

高速公路区域联网不停车收费示范工程暂行技术要求 第 20 部分

## 跨省（区、市）数据接口规范

2008 年 8 月



目 次

1 范围..... 2

2 规范性引用文件..... 2

3 术语和定义、符号、缩略语..... 2

4 传输规则..... 2

5 交易处理..... 13

6 用户状态处理..... 31

7 基础信息维护..... 36

附 录 A （规范性附录） 消息总结 ..... 44

A.1 消息列表..... 44

A.2 消息确认对应关系..... 45

参考文献..... 46

1 范围

本标准规定了高速公路联网电子收费系统跨省（区、市）电子收费系统中清分方之间的数据传输接口及处理流程。

本标准适用于公路联网电子收费系统的设计、开发与应用，城市收费道路也可参考执行。

2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本标准，然而，鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

GB/T 20610—2006/ISO/TS 14904 ： 2002 《道路运输与交通信息技术电子收费（EFC）参与方之间信息交互接口的规范》

3 术语和定义、符号、缩略语

3.1 术语和定义

《省（区、间）数据接口规范》中确立的术语和定义适用于本规范。

3.2 缩略语

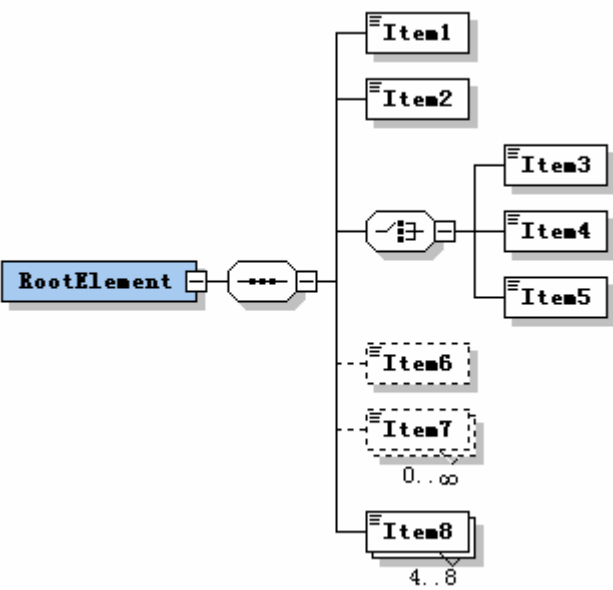
本标准所用缩略语如下表。

缩略语

缩略语	英文全称	含义
XML	eXtensible Markup Language	一种简单的数据存储语言，使用一系列简单的标记描述数据。
ID	Identity	身份标识号码，也叫帐号，是一个编码，具有唯一性。

3.3 XML 符号定义

本文中定义XML结构的Schema通过如下图形表示：




所有XML节点定义均以方框套节点名称定义，如上图中的RootElement及Item1到Item8。根据连接线可知各个节点的关系：Item1到Item8均为RootElement的子节点。

如果一个节点必须出现且仅能出现一次，则其方框为实线，没有任何下标，如Item1到Item5。

如果一个节点可以被省略，即其出现次数可以为0，则其方框为虚线，如Item6和Item7。Item6的虚框下无下标，说明Item6最多可以出现一次；Item7的虚框下有下标，指明其出现次数的上限（上图中定义为无穷大）。

Item8的下标说明其出现次数必须在4次到8次之间，否则不能通过XML合法性验证。

两个图形说明子节点的出现规则。前者表示子节点按结构图从上到下的顺序出现。例如，RootElement的子节点必须按Item1、Item2、Item3...的顺序出现，否则无法通过合法性验证。后者表示子节点的出现是选择关系。例如，Item3、Item4、Item5均为RootElement的子节点，但在任意一个XML文件中，只能出现这三者之一，不能同时出现。

4 传输规则

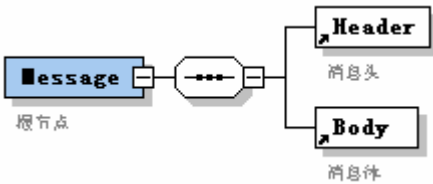
4.1 传输方式

所有数据均通过基于TCP/IP的通讯协议或中间件传送。

4.2 基本结构

4.2.1 数据存储形式

所有传输的数据及回应消息均采用XML，编码为UTF-8，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，所有数值类型的值均采用十进制表示，

所有表示金额的节点均采用十进制并精确到元，例如：123.45表示一百二十三元四角五分。

所有数据结构均以Schema形式定义，所有XML数据必须能够通过对应Schema的合法性验证。

4.2.2 数据结构定义

所有传输中的消息，均通过Schema定义文件结构。所有根据Schema生成的XML文件，必需是合法的。

Schema文件仅定义文件结构，不负责数据的逻辑合法性进行验证。

Schema定义中使用的标签名称（tag）与数据库定义使用的字段名没有必然关系。数据库定义时可以采用不同的名称表示Schema定义的内容。

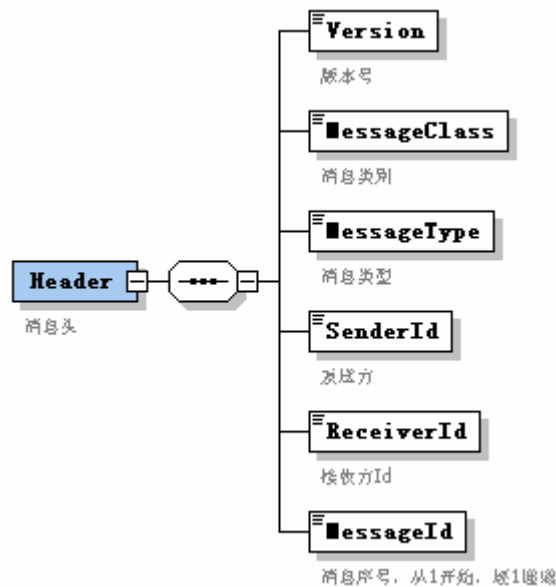
4.2.3 数据类型

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	

数据类型	说明	示例
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

#### 4.3 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，第三个字节表示次版本号，最后一个字节表示修改编号。如 00010102（16 进制）

名称	数据类型	取值及说明
		表示版本 1.1.2。
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)	发送方 Id，在整个系统中唯一
ReceiverId	Hex(16)	接收方 Id，在整个系统中唯一
MessageId	Long	消息序号，8 字节整数，从 1 开始，逐 1 递增

通过SenderId、ReceiverId及MessageId组合，可以在整个系统中唯一确定一条消息。  
MessageId由发送方产生。

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	

以C#定义为：

```
public enum MessageClass
{
    Request = 1,
    RequestResponse,
    Advice,
    AdviceResponse,
    Notification,
    NotificationResponse
}
```

MessageType以4字节整型表示。

名称	值
服务列表 Service List	1
价目表 Fare Products List	2
用户信息 Customer Details	3

名称	值
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
保留	16
标签拆卸	17
未定义的消息类型 Undefined Message Type	18 至 65535
个性化消息类型	大于 65535 的整数

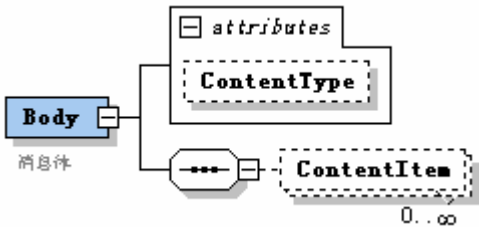
以C#定义为：

```
public enum MessageClass
{
    ServiceList = 1,
    FareProductsList,
    CustomerDetails,
    ApportionmentRules,
    ReconciliationTotals,
    Authorization,
    Transaction,
    ReportSent,
    KeyManagement,
    StatusList,
    EquipmentStatus,
    EventException,
    PaymentMethodAcceptance,
    OperatorList,
```



```
PrivilegeList,  
Reserved,  
TagAbuse,  
Undefined  
}
```

4.4 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。

4.5 消息文件的命名规则

接收方为校验文件在传输过程中的完整性，约定收发双方以标准MD5算法（RFC 1321 The MD5 Message-Digest Algorithm, MD5信息分类算法）对文件进行校验。发送方生成文件后，将文件转换成2进制流用于MD5计算。计算所得结果为16字节2进制数据（128位），并用MD5结果对文件命名。

命名规则为：  
MessageId（消息包ID，10进制）+ ‘\_’ + MD5验证结果（16进制，不足左补0）+ 文件扩展名。

未压缩的原始消息文件，文件扩展名为 ‘.XML’ ,压缩后的扩展名为 ‘.ZIP’ 。  
每一个压缩文件仅包含一个原始数据文件。压缩文件与原始数据文件除扩展名不同外，文件名部分完全相同。

4.6 传输控制

发送方与接收方的数据传输采用一问一答方式。发送方在规定时间内未接收到接收方的应答需通过自动重发、手动重发及文件导入/导出功能将数据传送到接收方。重发消息、导出消息的MessageId保持不变。

发送方发送的两条消息之间不存在必然的逻辑关系。

4.6.1 通用确认消息结构

4.6.1.1 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。

接收方对于消息结构不正确（例如MessageClass值未定义）的消息，使用通用确认消息结构通知发送方消息异常。

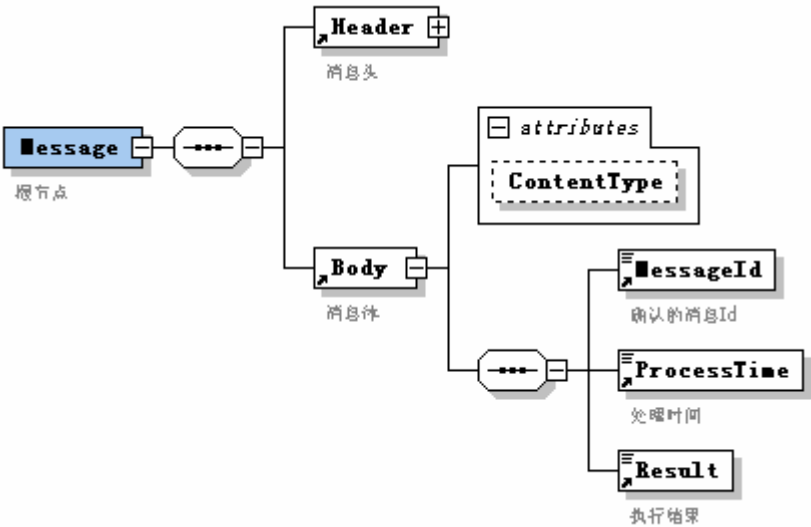
各消息的详细回应说明请参与相关章节。

4.6.1.2 消息头

名称	数据类型	取值或说明
MessageClass	Int	使用所接收消息的 MessageClass

名称	数据类型	取值或说明
MessageType	Int	若所接收消息的 MessageType 有效，使用与其对应的 Response 值；否则使用所接收消息的 MessageType 值加 1
SenderId	Hex(16)	当前参与方 Id
ReceiverId	Hex(16)	准备接收确认消息的参与方 Id

#### 4.6.1.3 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
ProcessTime	DateTime	处理时间
Result	Short	执行结果： <ol style="list-style-type: none"> <li>1. 消息已正常接收（用于 Advice Response 时含已接受建议）</li> <li>2. 消息头错误，如 MessageClass 或 MessageType 不符合定义，SenderId 不存在等</li> <li>3. 消息格式不正确，即 XML Schema 验证未通过</li> <li>4. 消息格式正确但内容错误，包括数量不符，内容重复等</li> <li>5. 消息重复</li> <li>6. 消息正常接收，但不接受建议（仅用于 Advice Response）</li> <li>7. 消息版本错误</li> <li>8. 参与方 ID 未定义：消息体内部包含的有关参与方信息的 ID 在系统中未定义</li> </ol>

名称	数据类型	取值或说明
		9. 不支持的业务：消息格式等均正确，但发送方不应将该消息发送给接收方，例如，服务类型只能由清分方产生并发送给其他参与方，发行方和收费服务方不能产生并发送该类型的消息

4.6.2 通用重发请求消息结构

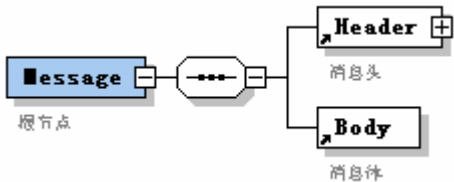
4.6.2.1 应用范围

应用于数据接收方向数据发送方请求重发某些数据，如基础信息。

4.6.2.2 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)	当前参与方 Id
ReceiverId	Hex(16)	准备接收确认消息的参与方 Id

