服务端接收消息，根据报文长度判断接收完毕后，应答客户端通讯接收ok，客户端接收应答后，主动关闭socket。

## 数据流程

## 基本结构

### TCP通讯报文格式

格式如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **MESSAGEID**  **消息报文序号** | **压缩后XML消息包长度** | **32字节MD5校验串** | **压缩后的XML**  **消息包** |
| Char （20） | Char（6） | Char(32) | Char ( 不定长) |
| SSSSSSSSSSSSSSSSSSS | CCCCCC | HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH | AAAAAAAAAAAAAAA…… |

即： 消息报文顺序号 （SSSSSSS） 20字节 + 压缩后的XML消息长度(CCCCCC) 6字节 + 32字节MD5 (16进制显示字符串) + 二进制压缩后的XML消息包( AAAAAAAAAAAAAAA…)

其中：MESSAGEID长度20字节Char型，不足左补0，取值与XML消息包消息头中的MessageId 栏位相同。

压缩后的xml消息长度类型为6字节Char型压缩后的消息包长度，不足左补0。

MD5为32个16进制显示字符，不足左补0。

压缩后的xml报文最大长度999999字节。

通讯报文格式定义如下：

1. type SendStru struct {  
    Massageid [20]byte *//消息报文序号 20字节Char型，不足左补0* Xml\_length [6]byte *//压缩后XML消息包长度* Md5\_str [32]byte *//32字节MD5校验串 MD5值由<Message>* Xml\_msg []byte *//二进制压缩后的XML 消息包 .xml.lz77*}

### 报文接收处理

对于大报文（长度超过16K），接收方获得报文长度后，需进行多次Read，直至接收到所有报文数据后，应答接收OK报文。如 中间 报文接收超时，应答接收失败报文，关闭Socket退出接收处理。

### MD5计算方法

MD5值由<Message> …</Message> XML报文压缩前明文的二进制流进行计算获得。

### 压缩算法：

XML消息包的压缩算法采用标准的LZ77压缩算法。

### 通讯应答接收完成报文格式

MessageId + tcp接收结果

20字节char型 + 1字节char型

该应答用于通讯接收完成的即时应答，不校验消息内部信息格式，判断消息接收长度符合，即返回此应答给发送方。发送方依此应答作为报文发送结束标志，不作为消息内容正确的判断依据，消息内容的正确性判断，由接收方业务处理后通过通用确认消息格式异步返回。

|  |  |
| --- | --- |
| Tcp接收结果  Result | 取值 |
| 1 | 成功接收 |
| 0 | 接收超时长度不符 |

通讯报文应答格式定义如下：

type ReplyStru struct {

Massageid [20]byte *//消息报文序号【20字节】* result byte *//【1字节】 1 成功接收 0 接收超时、长度不符*

}

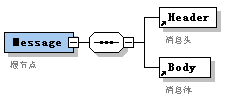
异步短连接通讯处理时序模型图说明：



XML消息包参考下述格式定义。

### XML消息包 数据存储形式

所有传输的数据及回应消息均采用XML，其基本结构如下：



所有消息，包括用于确认信息的消息均使用以上基本结构。

消息包含消息头Header和消息体Body。所有消息的消息头结构相同，仅使用的具体数值根据其不同应用有所区别。不同应用的消息体内部结构不同。

若未明确说明，所有整数类型的值均采用十进制，所有表示金额的节点均采用十进制并精确到分，如果表示一百二十三元四角五分为123.45 (报文中定义的为Decimal类型)。

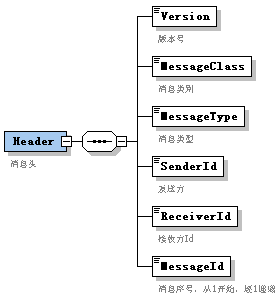
所有数据结构由以Schema形式定义，所有XML数据能够通过对应Schema的合法性验证。XML文件的应用系统应保存XML文件内容必需是符合逻辑的。

### 数据类型

Schema中用于定义XML结构的部分数据类型说明见下表：

|  |  |  |
| --- | --- | --- |
| 数据类型 | 说明 | 示例 |
| Short | 2 字节整数，以 10 进制表示 |  |
| Int | 4 字节整数，以 10 进制表示 |  |
| Long | 8 字节整数，以 10 进制表示 |  |
| Date | 日期 | YYYY-MM-DD，如 2008-01-25 |
| DateTime | 时间，采用 24 小时表示法，以字符“T”作为日 期与时间的分隔符，精确到秒 | YYYY-MM-DDTHH:mm:ss，如  2008-01-25T15:33:46 |
| HexBinary | 在后文定义中简略为 Hex(n)，以 16 进制数字对 的方式表示一串字节数组的内容，高位在前，低 位在后。n 为长度，每两个 16 进制数表示 1 个字 节，所以，n 必定是偶数。不足规定长度的，左 补 0。Schema 定义本身不规定 Hex 的长度（只要 保证是偶数），长度控制由应用程序负责 | 001a345f 表示 0x001a345f。若 使用 01a345f 则在验证 XML 文 件合法性时会产生错误，因为 数字串的长度是 7，不是偶数长 度。 |
| Decimal | 以 10 进制表示的浮点数 | 如 1340.56 等 |
| String | 字符串，为表示长度，在后文定义时使用 String(n) 进行表示。n 为字符串最终存储的最大字节数。 超过定义长度的部分将不被接收方处理。若省略 n，表示不规定字符串长度。 | 汉字字符串字节长度的计算应 以 GB18030 大字符集的编码为 依据，1 个汉字用 2 个字节保存 |

## 消息头



消息头是所有消息均包含的第一个节点，表示消息的身份及用途，数据类型及意义如下：

| 名称 | 数据类型 | 取值及说明 |
| --- | --- | --- |
| Version | Hex(8) | 版本号，按照从高位到低位分解4字节的整数，每两个字节表示一个序号：前两个字节表示主版本号，后两个字节表示次版本号。如0x00010000表示版本1.0。  本次版本号统一使用0x00010000，表示版本1.0 |
| MessageClass | Int | 说明消息传输的机制 |
| MessageType | Int | 说明消息的应用类型 |
| SenderId | Hex(16位，不足补零) | 发送方Id，在整个系统中唯一  0000000000000020: 高速公路联网中心  00000000000000FD; 通行宝 |
| ReceiverId | Hex(16位，不足补零) | 接收方Id，在整个系统中唯一；同上 |
| MessageId | Long | 消息序号，从1开始，逐1递增 ，8字节 |