4.6.2.3 消息内容



通用重发请求消息中没有更多的数据，其Body为空。

4.6.3 名单数据的形式

名单数据主要应用于用户状态名单、基础信息等所有经常变动的数据。

名单数据会随着系统运行不断更新。所有名单类数据的更新方式分为整体更新和增量更新两类。整体更新是数据包包含所有名单记录，接收方通过删除原有名单，直接使用接收到的新名单即可达到名单同步的目的。增量更新是发送方只告知接收方发生数据内容改变的记录，接收方根据增量内容修改其现有名单从而达到数据同步。

4.6.4 名单数据的版本控制

以下处理规则用于确保名单数据在通讯系统不保证数据的接收与发送顺序一致的情况下正确更新。如果通讯的双方能够保证发送顺序与接收顺序相同，则不必使用下文有关版本控制的处理逻辑。

4.6.4.1 名单顺序

整体下发可以保证发送方与接收方名单数据的同步，但每当名单发生变化时都使用整体形式下发会降低系统效率，因为大部分名单数据在两次下发之间是没有变化的。整体下发是静态的。

增量下发是动态的，相对整体下发数据量少，适合及时通知接收方名单的改变。

通过以上两种方式可以有效地同步发送方与接收方的名单数据，但这种方式对发送顺序与接收顺序要求十分严格。如果接收顺序与发送顺序不同，会使数据更新异常。大多数中间件均不能保证消息的发送顺序与接收顺序相同，所以在名单数据中，以版本号表示发送的先后顺序。

#### 4.6.4.2 主动发送的版本处理

##### 处理规则

发送方保证版本号逐一递增。接收方校验版本号，并根据版本号及名单形式执行相应处理。

设接收方已处理的版本号为OldVer，刚刚接收的名单版本号为NewVer，处理规则如下：

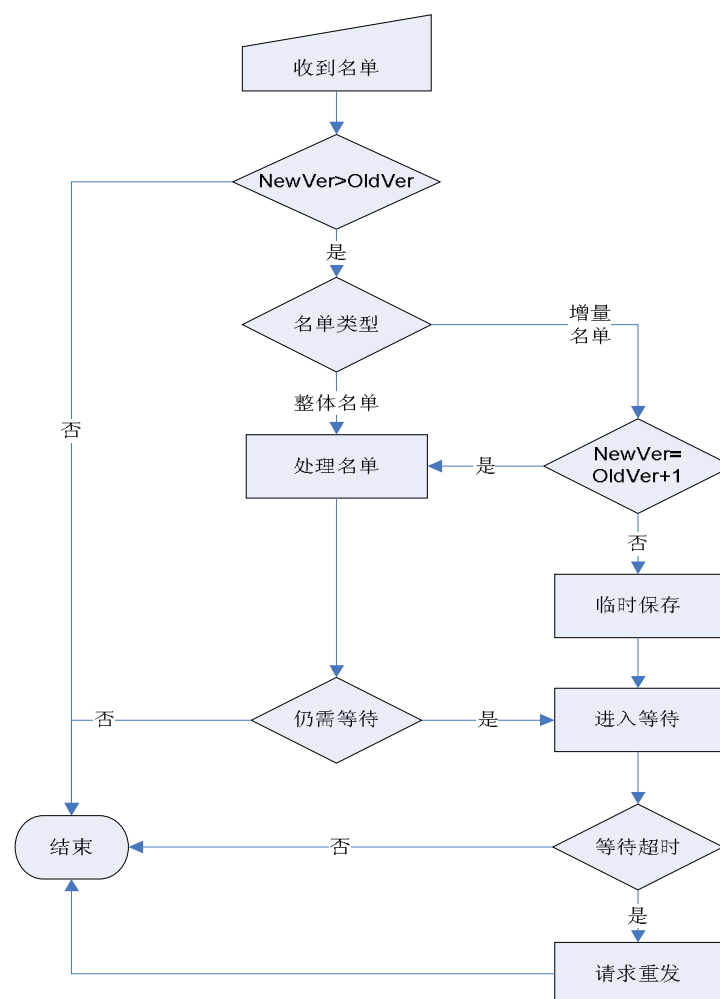
1) 若 $NewVer \leq OldVer$ ，说明当前使用的名单比接收到的名单版本更新，所以直接忽略接收到的名单。否则，转至下一步。

2) 如果新接收的名单是整体名单，则只要 $NewVer > OldVer$ 则可直接处理接收到的名单，清除在第3步中临时保存的版本 $\leq NewVer$ 名单，完成后更新OldVer的值，即设置 $OldVer = NewVer$ 。

3) 如果新接收的名单是增量名单，则只有 $NewVer = OldVer + 1$ 时方可立即处理，并更新OldVer的值后转到第4步。否则临时保存该名单直到合适的名单（ $NewVer = OldVer + 1$ 的增量名单或 $NewVer > OldVer$ 的整体名单）到达。等待时间可设定。若等待一段时间后仍没有合适的名单，则向发送方请求重发名单（如果以前已经发送过整体名单请求重发消息且没有收到回复则不发送），之后收到的名单分别按第2或3步处理。

##### 示例

以下是名单处理示例流程图：



上图中未包含退出等待状态，说明见下文示例。

“处理名单”包括的操作有：

- l 根据名单更新本地数据库；
- l 删除临时保存的版本号小于 NewVer 的名单；
- l 如果仍有临时保存的名单中存在，版本连续且与 NewVer 相临，则循环处理这些名单；
- l 更新 OldVer 值为最大已处理名单的版本号。

处理完成后临时保存的只有版本号大于 OldVer + 1 的名单。

等待状态中可以继续接收消息并处理。

举例：当前已处理的状态名单版本为3，之后收到的版本顺序为：6（增量），7（增量），4（增量），9（增量），8（整体），10（回复请求重发的整体名单），5（增量）。

处理过程为：

- l 收到版本为 6 的名单：临时保存，进入等待状态（假设等待时间结束为收到版本为 9 的名单之后）。
- l 收到版本为 7 的名单：临时保存，保持原等待状态。
- l 收到版本为 4 的名单：处理此名单，更新 OldVer 为 4，因为临时保存的名单显示仍缺版本为 5 的名单，所以不改变等待状态。等待时钟重新计时（假设等待时间结束仍为收到版本为 9 的名单之后）。
- l 收到版本为 9 的名单：临时保存，保持原等待状态。
- l 等待结束，说明通讯可能发生问题，为及时得到最新的名单，发送名单请求重发消息给发送方。
- l 收到版本为 8 的整体名单：处理此名单，更新 OldVer 为 8，删除临时保存的版本为 6 和 7 的名单，不对这两个名单进行处理。
- l 处理临时保存的版本为 9 的名单，更新 OldVer 为 9，退出等待状态。
- l 收到版本为 10 的回复名单：处理此名单，更新 OldVer 为 10。
- l 收到版本为 5 的名单：忽略。

#### 4.6.4.3 响应请求重发的版本处理

名单接收方可以向名单发送方请求发送当前完整的名单信息。名单的发送方有两类，一类是名单的产生方，即产生用户状态名单的发行方；另一类是名单的转发方，即清分方。

名单的产生方在接收到重发请求后，可以产生包含整体名单数据及新版本号的数据名单，接收方的处理与主动发送的版本处理相同。

名单的转发方自己并不产生名单，所以不能产生新的版本号，只能使用当前最新的版本号。

名单转发方的处理规则如下：

- l 名单转发方接收到重发请求后，根据本地数据产生整体名单，并使用最新的版本号作为名单的版本号。

I 设接收方已处理的版本号为 OldVer，刚刚接收的名单版本号为 NewVer，处理规则如下：

I 若  $NewVer < OldVer$ ，说明当前使用的名单比接收到的名单版本更新，所以直接忽略接收到的名单。否则，因为重发请求回应是整体名单，所以直接处理接收到的名单并更新 OldVer，删除所有临时保存的名单。

#### 4.7 名单数据的有效期

名单数据除服务类型外，通讯消息中每条记录均有生效时间。各个系统在使用名单数据时均需检查生效时间，并根据接收到的新的名单处理已有记录的失效时间。到达失效时间后，系统应自动将记录从名单中备份删除。名单数据应能保存多个版本。

用于传送名单数据的消息仅包含每条记录的生效时间，但在系统中，应保存名单记录的生效时间和失效时间。失效时间由接收系统自行维护。

以用户状态名单为例，当前系统使用的用户状态名单记录如下：（生效时间及失效时间均可精确到秒，但示例中为简单仅精确到日；卡ID也只使用发行序号）

8月1日状态名单

卡 ID	状态	生效时间	失效时间
1	透支	2007 年 8 月 1 日	无
2	挂失	2007 年 8 月 1 日	无
3	禁用	2007 年 8 月 1 日	无

因为名单消息中仅指出每条记录的生效时间，没有失效时间，所以默认为永不失效。失效时间根据新接收到的名单发生变化。

2007年8月4日接收到增量状态名单，包含卡ID为1、2、4、5的记录：

8月4日增量名单

卡 ID	状态	生效时间
1	正常	2007 年 8 月 5 日
2	禁用	2007 年 8 月 5 日
4	挂失	2007 年 8 月 5 日
5	禁用	2007 年 8 月 5 日

根据该增量名单更新后，系统中的名单记录为：

8月4日更新后的状态名单

卡 ID	状态	生效时间	失效时间
1	透支	2007 年 8 月 1 日	2007 年 8 月 5 日
2	挂失	2007 年 8 月 1 日	2007 年 8 月 5 日
2	禁用	2007 年 8 月 5 日	无
3	禁用	2007 年 8 月 1 日	无

卡 ID	状态	生效时间	失效时间
4	透支	2007 年 8 月 5 日	无
5	挂失	2007 年 8 月 5 日	无

在8月5日之前，1号卡仍然处于透支状态；到8月5日，系统应自动将其删除。同理，系统在8月5日删除2号卡状态为挂失的记录。在未收到新的名单之前，2至5号卡的状态均永久有效。

在以上结果中，同时保存了有关2号卡的两条记录。系统根据自行维护的失效时间删除符合失效时间范围的记录。

8月6日收到整体名单如下：

8月6日的整体名单

卡 ID	状态	生效时间
2	禁用	2007 年 8 月 5 日
4	透支	2007 年 8 月 5 日
6	禁用	2007 年 8 月 8 日

则系统内状态名单更新为：

8月6日更新后的状态名单

卡 ID	状态	生效时间	失效时间
2	禁用	2007 年 8 月 5 日	无
4	透支	2007 年 8 月 5 日	无
6	禁用	2007 年 8 月 8 日	无

整体名单包含全部最新的状态记录，所以直接删除系统中现有记录，并使用新数据更新即可。

4.8 参与方 ID

在消息交换中使用的发送方ID、接收方ID，以及消息中包含的收费服务方ID、发行方ID及清分方ID均可以8字节整数存储，在整个系统内唯一。

在XML中，参与方ID表现为16位长的16进制字符串，数据类型为HexBinary，不足16位左侧补0。

4.9 卡 ID 及卡类型

卡类型分为国标IC卡和车载单元两种。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16进制数，卡发行号为16位16进制数。在XML文件中，均以HexBinary表示。不足位数左侧补0。

5 交易处理

## 5.1 应用范围

交易处理是公路收费交易从收费服务方到清分方，再到发行方的整个传输、记帐、争议处理、清分统计、结算划帐等各个过程的总和。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。

## 5.2 处理规则

### 5.2.1 交易处理

参见《高速公路联网电子收费清分结算系统运行规则》。

### 5.2.2 清分

清分系统每天统计如下两组交易：

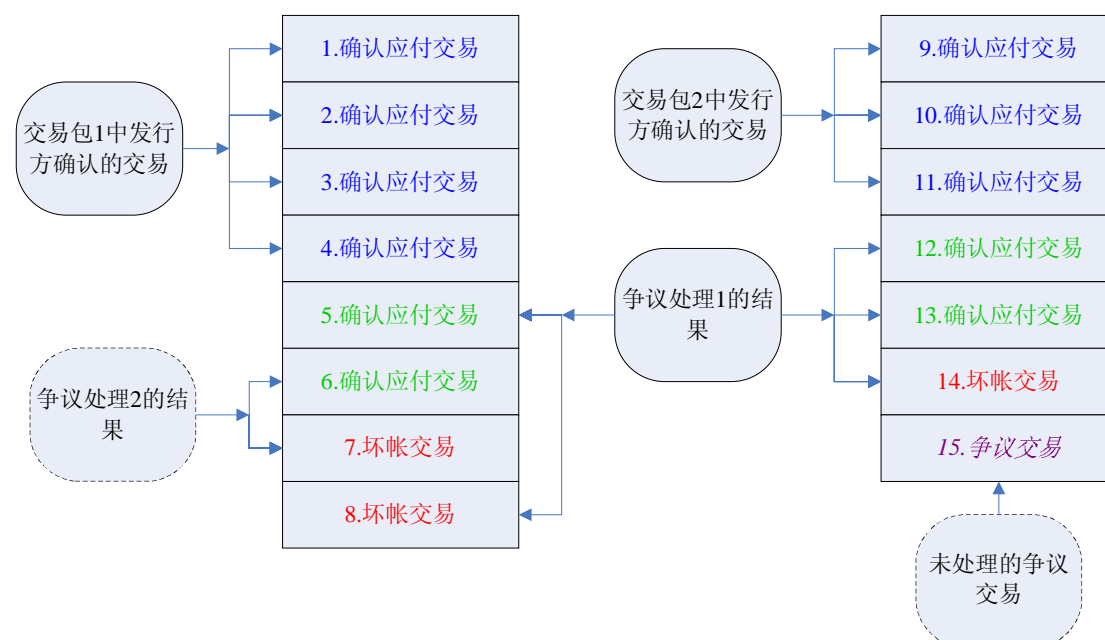
- I 所有收到的，已由发行方记帐确认的，消息包的清分目标日早于或等于需统计的清分目标日的交易包所包含的交易。统计时仍未收到的交易包，或交易包已收到但尚未得到发行方记帐确认的交易包后到下一日统计。
- I 在清分目标日内及以前产生的未清分统计过的争议交易处理结果所包含的交易。

所统计的交易必须同时满足以下所有条件：

- 1) 所统计的交易必须符合时间区段的约束，即交易包的清分目标日或争议处理日期必须早于时间区段的结束时间；
- 2) 所有已由发行方确认的交易包所包含的确认付款的交易；
- 3) 所有由争议处理确认付款的交易；
- 4) 所统计的交易尚未参与清分。

对进行过清分统计的清分目标日不再进行清分统计，其统计结果不能更改。

下图举例说明：



假设交易包1、2和争议处理结果1、2均未进行过清分统计，但争议结果2不符合清分目标日的时间范围，所以清分时仅统计交易包1、2和争议处结果1。

因此，清分结果包含交易1、2、3、4、5、8、9、10、11、12、13和14。

包含交易6、7的争议结果2会在后续的清分日统计。



交易15尚未进行过处理，所以直到争议处理后再统计。

5.3 原始交易消息结构

5.3.1 应用范围

由收费服务方将交易分组打包后发送给清分方的原始交易数据。交易数据的发送方向是：收费服务方→本地清分方，本地清分方→异地清分方，异地清分方→异地发行方，三个阶段的传输使用相同的消息结构。

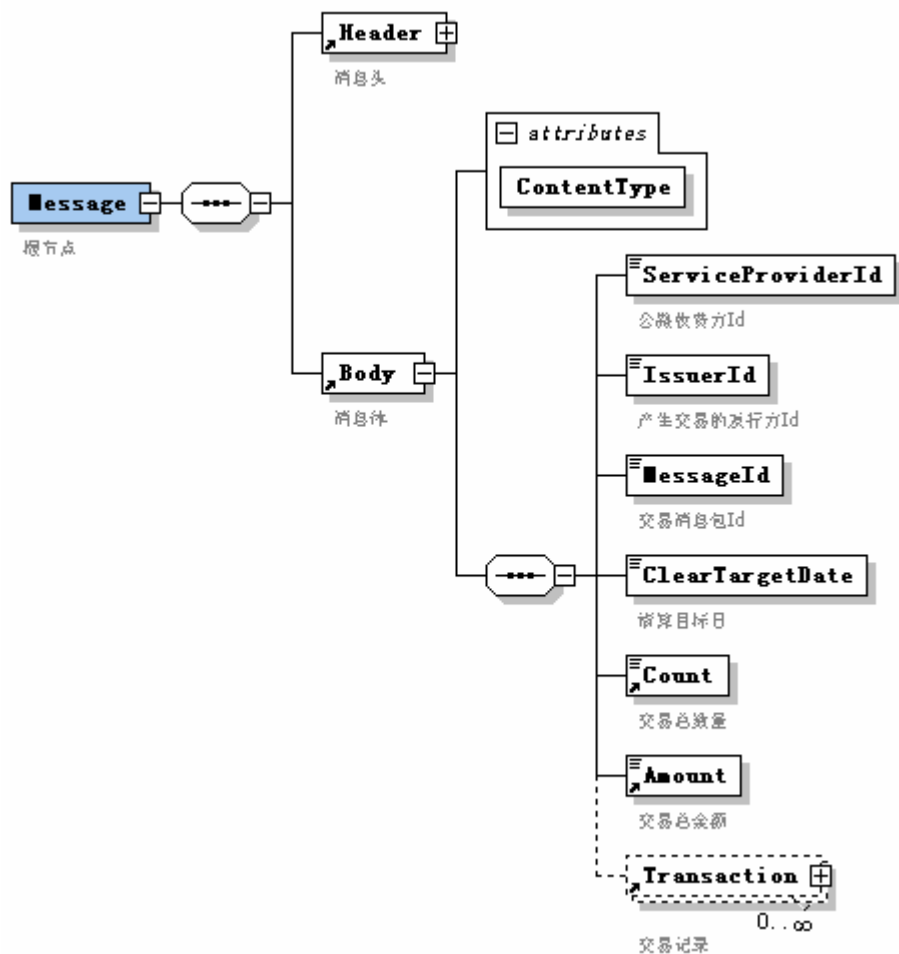
5.3.2 消息头

名称	数据类型	取值或说明
MessageClass	Int	5, Notification
MessageType	Int	7, Transaction
SenderId	Hex(16)	收费服务方系统 Id/清分方 Id
ReceiverId	Hex(16)	清分方 Id/发行方 Id

SenderId与ReceiverId的取值：

发送阶段	SenderId	ReceiverId
收费服务方→清分方	收费服务方 Id	清分方 Id
本地清分方→异地清分方	本地清分方 Id	异地清分方 Id
清分方→发行方	清分方 Id	发行方 Id

5.3.3 消息内容

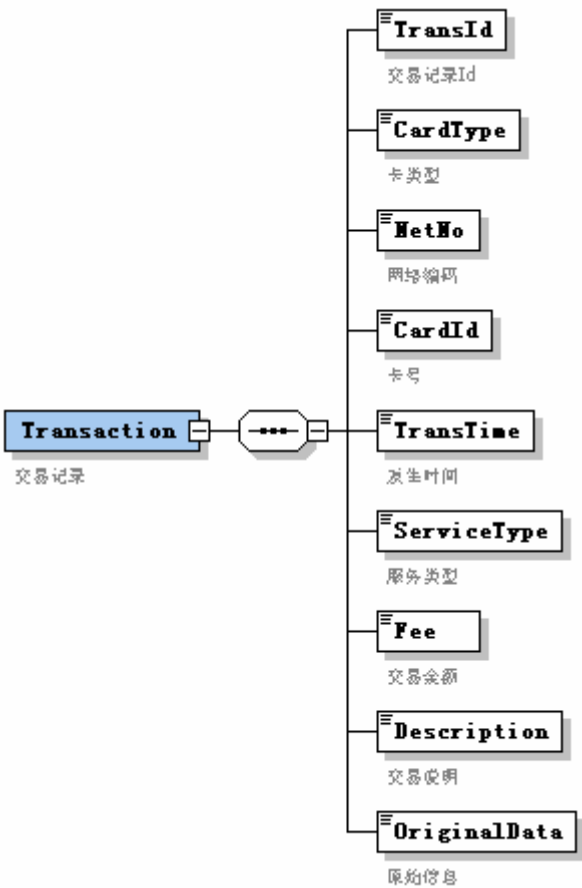


交易信息中，Body的ContentType始终为1，Body各个子节点说明如下：

名称	数据类型	取值及说明
ServiceProviderId	Hex(16)	收费服务方 Id，表示消息包中的交易是由哪个收费服务方产生的。
IssuerId	Hex(16)	发行方 Id，表示产生交易记录的电子介质所属的发行方。
MessageId	Long	交易消息包 Id。由收费服务方发送包到清分方时，该字段与消息头的 MessageId 相同。清分方转发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。
ClearTargetDate	Date	清分目标日
Count	Int	本消息包含的记录数量
Amount	Decimal	交易总金额

收费服务方按照电子介质所属的发行方，将原始交易分组打包，发送给清分方。清分方按交易包中指定的发行方将交易提交给对应发行方处理。

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	是由收费服务方产生的该包内顺序 Id，从 1 开始递增。在收费服务方、清分方、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。
CardType	Short	卡类型
NetNo	Hex(4)	网络编码
CardId	Hex(16)	IC 卡物理编号（发行号）。
TransTime	DateTime	交易的发生时间
ServiceType	Short	交易的服务类型，取值见基础信息维护
Fee	Decimal	交易的发生金额
Description	String(100)	对交易的文字解释。如：回龙观北入至清河主出
OriginalData	String	包含所有校验信息在内的原始信息

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。

为减小每个数据包的大小，保证通讯质量，每个交易消息包最多包含10000条交易。若一次需传送的交易数量大于10000条，则需分多个消息传送。

每个交易包中的交易，必须属于同一个清分目标日，不属于同一清分目标日的交易需分别打包。

OriginalData项以字符串方式记录收费交易详细，各字段数据间以“|”作为分隔符。若某字段数据为空，则在前后两个“|”之间不填写任何字符串。

本规范列出的七个字段是OriginalData必须填写的字段，由收费服务方产生，用于发行方做TAC码认证。发行方可以在TAC字段后增加个性化字段定义，包含在OriginalData中一并传送。省（区、市）外的收费服务方并不一定在OriginalData中生成这些个性化字段内容，发行方在处理时应首先确定OriginalData中的个性化字段是否是自己定义的内容。

清分方原则上不处理OriginalData中的数据。

字段组织顺序及内容如下：

序号	字段名	交易终端数据描述	接口数据格式描述
1.	卡类型	无	1：储值卡；2：计帐卡
2.	交易金额	4 字节，交易金额	8 位的 16 进制字符串，例如：11223344
3.	交易类型标识	1 字节，PBOC 定义	2 位的 16 进制字符串，例如：A8
4.	终端机编号	6 字节，即 PSAM 号，PSAM 中 0016 文件中的终端机编号	12 位 16 进制字符串，例如： 0102301AF3D9
5.	终端交易日期	4 字节	YYYY-MM-DD
6.	终端交易时间	3 字节	HH:mm:ss
7.	终端机交易序号	4 字节，PSAM 卡脱机交易序号，在 MAC1 计算过程中得到	8 位 16 进制字符串。仅在卡类型为储值卡时有效，若是计帐卡，此字段为空

储值卡TAC验证字段（上表第2到7项）引自《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》第1、2部分。

记帐卡交易时不生成终端机交易序号，因此对应字段填空值。

#### 5.3.4 发送规则

收费服务方将计算费率后的记录打包生成原始交易包发送给清分方需满足如下两个条件之一：

- I 到达预定义的时间间隔（如10分钟、半小时）；
  - I 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10000条）。
- 清分方仅转发收费服务方的交易数据包。