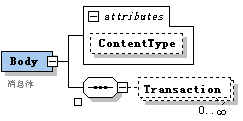
MessageClass以4字节整型表示。

| 名称 | 值 | 说明 |
| --- | --- | --- |
| 请求Request | 1 | 接收方需返回处理结果，可能包含大量数据  请求方式 |
| 请求应答Request Response | 2 |
| 建议Advice | 3 | 接收方需指明是否接受发送方的建议，返回信息简单 |
| 建议应答Advice Response | 4 |
| 通知Notification | 5 | 接收方仅需指明接收是否正确 |
| 通知应答Notification Response | 6 |

MessageType以4字节整型表示。

| 名称 | 值 |
| --- | --- |
| 服务列表Servcie List | 1 |
| 价目表Fare Products List | 2 |
| 用户信息Customer Details | 3 |
| 分账规则Apportionment Rules | 4 |
| 对账总金额Reconciliation Totals | 5 |
| 授权Authorization | 6 |
| 交易Transaction | 7 |
| 报告已发送Report Sent | 8 |
| 密钥管理Key Management | 9 |
| 状态名单Status List | 10 |
| 设备状态Equipment Status | 11 |
| 例外事件Event Exception | 12 |
| 接受付费方式Payment Method Acceptance | 13 |
| 参与方信息Operator List | 14 |
| 公务卡名单Privilege List | 15 |
| 一卡通可用卡类型BMACValidCardType | 16 |
| 标签拆卸 | 17 |
| 未定义的消息类型Undefined Message Type | 18 |
| 灰名单卡内最近历史交易信息History Transaction Detail | 19 |
| 重复收取外地车辆贷款道路通行费交易明细  Loan Refund Detail | 20 |
| 重复收取外地车辆贷款道路通行费交易对帐  Loan Refund Reconciliation | 21 |

## 消息体



消息体包含一个属性ContentType和多个内容对象。

消息头中的MessageClass说明消息传输、应答的方式；MessageType说明消息内容所属应用分类；ContentType说明在MessageType确定的应用中的具体分类。

并不是所有消息体均有ContentType属性。如果某MessageType下仅传递一种信息，则该类消息的消息体可忽略ContentType属性。

## 传输控制

发送方与接收方的数据传输采用一问一答方式。发送方在规定时间60秒内，未接收到接收方的应答需通过自动重发等方式将数据传送到接收方。重发消息、导出消息的MessageId保持不变。

### 通用确认消息结构

#### 应用范围

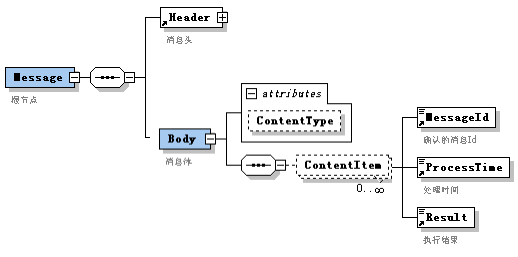
接收方收到发送方的消息后，必需给予发送方回应。不同的MessageClass，MessageType所使用的返回消息结构不尽相同。但如果消息结构不正确（例如MessageClass值未定义）等无法通过校验的情况发生时，接收方需通知发送方消息异常。此时需使用通用确认消息结构。另外，对某些消息的回应相对简单，也使用通用确认消息结构发送。

各消息的详细回应说明请参与相关章节。

#### 消息头

| 名称 | 数据类型 | 取值或说明 |
| --- | --- | --- |
| Version | Hex(8) | 本次版本号统一使用0x00010000，表示版本1.0 |
| MessageClass | Int | 使用所接收消息的MessageClass |
| MessageType | Int | 使用与其对应的Response值 |
| SenderId | Hex(16位，不足补零) | 当前参与方Id |
| ReceiverId | Hex(16位，不足补零) | 准备接收确认消息的参与方Id |
| MessageId | Long | 消息序号，从1开始，逐1递增 ，8字节 |

#### 应答消息内容



Body的ContentType属性是可选的，在消息头MessageClass和MessageType的基础上进一步指出响应的是哪一类消息，与所回应的消息的ContentType保持一致。Body各个子节点说明如下：

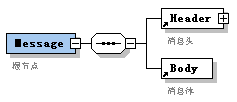
| 名称 | 数据类型 | 取值或说明 |
| --- | --- | --- |
| MessageId | Long | 确认的消息Id （对应于发送方的header里的MessageId） |
| ProcessTime | DateTime | 处理时间 |
| Result | Short | 执行结果：   1. 消息已正常接收（用于Advice Response时含已接受建议） 2. 消息头错误，如MessageClass或MessageType不符合定义，SenderId不存在等 3. 消息格式不正确，即XML Schema验证未通过 4. 消息格式正确但内容错误，包括数量不符，内容重复等 5. 消息重复 6. 消息正常接收，但不接受建议（仅用于Advice Response） 7. 消息版本错误 |

### 通用重发请求消息结构

#### 消息头

| 名称 | 数据类型 | 取值或说明 |
| --- | --- | --- |
| MessageClass | Int | 1，Request |
| MessageType | Int | 请求重发的数据类型对应的MessageType |
| SenderId | Hex(16位，不足补零) | 请求重发消息方Id  00000000000000FD; 通行宝  00000000000000FE;普通公路经营性中心  00000000000000FF; 普通公路中心  0000000000000020: 高速公路联网中心 |
| ReceiverId | Hex(16位，不足补零) | 接收消息的参与方Id  编码同上 |

#### 消息内容



通用重发请求消息中没有更多的数据，其Body为空。

## 参与方ID

通行宝中心ID ,联网中心ID统一使用代码；

如江苏省高速公路联网中心代码为0x20,相应XML报文值十六进制显示值为20

|  |  |
| --- | --- |
| 0000000000000020 | 江苏高速公路联网中心 |
| 00000000000000FF | 普通公路中心 |
| 00000000000000FE; | 普通公路经营性中心 |
| 00000000000000FD | 通行宝中心 |

## 卡ID及卡类型

卡类型分为国标IC卡、一卡通IC卡和电子标签三种。不排除各发行方使用同一个参与方ID发行多种卡的可能。

卡ID在本系统中的唯一性表示为：网络编码 + 卡发行号。

网络编码为4位BCD码，卡发行号为16位BCD码。在XML文件中，均以HexBinary表示。不足位数左侧补0。

电子标签及一卡通没有网络编码，均使用0000表示该字段。

# 外省ETC卡交易处理

## 应用范围

交易处理是收费交易从通行宝中心到联网中心转发，再到外省结算中心记帐、结算处理的过程。

本节说明在整个处理过程中使用的消息结构及处理流程。

所有交易消息的接收方均需通过通用确认消息通知发送方消息接收结果。

## 处理规则

交易记录在整个系统中打包传送。一个交易包中包含多条交易记录。

1. 通行宝中心生成外省ETC卡原始交易包发给联网中心。
2. 联网中心生成记账包发给通行宝。
3. 联网中心生成争议包发给通行宝。
4. 联网中心生成清分包发给通行宝。
5. 联网中心生成退费包发给通行宝。