#### 5.4 记帐处理消息结构

##### 5.4.1 应用范围

记帐消息是由发行方根据清分方传来的收费服务方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。

清分方接到记帐处理后要转发到相应收费服务方。记帐数据的发送方向是：异地发行方→异地清分方→本地清分方→本地收费服务方，三个阶段的传输使用相同的消息结构。

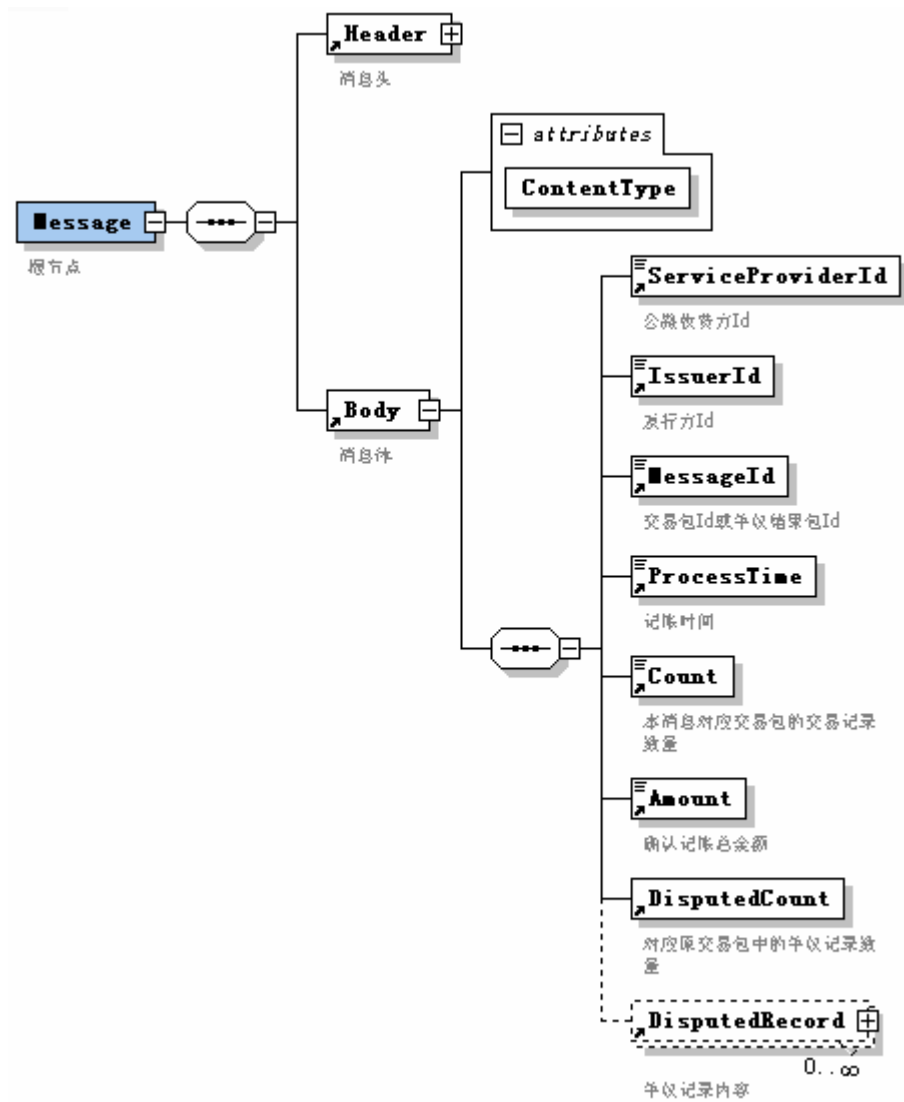
##### 5.4.2 消息头

名称	数据类型	取值或说明
MessageClass	Int	5, Notification
MessageType	Int	5, Reconciliation Totals
SenderId	Hex(16)	发行方 Id/清分方 Id
ReceiverId	Hex(16)	清分方 Id/收费服务方 Id

SenderId与ReceiverId的取值：

发送阶段	SenderId	ReceiverId
发行方→清分方	发行方 Id	清分方 Id
异地清分方→本地清分方	异地清分方 Id	本地清分方 Id
清分方→收费服务方	清分方 Id	收费服务方 Id

5.4.3 消息内容

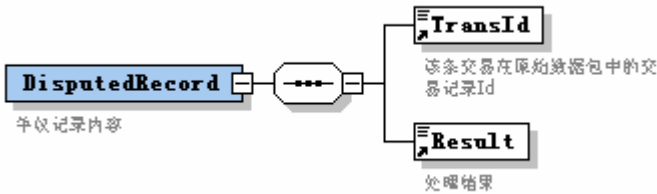


记帐消息的ContentType始终为1，Body各个子节点说明如下：

名称	数据类型	取值或说明
ServiceProviderId	Hex(16)	收费服务方 Id，指明当前记帐消息针对的是哪一个收费服务方
IssuerId	Hex(16)	发行方 Id，说明记帐消息是哪一个发行方产生的
MessageId	Long	表示是针对哪一个收费服务方的交易包。取值为原始交易包消息体中 MessageId 的值
ProcessTime	DateTime	记帐时间
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
DisputdCount	Int	本消息包含的争议交易数量

记帐处理结果仅返回有争议的交易记录明细。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。

争议记录的格式如下：



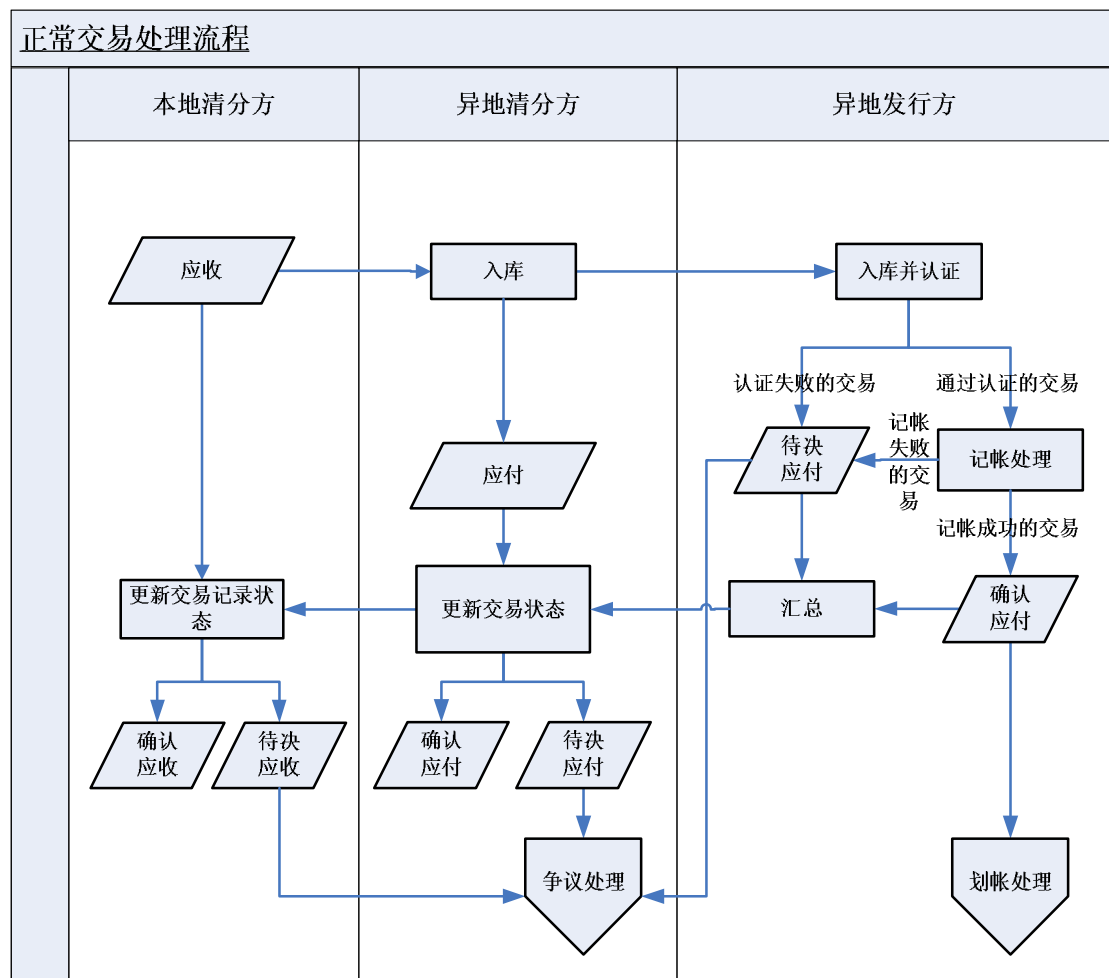
名称	数据类型	取值或说明
TransId	Int	表示该条交易在原始数据包中的交易记录 Id
Result	Short	处理结果，详细解释见下表

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	表示对此交易有疑议，由于用户状态变化拒付（如黑名单交易）
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

取值	说明
100	其它

#### 5.4.4 处理流程



本地清分方与异地清分方之间的数据通过上级清分方转发进行。

发送方发送交易处理的消息类别(MessageClass)的值都为5(通知Notification)，此类消息必须在发送后一段时间内收到接收方发来的MessageClass的值为6(通知应答Notification Response)、消息内容中MessageId与本消息的MessageId相同且Result值为0的确认消息才算正确发送，否则需要重新发送相同的包或根据Result值做相应处理后再发送。

接收方在收到MessageClass值为5的消息后解包并做完整性和正确性校验后给发送方发送MessageClass值为6的确认消息（确认消息结构见确认消息结构）。

- l 本地清分方按照交易的发行方将交易发送给相应的异地发行方做记帐处理。异地清分方在两都转发数据。
- l 发行方对交易认证。未通过认证交易的状态为待决应付，属于争议交易。
- l 发行方根据通过认证交易的明细及用户状态记账，把交易状态更改为待决应付或确认应付。
- l 发行方合并待决交易及确认应付交易结果，发送给清分方。处理结果中的



MessageId 是清分方发来交易包（收费服务方交易包或争议处理结果包）的 MessageId。

- Ⅰ 异地清分方根据发行方提供的记帐处理结果更新交易状态。
- Ⅰ 异地清分方根据 ServiceProviderId 确定收费服务方并将结果发送给相应的本地清分方。
- Ⅰ 收费服务方根据记帐处理结果和认证失败的交易修改原始交易的状态为确认应收或待决应收。
- Ⅰ 对于三方中状态为待决应收或待决应付的交易由三方协商做争议处理。

收费服务方发出的交易包与发行方生成的记帐信息包之间的关系如下：

- Ⅰ 收费服务方的交易包与清分方分组转发给发行方的交易包是一对一的关系。清分方不对原始交易包的内容作任何修改。包与包之间的对应关系由收费服务方交易包的 MessageId 确定。
- Ⅰ 发行方返回的记帐信息包与收费服务方的交易包是一对一关系；发行方返回的记帐信息包与清分方发送的争议处理结果包是一对一关系。以上关系均由包内的 MessageId 确定。即针对每一个包含交易信息的数据包（无论是收费服务方的原始交易包还是清分方的争议处理结果包），发行方均有且只有一个记帐消息与之对应。
- Ⅰ 交易记录之间的关系由 MessageId 和 TransId 确定。

5.5 清分统计消息结构

5.5.1 应用范围

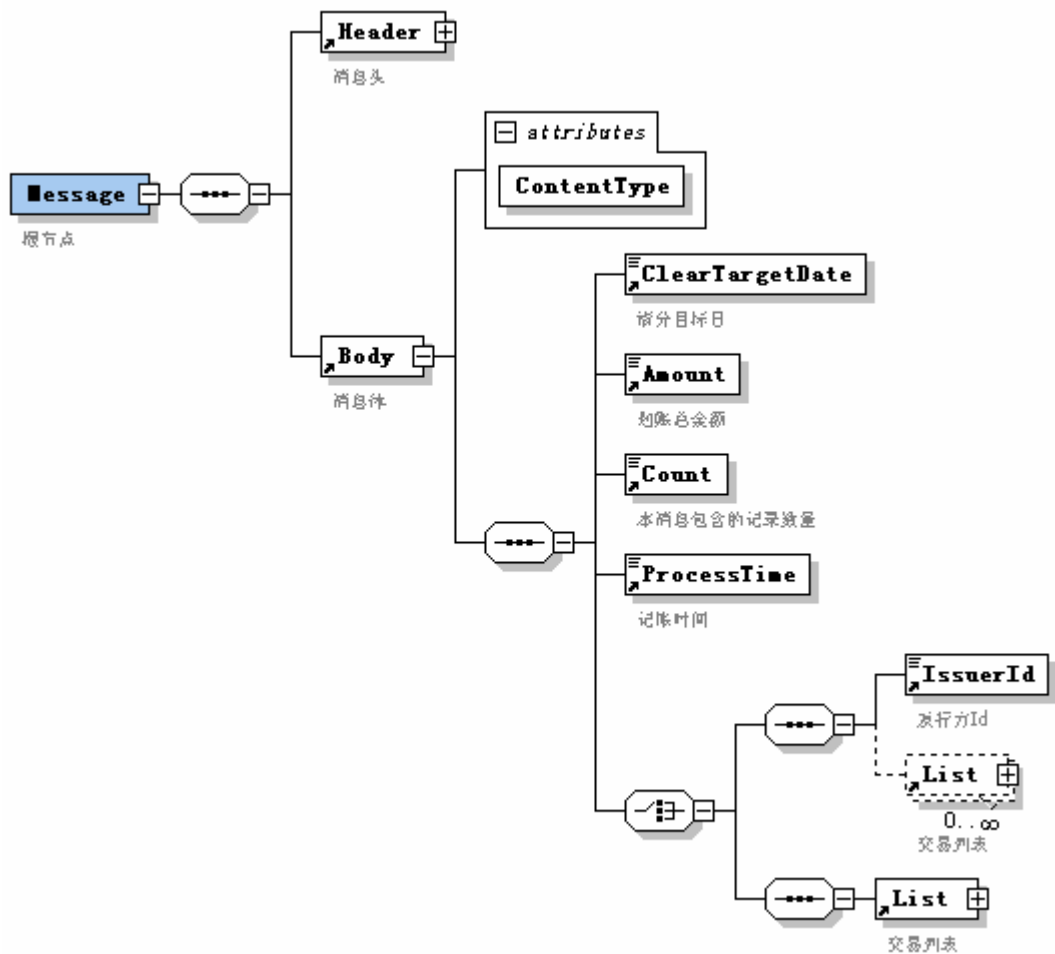
清分方每日定时统计已经发行方确认可以付款的交易，产生清分统计信息，发送给发行方和收费服务方核对。