## 原始交易消息结构

### 应用范围

由通行宝中心系统将交易分组打包后发送给联网中心的原始交易数据。通行宝与联网中心交易数据的发送方向是：通行宝中心系统→联网中心。外省份的数据由联网中心对数据进行转发。

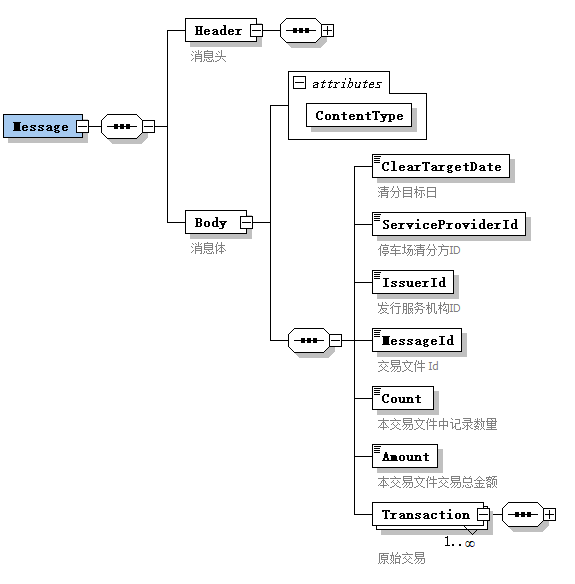
### 请求包格式：

#### 消息头

|  |  |  |
| --- | --- | --- |
| 名称 | 数据类型 | 取值或说明 |
| Version | Hex(8) | 本次版本号统一使用0x00010000，表示版本1.0 |
| MessageClass | Int | 5，Notification |
| MessageType | Int | 7，Transaction |
| SenderId | Hex(16位，不足补零) | 0000000000000020: 高速公路联网中心  00000000000000FE;普通公路经营性中心  00000000000000FF; 普通公路中心  00000000000000FD; 通行宝 |
| ReceiverId | Hex(16位，不足补零) | 编码同上 |
| MessageId | Long | 消息序号，从1开始，逐1递增 ，8字节【int64】 |

通行宝仅作为公路收费方不发行卡,将数据统一发送给高速公路联网中心，由高速公路联网中心结算或转发外地卡。

#### 消息内容

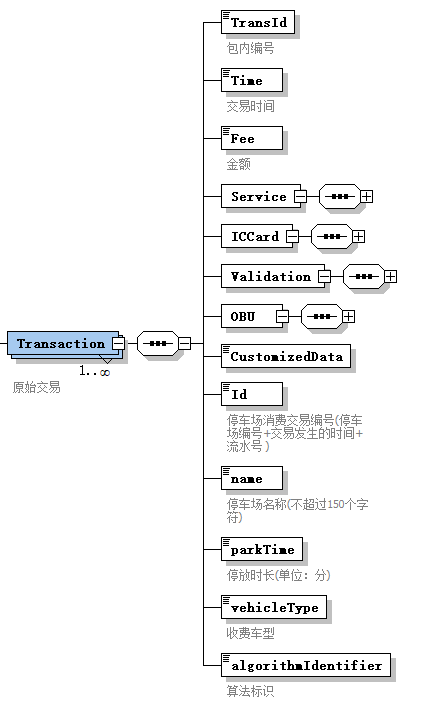


交易信息中，Body的ContentType始终为1，Body各个子节点说明如下：

| 名称 | 数据类型 | 取值及说明 |
| --- | --- | --- |
| ClearTargetDate | 日期 | 如：2017-06-05 |
| ServiceProviderId | String | 通行宝中心系统Id，表示消息包中的交易是由收费方产生的。 |
| IssuerId | String | 发行服务机构Id， 表示产生交易记录的发行服务机构。 |
| MessageId | Long | 交易消息包Id。由通行宝中心系统发送包到联网中心时，该字段与消息头的MessageId相同。联网中心发出的消息此字段不用改变。各参与方可通过ServiceProviderId和MessageId在系统唯一确定一个原始交易信息包。通行宝中心系统消息发送失败重发时，消息头与消息体使用原相同MessageId不变重发。 |
| Count | Int | 本消息包含的记录数量 |
| Amount | Int | 交易总金额(分) |

通行宝中心系统直接将原始交易打包，发送给联网中心。联网中心按交易包中指明的发行方将交易提交给对应对发行方处理。

交易包中包含原始交易记录。交易记录的格式如下：



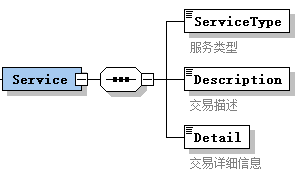
交易记录的属性说明如下：

| 名称 | 数据类型 | 取值及说明 |
| --- | --- | --- |
| TransId | Int | 是由通行宝中心产生的该包内顺序Id，从1开始递增。在通行宝中心、联网中心、发行方三方的交易通讯过程中均采用此Id表示包内唯一的交易记录。通过MessageId与TransId，可以在系统中唯一确定一条交易。 |
| Time | Datetime | 交易的发生时间，需要加TAC计算 |
| Fee | Int | 交易的发生金额(元) |
| Service |  | 服务信息 |
| ICCard |  | IC卡信息 |
| Validation |  | 与校验相关的信息 |
| OBU |  | 参加交易的电子标签信息 |
| CustomizedData | String | 特定发行方与通行宝收费方之间约定格式的交易信息 |
| Id | String | 停车场消费交易编号(停车场编号+交易发生的时间+流水号 ) |
| name | String | 停车场名称(不超过150个字符) |
| parkTime | Int | 停放时长(单位：分) |
| vehicleType | Int | 收费车型 |
| algorithmIdentifier | Int | 算法标识 1-3DEX 2-SM4 |