交易数据的发送方向是：本地清分方→异地清分方。

5.5.2 消息头

名称	数据类型	取值或说明
MessageClass	Int	5, Notification
MessageType	Int	5, Reconciliation Totals
SenderId	Hex(16)	清分方 Id
ReceiverId	Hex(16)	发行方 Id/收费服务方 Id

5.5.3 消息内容

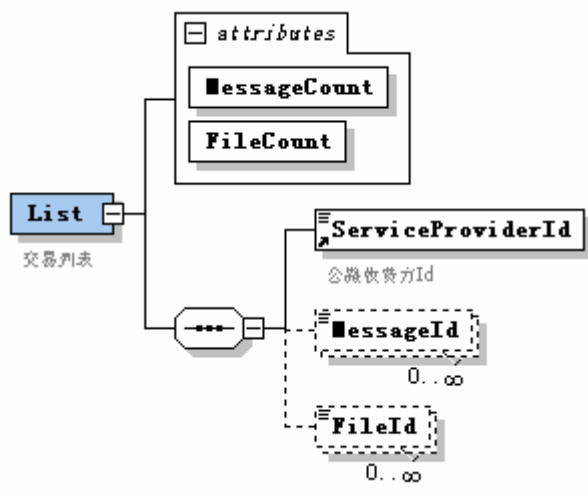


清分消息的MessageClass及MessageType与记帐消息相同。清分消息的ContentType始终为2，与记帐消息相区别。Body各个子节点说明如下：

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
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)	发行方 Id，向收费服务方发送的清分信息中不使用

ProcessTime之后的内容，或者包含包含IssuerId及0或多个List，或者仅包含一个List。前者是清分方为发行方产生的清分统计结果，后者是为收费服务方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	本次清分包含的原始交易包数量	
FileCount	Int	本次清分包含的争议处理结果包数量	
ServiceProviderId	Hex(16)	收费服务方 Id	在系统内唯一确定 原始交易消息包 ID
MessageId	Long	收费服务方发的原始交易包 Id	
FileId	Int	争议处理结果文件 Id	

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。

5.5.4 处理流程

清分方每日定时以前一日为清分目标日进行清分统计，产生的结果发送给发行方和收费服务方核对。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。

5.6 争议交易处理

5.6.1 应用范围

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。

本文所述的争议交易是本地收费服务方与异地发行方之间产生的争议。省（区、市）内的争议仅由本地清分方协调处理，不涉及多个省（区、市）之间的清分方。

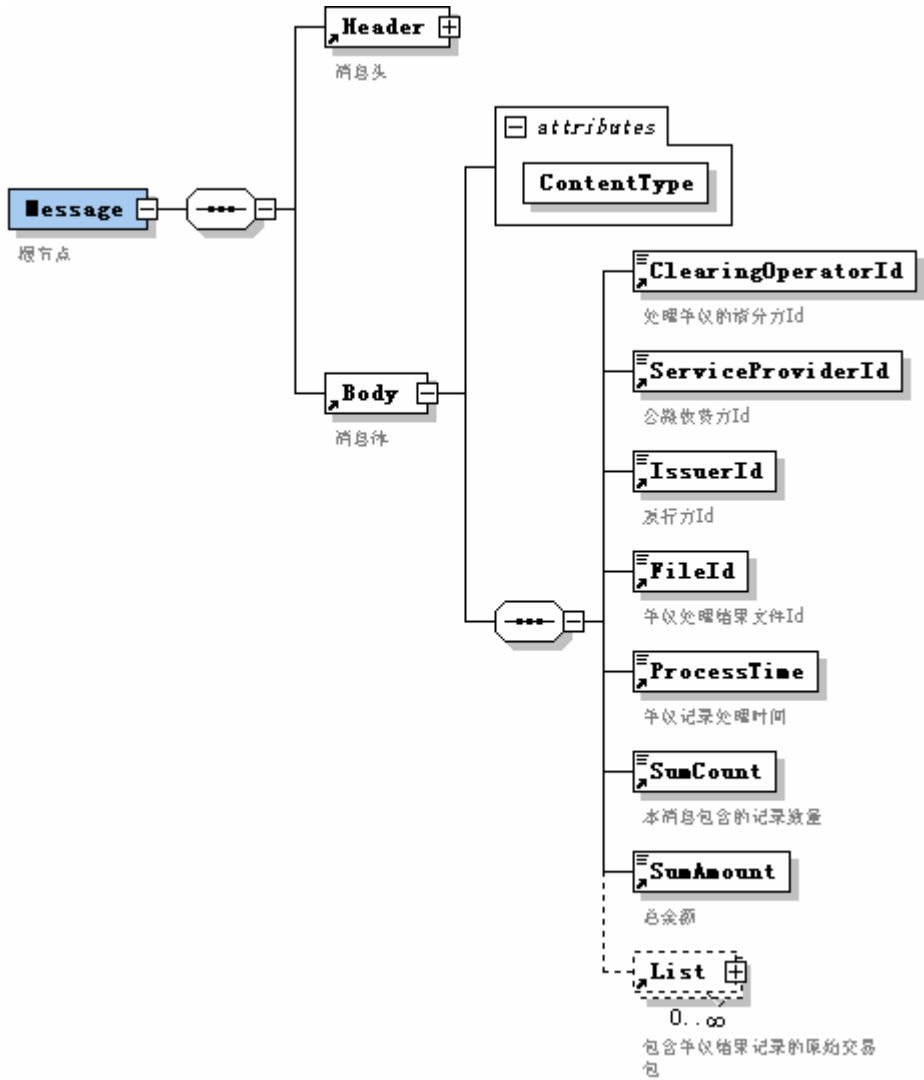
争议交易处理结果的发送方向是：上级清分方→下级清分方→发行方/收费服务方。

5.6.2 消息头

名称	数据类型	取值或说明
MessageClass	Int	5, Notification
MessageType	Int	7, Transaction

名称	数据类型	取值或说明
SenderId	Hex(16)	清分方 Id
ReceiverId	Hex(16)	清分方 Id/收费服务方 Id/发行方 Id

### 5.6.3 消息内容

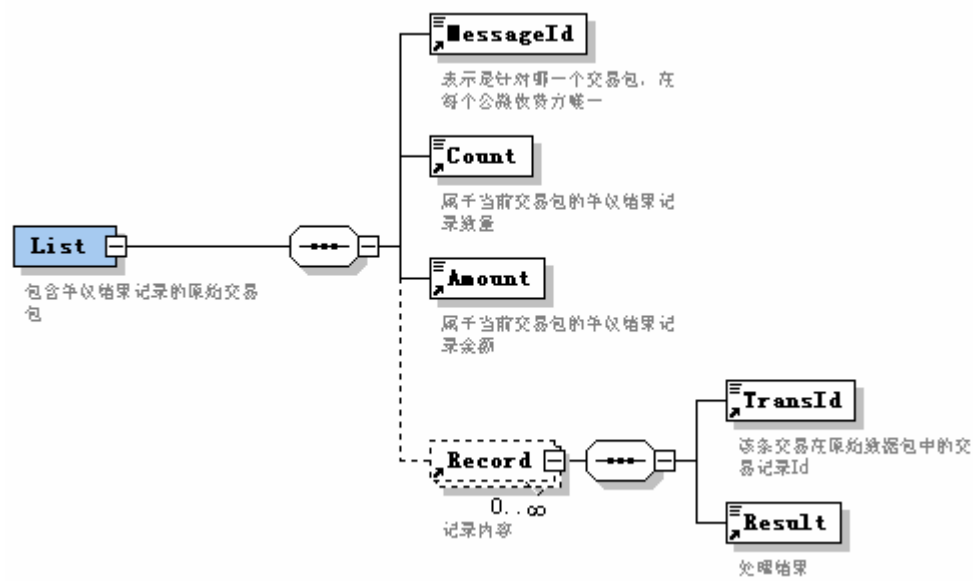


争议记录与交易记录的MessageClass和MessageType相同。争议记录Body的ContentType始终为2，与交易记录的1相区别。Body各个子节点说明如下：

名称	数据类型	取值或说明
ClearingOperatorId	Hex(16)	处理争议的清分方 Id
ServiceProviderId	Hex(16)	收费服务方 Id，表示争议交易是由哪个收费服务方产生的
IssuerId	Hex(16)	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

名称	数据类型	取值或说明
ProcessTime	DateTime	争议处理时间
SumCount	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
SumAmount	Decimal	确认需要记帐的总金额

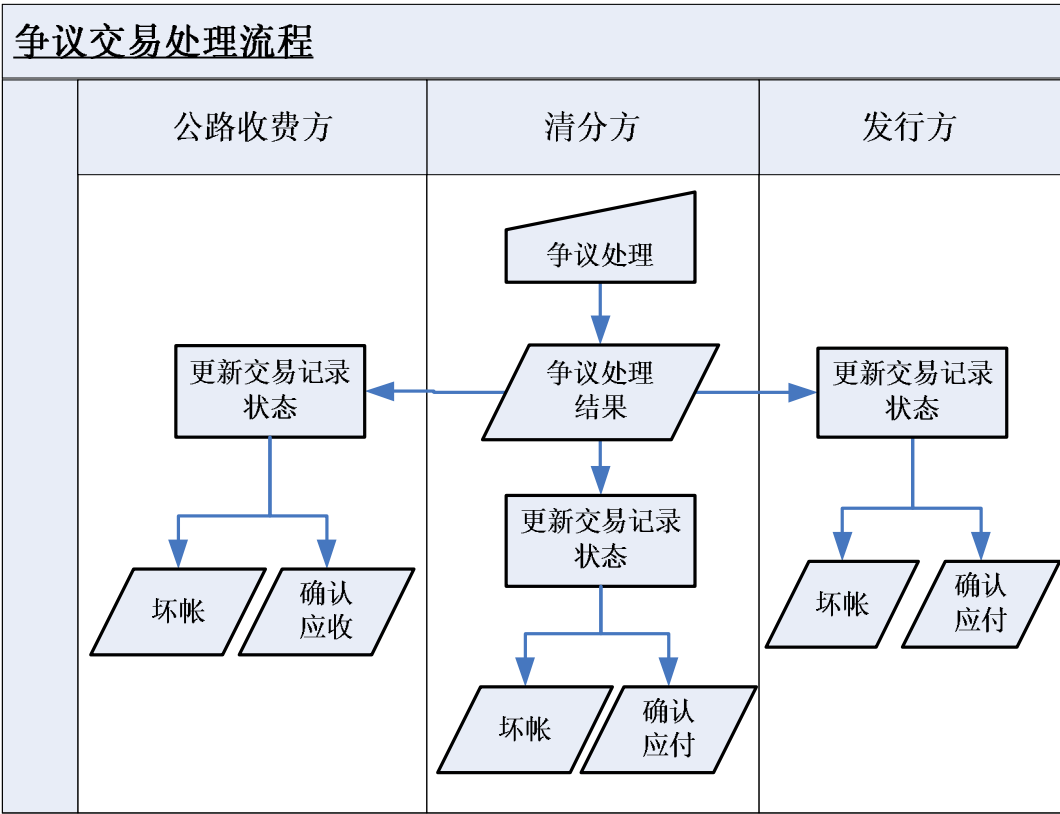
争议交易记录List的格式如下：



名称	数据类型	取值或说明
List		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	收费方原始交易包 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
Record		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。