CustomizedData为特定参与方之间约定的格式，所以此处不定义长度，仅定义数据类型。

为了给用户提供完整的消费清单，即使消费金额为0，也应将交易信息发送给发行方。

服务信息Service的结构如下：



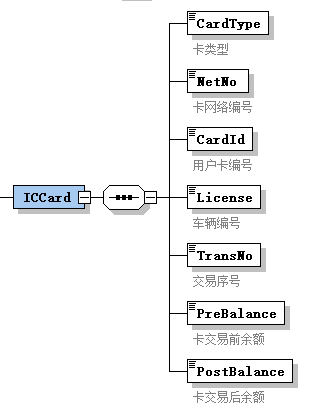
| 名称 | 数据类型 | 取值及说明 |
| --- | --- | --- |
| ServiceType | Short | 交易的服务类型，取值见基础信息维护 |
| Description | String | 对交易的文字解释。  停车场名|停车时长，如<Description>首都机场T3|3天1小时34分</Description>。时间精确到分，如果不满一天，则“天”省略，如果不满1小时，则“天”和“小时”均可省略，最大单位为“天”，不必转换成“年”、“月”、“周”。【数据库存的是分】 |
| Detail | String(500) | 交易详细信息 |

如下：



Detail字段结构：

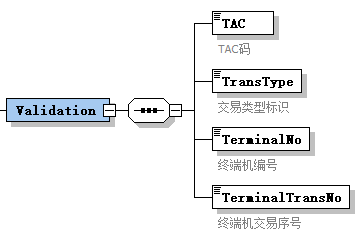
| 名称 | 取值及说明 |
| --- | --- |
| 收费车型 | 1-车型一；2-车型二；3-车型三；4-车型四；5-车型五；6-车型六；7~10：自定义- |
| 出口类型 | 00-保留；02-封闭MTC出口，04-封闭ETC出口；05-MTC开放式；06-ETC开放式；07~0F自定义；10~FF保留给未来使用 |
| 出口路网号 |  |
| 出口站/广场号 |  |
| 出口车道号 | YYYYMMDD HHMMSS |
| 入口类型 | 00-保留，01-封闭MTC入口；03-封闭ETC入口；07~0F自定义；10~FF保留给未来使用 |
| 入口路网号 |  |
| 入口站/广场号 |  |
| 入口车道号 |  |
| 入口时间 |  |



| 名称 | 数据类型 | 取值及说明 |
| --- | --- | --- |
| Cardtype | short | 卡类型，22为储值卡；23记账卡 |
| NetNo | Hex(4) | 网络编码，BCD码 |
| CardID | Hex(16) | IC卡物理编号，BCD码 |
| License | String(25) | 0015文件中记录的车牌号 |
| TransNo | String | 卡交易序号 |
| PreBalance | Decimal | 交易前余额，以元为单位 |
| PostBalance | Decimal | 交易后余额，以元为单位 |

清分方原则上不处理Validation和CustomizedData中的数据。

Validation中的数据主要用于TAC计算，结构如下：

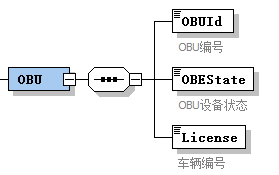


| 名称 | 数据类型 | 取值及说明 |
| --- | --- | --- |
| TAC | Hex(8) | 交易时产生的TAC码，8位16进制数 |
| TransType | Hex(2) | 交易标识，2位16进制数，PBOC定义，如06为传统交易，09为复合交易 |
| TerminalNo | Hex(2) | 12位16进制数据，即PSAM号，PSAM中0016文件中的终端机编号 |
| TerminalTransNo | Hex(8) | 8位16进制数，PSAM卡脱机交易序号，在MAC1计算过程中得到 |

TAC的计算方法参见《中华人民共和国金融行业JR/T 0025-2005中国金融集成电路IC卡规范》。TAC计算所需字段在该规范中定义与本消息结构的关系为：

| 名称 | 数据类型 | 取值及说明 |
| --- | --- | --- |
| 交易金额 | Hex(8) | 8位16进制数，以分为单位，需根据Transaction.Fee转换 |
| 交易类型标识 | Hex(2) | 2位16进制数，取值为TransType |
| 终端机编号 | Hex(12) | 12位16进制，取值为TerminalNo |
| 终端机交易序号 | Hex(8) | 8位16进制数，取值为TerminalTransNo，记账卡不需要此字段 |
| 终端交易日期 | Hex(8) | 8位16进制数，BCD码，YYYYMMDD，根据Transaction.Time转换 |
| 终端交易时间 | Hex(6) | 6位16进制数，BCD码，HHMMSS，根据Transaction.Time转换 |

如果交易是通过ETC车道产生的，需附加电子标签信息，格式如下：



| 名称 | 数据类型 | 取值及说明 |
| --- | --- | --- |
| OBUId | Hex(12) | OBU物理编号，BCD码 |
| OBUState | Hex(4) | 2字节的OBU状态 |
| License | String(25) | OBU中记录的车牌号 |

无论产生ETC交易的IC卡与OBU是否属于同一发行方，均应传送OBU信息。

CustomizedData按照如下格式：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **内部名称** | **Xml 类型** | **Xml宽度** | **备注** |
| 1 | TAC校验码 | TAC | hexbinary | 8 |  |
| 2 | 交易金额 | TxnAmt | hexbinary | 8 |  |
| 3 | 交易类型 | TxnType | hexbinary | 2 |  |
| 4 | 终端机编号 | PosId | hexbinary | 12 |  |
| 5 | 交易的终端流水号 | PosSeq | hexbinary | 8 |  |
| 6 | 交易时间 | TxTime(TxnDate+TxnTime) | hexbinary | 14 | YYYYMMDDHHMISS |
| 7 | TAC保留串 | Reserve | hexbinary | 4 |  |
| 8 | 交易后余额 | Balance | hexbinary | 8 |  |
| 9 | 交易前余额 | PreBalance | hexbinary | 8 |  |
| 10 | 卡计数器 | Iccounter | hexbinary | 4 |  |

发送规则

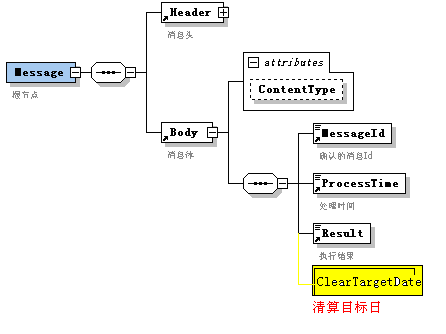
通行宝中心将原始交易记账包发送给联网中心需满足如下两个条件之一：

* 到达预定义的时间间隔（如10分钟、半小时）；

在时间间隔未到达前，积累未发送的记录达到一定数量（如前面定义的1000条）。

### 应答包格式：

#### 应答消息内容



#### 消息头

|  |  |  |
| --- | --- | --- |
| 名称 | 数据类型 | 取值或说明 |
| Version | Hex(8) | 本次版本号统一使用0x00010000，表示版本1.0 |
| MessageClass | Int | 6 |
| MessageType | Int | 7 |
| SenderId | Hex(16位，不足补零) | 当前参与方Id  0000000000000020: 高速公路联网中心  00000000000000FE;普通公路经营性中心  00000000000000FF; 普通公路中心  00000000000000FD; 通行宝 |
| ReceiverId | Hex(16位，不足补零) | 准备接收确认消息的参与方Id  编码同上 |
| MessageId | Long | 消息序号，从1开始，逐1递增 ，8字节 |

Body的Body的ContentType始终为1。

Body各个子节点说明如下：

| 名称 | 数据类型 | 取值或说明 |
| --- | --- | --- |
| MessageId | Long | 确认的消息Id |
| ProcessTime | DateTime | 处理时间 |
| Result | Short | 执行结果：   1. 消息已正常接收（用于Advice Response时含已接受建议） 2. 消息头错误，如MessageClass或MessageType不符合定义，SenderId不存在等 3. 消息格式不正确，即XML Schema验证未通过 4. 消息格式正确但内容错误，包括数量不符，内容重复等 5. 消息重复 6. 消息正常接收，但不接受建议（仅用于Advice Response） 7. 消息版本错误 |
| ClearTargetDate | String | YYYYMMDD 联网中心（发行方）确认归属的清算目标日期  接收方业务处理失败添全0  对于消息重复应答仍需回填归属清算目标日期 |

## 记帐处理消息结构

### 应用范围

用于发行方日间校验处理异地原始交易后，将校验处理后的扣款结果（包括可疑帐）发送或转发给通行宝中心系统。

记帐消息是由发行方根据联网中心传来的通行宝中心系统原始交易包经过记帐处理后的结果。对每一个收费方原始交易包，发行方均返回且仅返回一个记帐处理结果。

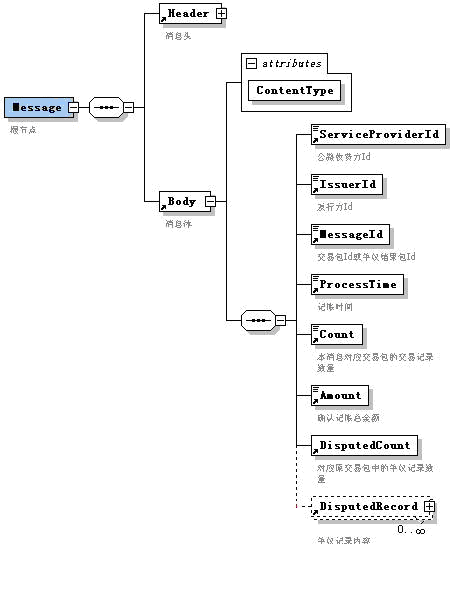
联网中心接到记帐处理后要转发到相应通行宝中心系统。记帐数据的发送方向是：联网中心→通行宝中心系统。

### 请求包格式

#### 消息头

|  |  |  |
| --- | --- | --- |
| 名称 | 数据类型 | 取值或说明 |
| Version | Hex(8) | 本次版本号统一使用0x00010000，表示版本1.0 |
| MessageClass | Int | 5，Notification |
| MessageType | Int | 5，Reconciliation Totals |
| SenderId | Hex(16位，不足补零) | 0000000000000020: 高速公路联网中心  00000000000000FE;普通公路经营性中心  00000000000000FF; 普通公路中心  00000000000000FD; 通行宝 |
| ReceiverId | Hex(16位，不足补零) | 0000000000000020: 高速公路联网中心  00000000000000FE;普通公路经营性中心  00000000000000FF; 普通公路中心  00000000000000FD; 通行宝 |
| MessageId | Long | 消息序号，从1开始，逐1递增 ，8字节 |

#### 消息内容



记帐消息的ContentType始终为1，Body各个子节点说明如下：

| 名称 | 数据类型 | 取值或说明 |
| --- | --- | --- |
| ServiceProviderId | Hex(16位，不足补零) | 通行宝中心系统Id |
| IssuerId | Hex(16位，不足补零) | 发行方Id，说明记帐消息是哪一个发行方产生的 |
| MessageId | Long | 取值为原始交易包消息体中MessageId的值 |
| ProcessTime | DateTime |  |
| Count | Int | 本消息对应的原始交易包中交易记录的数量 |
| Amount | Decimal | 确认记帐总金额 |
| DisputedCount | Int | 本消息包含的争议交易数量 （可疑帐笔数） |
| DisputedRecord |  |  |

记帐处理结果仅返回有争议（可疑）的交易记录明细。未包含在争议交易记录明细中的交易，均默认为发行方已确认可以付款。

<?xml version="1.0" encoding="UTF-8"?>

<Message>

<Header>

<Version>00020000</Version>

<MessageClass>5</MessageClass>

<MessageType>5</MessageType>

<SenderId>0000000000000020</SenderId>

<ReceiverId>00000000000000FD</ReceiverId>

<MessageId>114484</MessageId>

</Header>

<Body ContentType="1">

<ServiceProviderId>00000000000000FD</ServiceProviderId>

<IssuerId>0000000000000020</IssuerId>

<MessageId>6520</MessageId>

<ProcessTime>2018-07-03T09:57:50</ProcessTime>

<Count>2</Count>

<Amount>21.00</Amount>

<DisputedCount>1</DisputedCount>

<DisputedRecord>

<TransId>2</TransId>

<Result>3</Result>

</DisputedRecord>

</Body>

</Message>

<?xml version="1.0" encoding="UTF-8"?>

<Message>

<Header>

<Version>00020000</Version>

<MessageClass>5</MessageClass>

<MessageType>5</MessageType>

<SenderId>0000000000000020</SenderId>

<ReceiverId>00000000000000FD</ReceiverId>

<MessageId>114803</MessageId>

</Header>

<Body ContentType="1">

<ServiceProviderId>00000000000000FD</ServiceProviderId>

<IssuerId>0000000000000020</IssuerId>

<MessageId>6806</MessageId>

<ProcessTime>2018-07-05T13:31:41</ProcessTime>

<Count>5</Count>

<Amount>18.00</Amount>

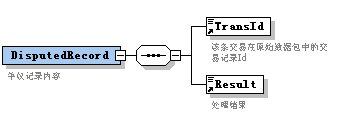
<DisputedCount>0</DisputedCount>

</Body>

</Message>

海玲它保存了重复了

争议记录的格式如下：



|  |  |  |
| --- | --- | --- |
| 名称 | 数据类型 | 取值或说明 |
| TransId | Int | 表示该条交易在原始数据包中的交易记录Id  是由通行宝中心系统产生的该包内顺序Id，从1开始递增。在通行宝中心系统、联网中心的交易通讯过程中均采用此Id表示包内唯一的交易记录。通过MessageId与TransId，可以在系统中唯一确定一条交易。 |
| Result | Short | 处理结果，详细解释见下表 |