5.6.4 处理流程



争议交易由清分方、发行方、收费服务方协商后得出处理结果，由清分方负责产生争议交易处理结果数据包并发送给收费服务方和发行方。

每次争议处理，均只能处理一个收费服务方与一个发行方间的争议交易。每次处理可以包含该组收费服务方与发行方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。

由清分方提交争议结果前，清分方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。

通过协商后由清分方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。收费服务方和发行方应使用交易记录号TransId获取交易细节以进一步处理。

发行方和收费服务方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。

5.7 拆卸标签信息处理

5.7.1 应用范围

当收费服务方检测到标签被拆卸后，发送消息到清分中心，由清分中心转发到相应发行方。

标签拆卸信息的发送方向是：收费服务方→清分方，清分方→公路收发行方，两个阶段的传输使用相同的消息结构。

收费服务方发送标签拆卸信息时，按照发行方打包，即一个消息包内只有一个发行方的标签拆卸信息。从原则上讲，同一个标签的拆卸信息只发送一次即可，既包内不应出现两次同一标签的拆卸信息。

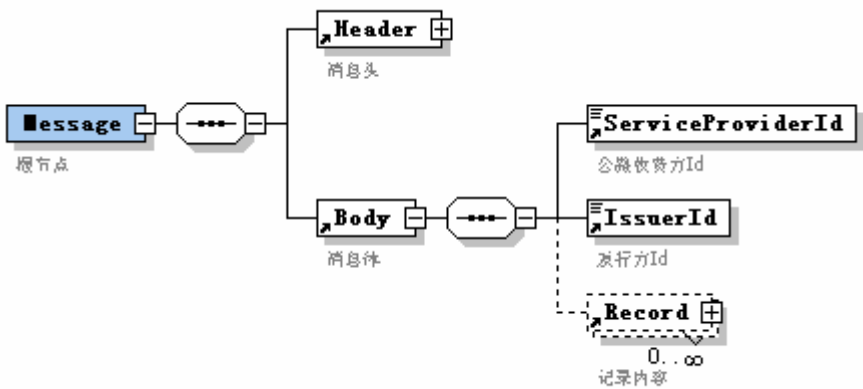
5.7.2 消息头

名称	数据类型	取值或说明
MessageClass	Int	5, Notification
MessageType	Int	17, TagAbuse
SenderId	Hex(16)	清分方 Id
ReceiverId	Hex(16)	收费服务方 Id/发行方 Id

SenderId与ReceiverId的取值：

发送阶段	SendId	ReceiverId
收费服务方→清分方	收费服务方 Id	清分方 Id
清分方→发行方	清分方 Id	发行方 Id

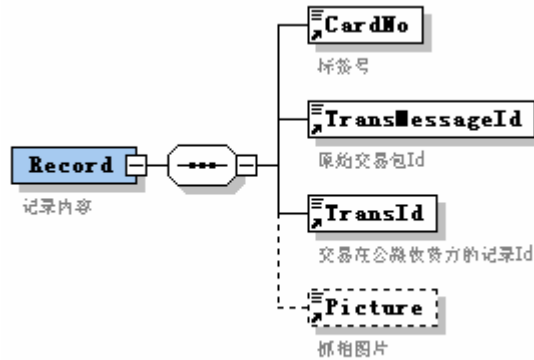
5.7.3 消息内容



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)	收费服务方 Id，表示信息是由哪个收费服务方产生的
IssuerId	Hex(16 位，不足补零)	发行方 Id，表示标签属于哪一个发行方
Record		表示具体的标签拆卸信息

具体的标签拆卸信息记录Record的格式如下：



名称	数据类型	取值或说明
CardNo	Hex(16)	检测到被拆卸的标签的标签号
TransMessageId	Long	检测到被拆卸时的交易所属原始交易包 ID
TransId	Int	在原始交易包中，检测到被拆卸时的原始交易顺序号
Picture	Base64Binary	可选项，如当时有抓拍图片，则将图片以 base64 的格式保存在消息中。

5.8 确认消息结构

5.8.1 应用范围

接收方与发送方之间对交易消息、记帐消息和划帐消息的应答均使用通用确认消息结构。确认消息结构仅表示正确接收，并不包含详细的处理结果。

5.8.2 消息头

名称	数据类型	取值或说明
MessageClass	Int	6, Notification Response
MessageType	Int	7, Transaction 5, Reconciliation Totals
SenderId	Hex(16)	发行方 Id/清分方 Id/收费服务方 Id
ReceiverId	Hex(16)	清分方 Id/收费服务方 Id/发行方

SenderId与ReceiverId的取值：

发送阶段	SenderId	ReceiverId
收费服务方→清分方	收费服务方 Id	清分方 Id
清分方→发行方	清分方 Id	发行方 Id
发行方→清分方	发行方 Id	清分方 Id



发送阶段	SenderId	ReceiverId
清分方→收费服务方	清分方 Id	收费服务方 Id

当确认交易记录及争议时，MessageType为7；确认记帐、划账、对帐单时，MessageType为5；确认标签拆卸信息时，MessageType为17。具体确认的是那一个子类的消息，由Body的ContentType说明。

6 用户状态处理

用户状态名单在接收方处理时不会有延迟，采用建议（Advice）的形式传送。

6.1 发送状态名单

随着用户在系统中使用各种服务，其账户状态不可避免地会发生改变。例如从正常状态变为低值，又从低值变为透支，最后经付费后又恢复为正常等。

为用户提供更好的服务，也为了避免公路收费方系统因用户状态异常而产生不必要的损失，及时地更新用户状态是必要的。

清分系统只对各个发行系统提交的用户状态信息进行转发，不自行修改用户状态信息。

在收到发行方或其它清分方发送过来的用户状态列表后，清分方更新本地数据并立即转发给收费服务方，整个转发过程有详细的日志，并且支持手动指定下发功能。在特殊情况下，可以支持人工使用移动介质传输数据。

6.1.1 发送消息结构

从发行方到清分方再到收费服务方使用相同的记录格式。

6.1.1.1 消息头

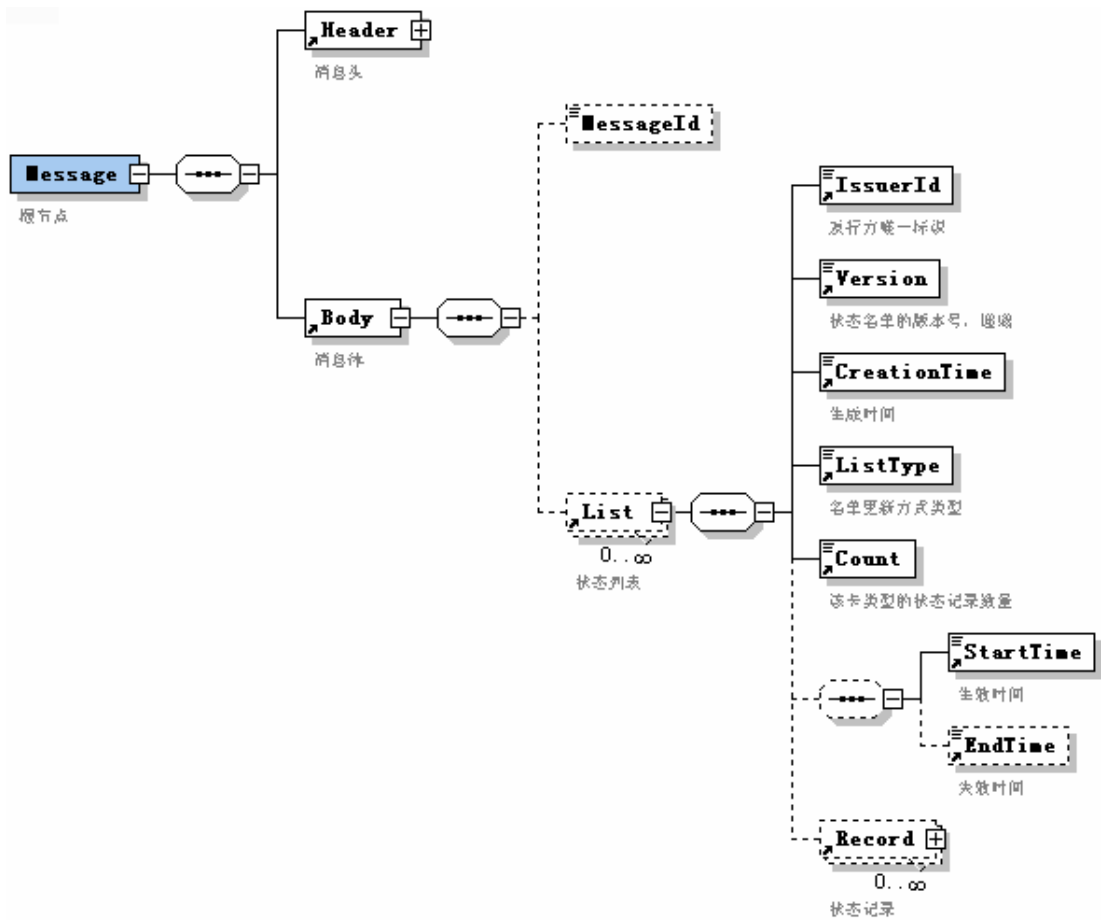
名称	数据类型	取值或说明
MessageClass	Int	3, Advice 或 2, Request Response
MessageType	Int	10, Status List
SenderId	Hex(16)	发送方 Id，可以是发行方或清分方
ReceiverId	Hex(16)	接收方 Id，可以是清分方或收费服务方

SenderId与ReceiverId的取值：

发送阶段	SenderId	ReceiverId
发行方→清分方	发行方 Id	清分方 Id
清分方→收费服务方	清分方 Id	收费服务方 Id

主动发送时MessageClass使用Advice，响应请求重发时使用Request Response。

6.1.1.2 消息内容



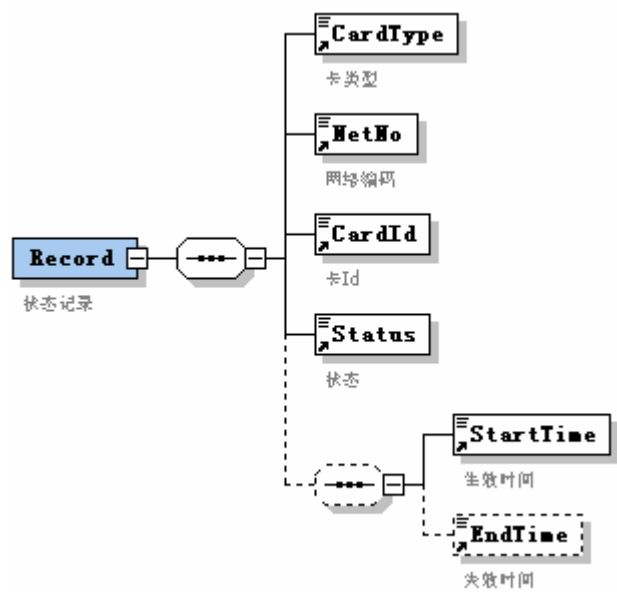
MessageId是可选对象，用于消息接收方回应消息发送方对状态名单的重发请求，表示发送的状态名单回应的是哪一条重发请求消息。参与方主动发送状态名单时不使用该对象。

状态名单消息包含多个状态列表List。每个List对应一个发行方产生的状态名单。通过一个消息，清分方可以将所有发行方产生的状态发送给收费服务方。在逻辑处理上，清分方可以一个一个地发送状态名单，但在数据结构上，消息定义支持一个消息包含多个发行方的状态名单。

List的各个属性说明如下：

名称	数据类型	取值或说明
IssuerId	Hex(16)	发行方唯一标识
Version	Int	状态名单的版本号，递增，由发行方维护
CreationTime	DateTime	发行方生成状态名单的时间
ListType	Short	0 表示整体更新，1 表示增量更新
Count	Int	状态记录数量
StartTime	DateTime	生效时间，对 List 中所有记录均有效。若名单立即生效，此处可填写生成时间。
EndTime	DateTime	失效时间，对 List 中所有记录均有效。若省略表示永不失效。