记帐处理结果Result定义：

| 取值 | 说明 |
| --- | --- |
| 1 | 验证未通过（如：TAC错误） |
| 2 | 重复的交易信息 |
| 3 | 用户状态变化 |
| 4 | 无效交易类型 |
| 5 | 逾期超过设定值 |
| 6 | 交易数据域错 |
| 7 | 超过最大交易限额 |
| 8 | 卡号不存在 |
| 9 | 卡状态不匹配 |
| 10 | 卡超过有效期 |
| 11 | 不允许的交易 |
| 12 | 卡片CSN不匹配 |
| 13 | 测试交易 |
| 14 | 卡帐不符（仅用于储值卡） |
| 15 | 无效卡类型 |
| 16 | 车道对时错误 |
| 17 | OBU号不存在 |
| 100 | 其它 |
|  |  |

### 应答包格式

#### 应答消息内容



#### 消息头

|  |  |  |
| --- | --- | --- |
| 名称 | 数据类型 | 取值或说明 |
| Version | Hex(8) | 本次版本号统一使用0x00010000，表示版本1.0 |
| MessageClass | Int | 6 |
| MessageType | Int | 5 |
| SenderId | Hex(16位，不足补零) | 当前参与方Id  0000000000000020: 高速公路联网中心  00000000000000FE;普通公路经营性中心  00000000000000FF; 普通公路中心  00000000000000FD; 通行宝 |
| ReceiverId | Hex(16位，不足补零) | 0000000000000020: 高速公路联网中心  00000000000000FE;普通公路经营性中心  00000000000000FF; 普通公路中心  00000000000000FD; 通行宝 |
| MessageId | Long | 消息序号，从1开始，逐1递增 ，8字节 |

Body的ContentType属性是可选的，在消息头MessageClass和MessageType的基础上进一步指出响应的是哪一类消息，与所回应的消息的ContentType保持一致。Body各个子节点说明如下：

| 名称 | 数据类型 | 取值或说明 |
| --- | --- | --- |
| MessageId | Long | 确认的消息Id |
| ProcessTime | DateTime | 处理时间 |
| Result | Short | 执行结果：   1. 消息已正常接收（用于Advice Response时含已接受建议） 2. 消息头错误，如MessageClass或MessageType不符合定义，SenderId不存在等 3. 消息格式不正确，即XML Schema验证未通过 4. 消息格式正确但内容错误，包括数量不符，内容重复等 5. 消息重复 6. 消息正常接收，但不接受建议（仅用于Advice Response） 7. 消息版本错误 |

## 争议交易处理（可疑帐调整数据）

### 应用范围

每日日终处理后联网中心将上一清算日进行可疑调整确认的争议结果处理数据发送给通行宝中心系统。

争议交易一般出现在以下几种情况：数据重复；未通过认证；发行方由于用户状态等原因拒付；交易额超出最大限额等。

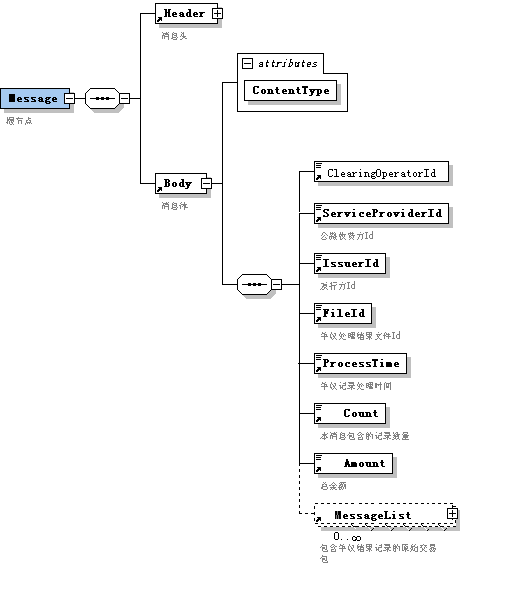
争议交易处理结果的发送方向是：联网中心→通行宝中心系统。

### 请求包格式

#### 消息头

|  |  |  |
| --- | --- | --- |
| 名称 | 数据类型 | 取值或说明 |
| Version | Hex(8) | 本次版本号统一使用0x00010000，表示版本1.0 |
| MessageClass | Int | 5，Notification |
| MessageType | Int | 7，Transaction |
| SenderId | Hex(16位，不足补零) | 0000000000000020: 高速公路联网中心  00000000000000FE;普通公路经营性中心  00000000000000FF; 普通公路中心  00000000000000FD; 通行宝 |
| ReceiverId | Hex(16位，不足补零) | 0000000000000020: 高速公路联网中心  00000000000000FE;普通公路经营性中心  00000000000000FF; 普通公路中心  00000000000000FD; 通行宝 |
| MessageId | Long | 消息序号，从1开始，逐1递增 ，8字节 |

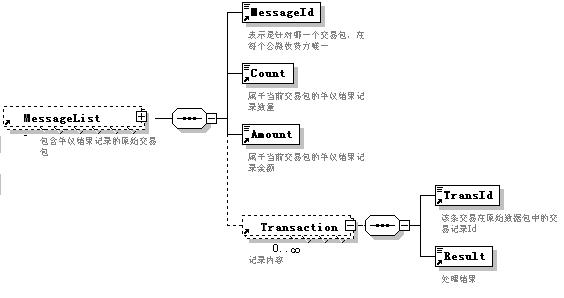
#### 消息内容



争议记录与原始交易记录的MessageClass和MessageType相同。争议记录Body的ContentType始终为2，与交易记录的1相区别。Body各个子节点说明如下：

| 名称 | 数据类型 | 取值或说明 |
| --- | --- | --- |
| ClearingOperatorId | Hex(16位，不足补零) | 清分方ID，联网中心ID |
| ServiceProviderId | Hex(16位，不足补零) | 联网中心ID |
| IssuerId | Hex(16位，不足补零) | 发行方Id，表示争议交易属于哪一个发行方 |
| FileId | Int | 争议结果文件Id ，使用此栏位确认争议归属清算日期  （现使用OBU中心清算日ClearTargetDate作为唯一标识,YYYYMMDD）  如果争议结果文件每天需要交互多次，建议FileID编码规则改为前8位为争议处理结果归属清算日yyyymmdd,加1位序号 |
| ProcessTime | DateTime | 争议处理时间 |
| Count | Int | 本消息包含的记录数量，包括经讨论确认付款和记录和坏帐  记录数量 |
| Amount | Decimal十进制浮点数 | 确认需要记帐的总金额 |