状态记录结构如下：



状态记录的格式如下：

名称	数据类型	取值或说明
CardType	Short	卡类型
NetNo	Hex(4)	网络编号
CardId	Hex(16)	与 IC 卡物理编号（卡发行号）。
Status	Short	状态类型： 1：正常 2：挂失 3：低值 4：透支 5：禁用 6：拆卸车载单元
StartTime	DateTime	生效时间，若在 Record 节点中使用此项，则 Record 的生效/失效时间覆盖 List 中的生效/失效时间
EndTime	DateTime	失效时间，若省略表示永不失效。

若在List及Record中均未定义生效/失效时间，则认为状态记录立即生效且永不失效。

在数据结构上支持一次性将所有状态名单下发。在实际处理中可以一次只发送一个发行方某一个卡类型的状态名单。提供数据结构上的支持是为了接收方请求重新发送状态名单时处理简单。

6.1.2 确认消息结构

使用通用确认消息结构。

6.1.2.1 应用范围

接收方使用通用确认消息结构建立回应消息告知发送方接收结果。

6.1.2.2 消息头

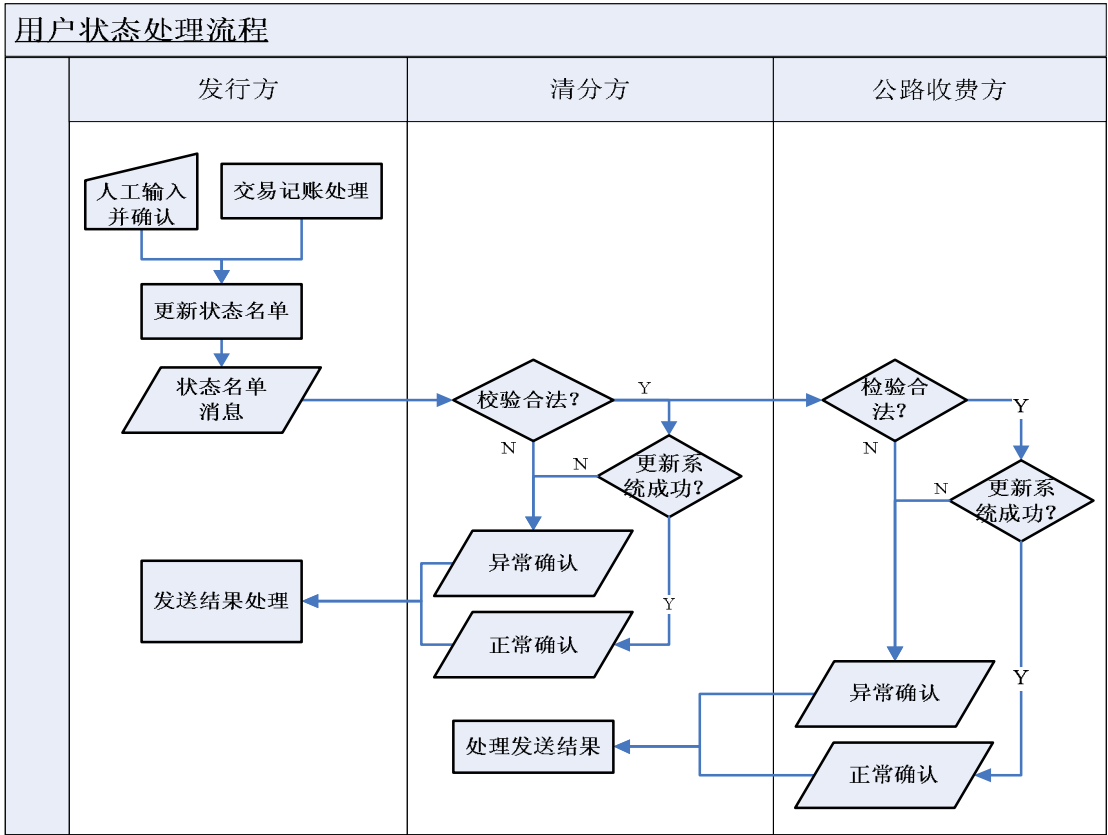
名称	数据类型	取值或说明
MessageClass	Int	4, AdviceResponse
MessageType	Int	10, Status List
SenderId	Hex(16)	发送方 Id
ReceiverId	Hex(16)	接收方 Id

SenderId与ReceiverId的取值：

发送阶段	SendId	ReceiverId
发行方→清分方	发行方 Id	清分方 Id
清分方→收费服务方	清分方 Id	收费服务方 Id

如果收到的状态名单版本号跳号，且本地系统需临时保存名单但暂不处理，则确认消息Result应使用6（接收正确）。

6.1.3 消息处理流程



状态的产生包含人工输入和交易处理自动更新两种。前者是客服系统根据用户的续费、挂失等活动产生的；后者是系统处理交易从用户账户中记帐导致帐户状态变化产生的。

用户状态一旦变化，系统可以立即下发，也可以间隔一定的时间汇总后下发。

发送方式分为整体发送和增量发送两种。

清分方只对各个发行方提交的用户状态信息进行转发，不自行修改用户状态信息。

在收到发行方或其它清分方发送过来的用户状态信息后，系统具备声音提醒功能，并进行消息合法性校验，校验此消息的合法性。如果消息非法，则给发行方返回一条异常确认消息。如果合法，则更新本地系统的用户状态名单，如果更新失败，则给发行方返回一条异常确认消息，更新成功就返回一条正常确认消息。更新成功后，立即将此消息包自动转发给所有收费服务方。

收费服务方在收到清分方转发过来的用户状态名单时，也要进行消息合法性校验，校验消息的合法性，如果消息非法就给清分方返回一条异常确认消息，合法则更新本地系统的用户状态信息，更新用户状态成功就给清分方返回一条正常确认消息，失败则返回一条异常确认消息。

收费服务方在正确更新本地用户状态信息后需立即将更新信息发布到所有的收费节点。

整个转发过程有详细的日志，并且支持手动指定下发功能。在特殊情况下，可以支持人工使用移动介质传输数据。

状态名单的版本管理见前文说明。

6.2 请求重发状态名单

6.2.1 发送消息结构

6.2.1.1 应用范围

发送方请求接收方提供完整的状态名单。

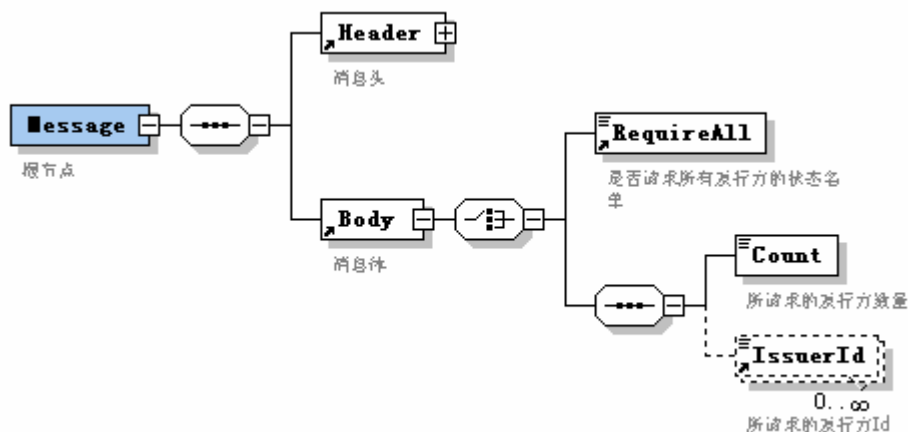
6.2.1.2 消息头

名称	数据类型	取值或说明
MessageClass	Int	1, Request
MessageType	Int	10, Status List
SenderId	Hex(16)	发送方 Id
ReceiverId	Hex(16)	接收方 Id

SenderId与ReceiverId的取值：

发送阶段	SenderId	ReceiverId
收费服务方→清分方	收费服务方 Id	清分方 Id
清分方→发行方	清分方 Id	发行方 Id

6.2.1.3 消息内容



消息体要么包含RequireAll，表示向上一级参与方索取全部发行方产生的状态名单；要么包含一组发行方Id，表示只需要该发行方的状态名单。

名称	数据类型	取值或说明
RequireAll	Boolean	始终为 true
Count	Int	发行方数量
IssuerId	Hex(16)	发行方 Id

### 6.2.2 确认消息结构

接收方生成状态名单后通过状态名单消息对重发状态名单消息进行回应。生成的状态名单必需是全体名单，不能采用增量更新。生成的消息中需使用MessageId节点。

## 7 基础信息维护

基础信息处理相对简单，时间上不会有延迟，所以采用建议（Advice）的形式传送。

### 7.1 服务类型

#### 7.1.1 发送消息结构

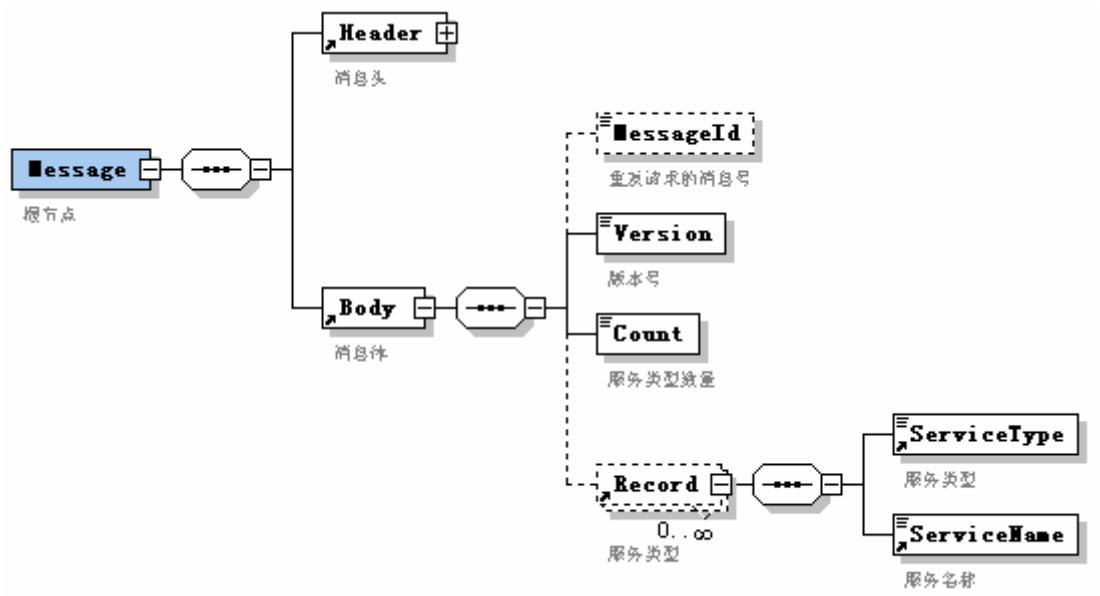
##### 7.1.1.1 应用范围

各个参与方均需使用此类信息。该信息由清分系统生成，传送给其它参与方系统。

##### 7.1.1.2 消息头

名称	数据类型	取值或说明
MessageClass	Int	3, Advice 或 2, Request Response
MessageType	Int	1, Service List
SenderId	Hex(16)	清分方 Id
ReceiverId	Hex(16)	接收方 Id

##### 7.1.1.3 消息内容



各个子节点说明如下：

名称	数据类型	取值或说明
MessageId	Long	当发行方或收费服务方请求重发服务类型时，清分方的回应使用此属性，说明是针对哪一个重发请求的回应；直接发送时不使用该属性
Version	Int	版本号，由最顶层的清分方维护
Count	Int	本操作包含的记录数量
ServiceType	Short	服务类型： 1：公路电子收费 2：停车场 3：加油 4 至 10000：系统保留 10001 以上：在系统实现时可自行定义内部使用的服务类型
ServiceName	String(50)	服务名称字符串

服务类型每次均以整体形式下发。

7.1.2 确认消息结构

使用通用确认消息结构。

7.1.2.1 消息头

名称	数据类型	取值或说明
MessageClass	Int	4，AdviceResponse
MessageType	Int	1，ServiceList
SenderId	Hex(16)	发送方 Id

名称	数据类型	取值或说明
ReceiverId	Hex(16)	接收方 Id

## 7.2 请求重发服务类型

### 7.2.1 发送消息结构

#### 7.2.1.1 应用范围

发行方和收费服务方向清分方请求将服务类型重新下发。

#### 7.2.1.2 消息头

名称	数据类型	取值或说明
MessageClass	Int	1, Request
MessageType	Int	1, Service List
SenderId	Hex(16)	发送方 Id
ReceiverId	Hex(16)	接收方 Id