争议交易记录List的格式如下：



|  |  |  |
| --- | --- | --- |
| 名称 | 数据类型 | 取值或说明 |
| MessageList |  | 表示其包含的争议结果记录来自于收费方某个原始交易包 |
| MessageId | Long | 通行宝中心系统原始交易包ID |
| Count | Int | 属于当前交易包的争议结果记录数量 |
| Amount | Decimal | 属于当前交易包的争议结果记录金额 |
| Transaction |  | 争议处理结果记录 |
| TransId | Int | 表示该条交易在原始数据包中的交易记录Id |
| Result | Short | 为0表示正常支付；为1表示此交易作坏账处理。 |

对争议交易的处理结果是或者全额付款，或者按坏帐处理不付款，不会发生只支付一部分的情况，所以此处不再说明应支付的金额。

<?xml version="1.0" encoding="UTF-8"?>

<Message>

<Header>

<Version>00010000</Version>

<MessageClass>5</MessageClass>

<MessageType>7</MessageType>

<SenderId>0000000000000020</SenderId>

<ReceiverId>00000000000000FD</ReceiverId>

<MessageId>114464</MessageId>

</Header>

<Body ContentType="2">

<ClearingOperatorId>0000000000000020</ClearingOperatorId>

<ServiceProviderId>0000000000000020</ServiceProviderId>

<IssuerId>00000000000000FD</IssuerId>

<FileId>20180702</FileId>

<ProcessTime>2018-07-03T09:06:04</ProcessTime>

<Count>2</Count>

<MessageList>

<MessageId>6410</MessageId>

<Count>1</Count>

<Amount>3.00</Amount>

<Transaction>

<TransId>1</TransId>

<Result>1</Result>

</Transaction>

</MessageList>

<MessageList>

<MessageId>6462</MessageId>

<Count>1</Count>

<Amount>2.00</Amount>

<Transaction>

<TransId>1</TransId>

<Result>1</Result>

</Transaction>

</MessageList>

</Body>

</Message>

### 应答包格式

#### 应答消息内容



#### 消息头

|  |  |  |
| --- | --- | --- |
| 名称 | 数据类型 | 取值或说明 |
| Version | Hex(8) | 本次版本号统一使用0x00010000，表示版本1.0 |
| MessageClass | Int | 6，Notification Response |
| MessageType | Int | 7，Transaction |
| SenderId | Hex(16位，不足补零) | 0000000000000020: 高速公路联网中心  00000000000000FE;普通公路经营性中心  00000000000000FF; 普通公路中心  00000000000000FD; 通行宝 |
| ReceiverId | Hex(16位，不足补零) | 0000000000000020: 高速公路联网中心  00000000000000FE;普通公路经营性中心  00000000000000FF; 普通公路中心  00000000000000FD; 通行宝 |
| MessageId | Long | 消息序号，从1开始，逐1递增 ，8字节 |

Body的ContentType始终为2。Body各个子节点说明如下：

| 名称 | 数据类型 | 取值或说明 |
| --- | --- | --- |
| MessageId | Long | 确认的消息Id |
| ProcessTime | DateTime | 处理时间 |
| Result | Short | 执行结果：   1. 消息已正常接收（用于Advice Response时含已接受建议） 2. 消息头错误，如MessageClass或MessageType不符合定义，SenderId不存在等 3. 消息格式不正确，即XML Schema验证未通过 4. 消息格式正确但内容错误，包括数量不符，内容重复等 5. 消息重复 6. 消息正常接收，但不接受建议（仅用于Advice Response） 7. 消息版本错误 |

### 处理流程



争议交易由联网中心、通行宝中心系统方协商后得出处理结果，由联网中心负责产生争议交易处理结果数据包并发送给通行宝中心系统。对于外省的争议，由联网中心负责与外省份协调后，将争议处理结果转发给通行宝中心系统。

各方产生的争议交易限定在T+2内解决，确认是否付款还是作为坏帐处理，并由联网中心通过争议结果消息发送给通行宝中心系统。

每次处理可以包含该组通行宝中心系统与联网中心之间多个交易信息包中的交易。一个原始交易包中被发行方认定有争议的交易可以分多次处理，被包含在多个争议处理结果中。已被处理的争议交易不得被再次处理。

每次处理，产生一个处理结果文件，以FileId作为唯一标识。FileId编码规则：暂时使用联网中心清算日期ClearTargetDate ,通过FileId可直接确认本次争议结果归属的清算日期便于清分统计对帐。

由联网中心提交争议结果前，联网中心按争议结果生成一份正式的文件，打印后由各方签字保存，作为争议处理结果的凭证。每份凭证均有唯一的文件号FileId。该凭证是执行争议结果的依据。

通过协商后由联网中心产生的争议处理结果为最终结果，各方均需按该结果执行相应操作。

争议交易消息中的记录不包含交易细节，仅提供交易记录号。通行宝中心系统应使用交易记录号TransId获取交易细节以进一步处理。

通行宝中心系统接收到争议处理结果后应按其内容将对应交易记录设置为确认应收/付或坏帐。

## 清分统计消息结构

### 应用范围

联网中心按清算日统计已经发行方确认可以付款的交易，产生清分统计信息，发送给通行宝中心系统核对。

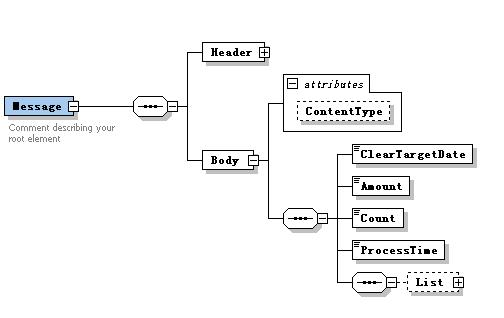
交易数据的发送方向是：联网中心→通行宝中心系统。

### 请求包格式

#### 消息头

|  |  |  |
| --- | --- | --- |
| 名称 | 数据类型 | 取值或说明 |
| Version | Hex(8) | 本次版本号统一使用0x00010000，表示版本1.0 |
| MessageClass | Int | 5，Notification |
| MessageType | Int | 5，Reconciliation Totals |
| SenderId | Hex(16位，不足补零) | 0000000000000020: 高速公路联网中心  00000000000000FE;普通公路经营性中心  00000000000000FF; 普通公路中心  00000000000000FD; 通行宝 |
| ReceiverId | Hex(16位，不足补零) | 0000000000000020: 高速公路联网中心  00000000000000FE;普通公路经营性中心  00000000000000FF; 普通公路中心  00000000000000FD; 通行宝 |
| MessageId | Long | 消息序号，从1开始，逐1递增 ，8字节 |