#### 7.2.1.3 消息内容

使用通用重发请求消息。

### 7.2.2 确认消息结构

接收到重发请求后，清分方将把所有服务类型一次性返回。生成的消息中需使用 MessageId 节点。

## 7.3 参与方信息

所有参与方信息均由清分方设置。清分方仅需将发行方信息下发给收费服务方。收费服务方按照发行方信息对通过认证的IC卡做判断，决定是否提供服务。

其它参与方信息不必下发。参与方信息不必发送给发行方。

### 7.3.1 发送消息结构

#### 7.3.1.1 应用范围

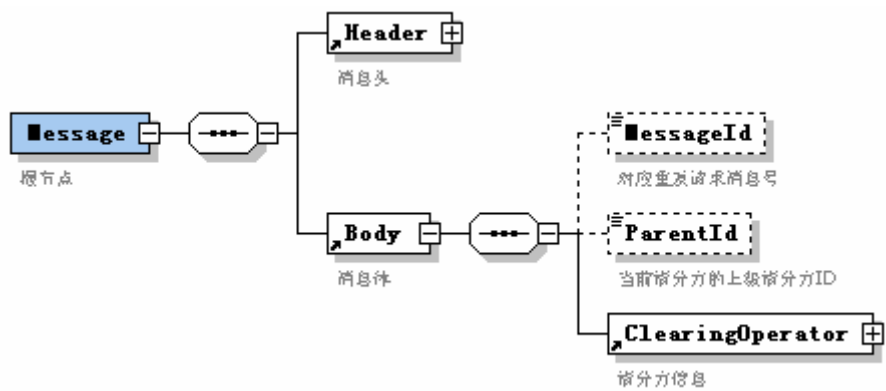
收费服务方和发行方需使用此类信息，收费服务方对通过认证的IC卡进行是否提供服务的判断。该信息由清分系统生成，传送给收费服务方的是发行方的信息，传送给发行方的收费服务方的信息。

#### 7.3.1.2 消息头

名称	数据类型	取值或说明
MessageClass	Int	3, Advice 或 2, Request Response
MessageType	Int	14, OperatorList
SenderId	Hex(16)	清分方 Id
ReceiverId	Hex(16)	收费服务方 Id

#### 7.3.1.3 消息内容



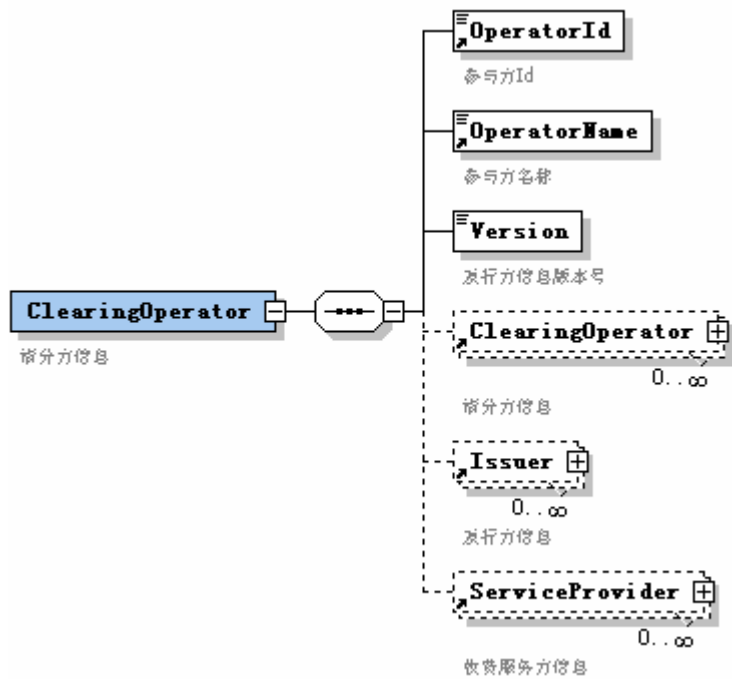


消息结构的各个子节点说明如下：

名称	数据类型	取值或说明
MessageId	Long	用于返回重发请求的消息 ID，主动发送时不使用
ParentId	Hex(16)	生成参与方信息的清分方的上级清分方 ID，如果是最顶层的清分方则省略该项，否则必须包含项
ClearingOperator		生成参与方信息的清分方信息，结构后面详述

通过是否包含ParentId，可以确定消息体中的清分方信息是以顶层清分方为根节点表示的整个电子收费框架中的参与方信息，还是以框架中某一个清分方为根节点的该清分方以下的参与方信息。

清分方信息结构如下：



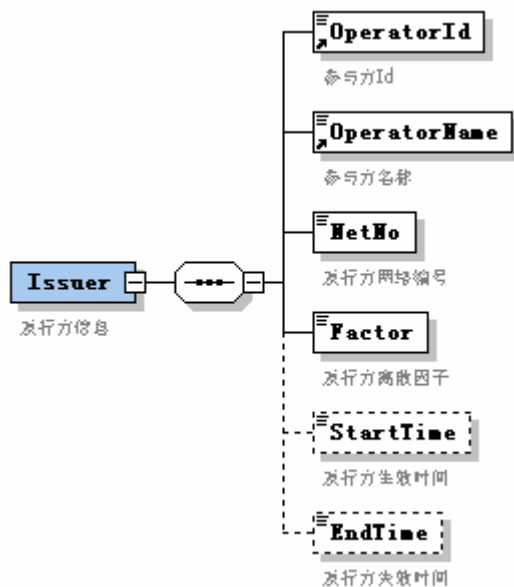
清分方信息结构说明：

名称	数据类型	取值或说明
OperatorId	Hex(16)	清分方 Id

名称	数据类型	取值或说明
OperatorName	String(50)	清分方名称
Version	Int	由 ClearingOperator 节点所表示的清分方维护的该清分方信息的版本号。每个清分方维护自己的版本号
ClearingOperator		与当前清分方具有直接业务关系的下级清分方信息
Issuer		与当前清分方具有直接业务关系的发行方信息
ServiceProvider		与当前清分方具有直接业务关系的收费服务方信息

一个清分方信息，可以包含多个下级清分方信息。通过这种递归结构，可表示以当前清分方为根节点的电子收费柜架结构中的所有参与方信息。

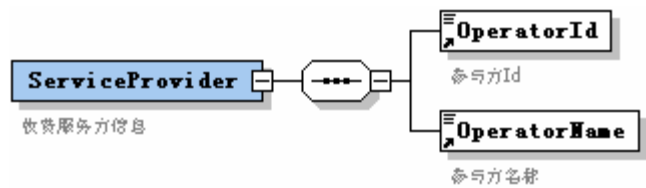
发行方信息结构如下图：



发行方信息结构说明：

名称	数据类型	取值或说明
OperatorId	Hex(16)	发行方 Id
OperatorName	String(50)	发行方名称
NetNo	Hex(4)	发行方网络编号
Factor	Hex(16)	发行方离散因子
StartTime	DateTime	生效时间，若省略，则表示立即生效。生效时间说明新加入系统运营的发行方发行的 IC 卡何时可以在收费服务方使用。
EndTime	DateTime	失效时间，若省略，则表示永不失效

收费服务方信息结构如下图：



收费服务方信息结构说明：

名称	数据类型	取值或说明
OperatorId	Hex(16)	收费服务方 Id
OperatorName	String(50)	收费服务方名称

7.3.2 确认消息结构

使用通用确认消息结构。

7.3.2.1 消息头

名称	数据类型	取值或说明
MessageClass	Int	4, AdviceResponse
MessageType	Int	14, OperatorList
SenderId	Hex(16)	发送方 Id
ReceiverId	Hex(16)	清分方 Id

7.3.3 处理规则

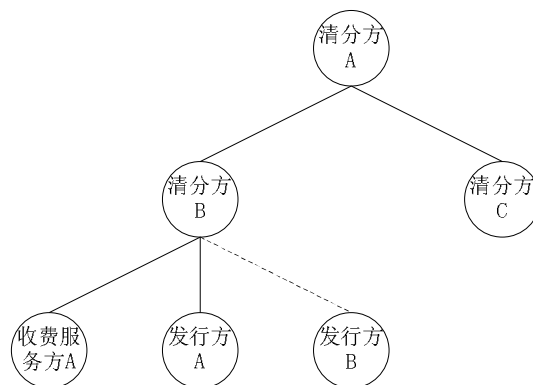
参与方信息每次均以整体形式发送，不支持增量发送。

清分方作为参与方信息的发送方（生成发送或转发）仅发送相关信息给目标参与方：

消息接收方	消息包含的内容
清分方	清分方、收费服务方和发行方信息
收费服务方	清分方、发行方信息
发行方	清分方、收费服务方信息

参与方的消息变更仅由该参与方的直接上级清分方处理。该上级清分方修改其下属参与方信息后，将其下属的所有参与方信息发送给更上一级的清分方以及下属所有相关的参与方。

以下图为例说明处理规则：



- 在清分方B的区域内，新增加发行方B，如虚线所示。
- 发行方B的信息由清分方B生成。
- 清分方B将收费服务方A、发行方A及发行方B的信息被放在同一个消息中，发送给清分方A。
- 清分方A根据该消息更新本地数据后将其转发给清分方C以及其他清分方，以保证清分方B的信息变更传送到整个系统。
- 清分方B将发行方A和发行方B的信息作为一个消息，发送给收费服务方A。
- 清分方B生成以顶层清分方为根节点（本图中为清分方A）的参与方信息消息，发送给发行方B。

## 7.4 请求重发参与方信息

### 7.4.1 发送消息结构

#### 7.4.1.1 应用范围

发行方和收费服务方向清分方请求将参与方信息重新下发。

#### 7.4.1.2 消息头

名称	数据类型	取值或说明
MessageClass	Int	1, Request
MessageType	Int	14, Operator List
SenderId	Hex(16)	发送方 Id
ReceiverId	Hex(16)	接收方 Id

#### 7.4.1.3 消息内容

使用通用重发请求消息。

### 7.4.2 确认消息结构

接收到重发请求后，清分方按以下规则将返回参与方信息。

请求方向	返回消息包含的内容
下级清分方、收费服务方及发行方向直接上级清分方请求重发参与方信息	以顶层清分方为根节点生成参与方信息列表作为返回结果
上级清分方向直接下级清分方请求重发参与方	下级清分方以其自身为根节点生成该下级清

请求方向	返回消息包含的内容
信息	分方的所有下级参与方信息作为返回结果

生成的消息中需使用**MessageId**节点。

(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。

A.1 消息列表

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

消息应用	MessageClass	MessageType	Content-Type
交易消息	5, Notification	7, Transaction	1
争议交易消息	5, Notification	7, Transaction	2
记帐消息	5, Notification	5, Reconciliation Totals	1
清分统计消息	5, Notification	5, Reconciliation Totals	2
标签拆卸信息	5, Notification	17, TagAbuse	
交易通讯确认	6, Notification Response	5, Reconciliation Totals 7, Transaction 17 TagAbuse	
状态名单	3, Advice	10, Status List	
状态名单确认	4, Advice Response	10, Status List	
状态名单重发	1, Request	10, Status List	
状态名单回复	2, Request Response	10, Status List	
服务列表	3, Advice	1, Service List	
服务列表确认	4, Advice Response	1, Service List	
服务列表重发	1, Request	1, Service List	
服务列表回复	2, Request Response	1, Service List	
参与方列表	3, Advice	14, Operator List	
参与方确认	4, Advice Response	14, Operator List	
参与方重发	1, Request	14, Operator List	

消息应用	MessageClass	MessageType	Content-Type
参与方回复	2, Request Response	14, Operator List	

A.2 消息确认对应关系

发送的消息	确认消息
交易消息	交易通讯确认
争议交易消息	交易通讯确认
记帐消息	交易通讯确认
标签拆卸信息	交易通讯确认
清分统计消息	交易通讯确认
状态名单	状态名单确认
状态名单重发	状态名单
服务列表	服务列表确认
服务列表重发	服务列表
参与方列表	参与方确认
参与方重发	参与方列表

---

### 参考文献

- [1] 《北京联网电子收费系统—软件系统设计报告》
- [2] 《道路运输与交通信息技术电子收费（EFC）参与方之间信息交互接口的规范》
- [3] 《北京市高速公路电子收费系统运营管理规则》
- [4] 《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》