#### 消息内容

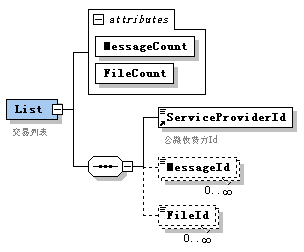


清分消息的MessageClass及MessageType与记帐消息相同。清分消息的ContentType始终为2，与记帐消息相区别。Body各个子节点说明如下：

| 名称 | 数据类型 | 取值或说明 |
| --- | --- | --- |
| ClearTargetDate | Date | 清分目标日 |
| Amount | Decimal | 清分总金额(确认付款金额) |
| Count | Int | 对应清分总金额的的交易记录数量，包含原始交易包中由发行方确认应付的交易数量和争议处理结果中确认应付的交易数量之和，不包含争议处理结果中为坏帐的记录数量。 |
| ProcessTime | DateTime | 清分统计处理时间 |
| IssuerId | Hex(16位，不足补零) | 发行方Id，向通行宝中心系统发送的清分信息中不使用 |

ProcessTime之后的内容，或者包含IssuerId及0或多个List，或者仅包含一个List。前者是联网中心为发行方产生的清分统计结果，后者是为通行宝中心系统产生的清分统计结果。

清分细节List格式如下：



|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据类型 | 取值或说明 | |
| MessageCount | Int | 本次清分包含的所有原始交易包数量 | |
| FileCount | Int | 本次清分包含的所有争议处理结果包数量 | |
| ServiceProviderId | Hex(16位，不足补零) | 通行宝中心系统Id | 在系统内唯一确定原始交易消息包ID |
| MessageId | Long | 通行宝中心系统发的原始交易包Id  由于此处为messageid的明细，所以建议收费方上传打包时尽量在一个messageid中多打包原始记录（小于10000条），减少此处的数据量，因为该包格式为单包 |
| FileId | Int | 争议处理结果文件Id (可疑调整数据) | |

List说明本次清分所包含的交易记录范围。该范围可以通过交易记录消息包MessageId和争议处理结果FileId确定。

<?xml version="1.0" encoding="UTF-8"?>

<Message>

<Header>

<Version>00010000</Version>

<MessageClass>5</MessageClass>

<MessageType>5</MessageType>

<SenderId>0000000000000020</SenderId>

<ReceiverId>00000000000000FD</ReceiverId>

<MessageId>114462</MessageId>

</Header>

<Body ContentType="2">

<ClearTargetDate>2018-06-30</ClearTargetDate>

<Amount>0.00</Amount>

<Count>0</Count>

<ProcessTime>2018-07-02T09:18:05</ProcessTime>

<List MessageCount="0" FileCount="0">

<ServiceProviderId>00000000000000FD</ServiceProviderId>

<MessageId/>

</List>

</Body>

</Message>

### 应答包格式

#### 应答消息内容



#### 消息头

|  |  |  |
| --- | --- | --- |
| 名称 | 数据类型 | 取值或说明 |
| Version | Hex(8) | 本次版本号统一使用0x00010000，表示版本1.0 |
| MessageClass | Int | 6，Notification Response |
| MessageType | Int | 5，Reconciliation Totals |
| SenderId | Hex(16位，不足补零) | 当前参与方Id  0000000000000020: 高速公路联网中心  00000000000000FE;普通公路经营性中心  00000000000000FF; 普通公路中心  00000000000000FD; 通行宝 |
| ReceiverId | Hex(16位，不足补零) | 准备接收确认消息的参与方Id  编码同上 |
| MessageId | Long | 消息序号，从1开始，逐1递增 ，8字节 |

Body的ContentType属性是可选的，在消息头MessageClass和MessageType的基础上进一步指出响应的是哪一类消息，与所回应的消息的ContentType保持一致。Body各个子节点说明如下：

| 名称 | 数据类型 | 取值或说明 |
| --- | --- | --- |
| MessageId | Long | 确认的消息Id |
| ProcessTime | DateTime | 处理时间 |
| Result | Short | 执行结果：   1. 消息已正常接收（用于Advice Response时含已接受建议） 2. 消息头错误，如MessageClass或MessageType不符合定义，SenderId不存在等 3. 消息格式不正确，即XML Schema验证未通过 4. 消息格式正确但内容错误，包括数量不符，内容重复等 5. 消息重复 6. 消息正常接收，但不接受建议（仅用于Advice Response） 7. 消息版本错误 8. 清分统计对帐失败 |

### 处理流程

联网中心每日定时以前一日为清分目标日进行清分统计，产生的结果发送给通行宝中心系统核对。清分统计结果是最终结算划帐的依据。

清分统计生成后不能更改。

# （规范性附录）消息总结

前面定义的所有消息，均只定义了基本的数据结构，并未使用XML的内在功能全面定义数据约束。这些数据约束应由应用程序自行处理。

很多确认消息的结果现阶段是相同的，但为保证后续可能的变动不影响开发，每种确认消息均使用独立的Schema。

## 消息列表

下表为各个消息使用MessageClass、MessageType、ContentType和Schema文件总结。

| 消息应用 | MessageClass | MessageType | Content- Type | 备注 |
| --- | --- | --- | --- | --- |
| 原始交易消息 | 5, Notification | 7, Transaction | 1 | 已定义 |
| 争议消息记录 | 5, Notification | 7, Transaction | 2 | 未定义 |
| 退费消息 | 5, Notification | 7, Transaction | 3 | 已定义 |
| 调账交易消息 | 5, Notification | 7, Transaction | 9 | 已定义 |
| 记帐消息 | 5, Notification | 5，Reconciliation Totals | 1 | 未定义 |
| 清分消息 | 5, Notification | 5，Reconciliation Totals | 2 | 未定义 |
| 结算消息 | 5, Notification | 5，Reconciliation Totals | 3 | 未定义 |
| 交易通讯确认 | 6,Notification Response | 5，Reconciliation Totals  7, Transaction  17 TagAbuse  19,HistoryTransaction |  |  |
| 重复收费交易明细请求 | 5, Notification | 20, |  |  |
| 重复收费交易明细确认 | 6,Notification | 20 |  |  |
| 重复收费交易对帐请求 | 5, Notification | 21 |  |  |
| 重复收费交易对帐确认 | 6,Notification | 21 |  |  |

## 消息确认对应关系

| 发送的消息 | 确认消息 |
| --- | --- |
| 记帐消息 | 交易通讯确认 |
| 调帐消息 | 交易通讯确认 |

# 六、测试用例