

# “码上放心”追溯平台药品追溯码 数据采集器接口标准

3.2 版

码上放心追溯平台

2021 年 8 月

修订历史：

版本	修订内容	修订人
1.0	创建文档	周浩、笑菲
2.0	修订 PDA 部分（往来单位、入出库单据）	谷萸、均昊
3.2	修订支持 PDA 收集收发货人、配送单位、委托单位	均昊

# 目 录

<b>1 概述</b> .....	<b>3</b>
<b>2 适用范围</b> .....	<b>3</b>
<b>3 术语和定义</b> .....	<b>3</b>
<b>4 采集器功能需求</b> .....	<b>4</b>
4.1 业务功能 .....	4
4.1.1 核注入库.....	4
4.1.2 核销出库.....	7
4.1.3 库存盘点.....	错误!未定义书签。
4.1.4 追溯码注销.....	错误!未定义书签。
4.1.5 追溯码替换.....	错误!未定义书签。
4.2 系统功能 .....	10
4.2.1 用户登录.....	10
4.2.2 数据通讯.....	10
4.2.3 系统设置.....	12
4.2.4 数据删除.....	14
<b>5 数据下载接口</b> .....	<b>14</b>
5.1 上游企业出库数据下载接口.....	14
5.2 往来单位列表下载接口.....	15
5.3 货主列表下载接口（第三方物流 PDA 专用） .....	16
5.4 药品信息下载接口.....	17
<b>6 业务数据上传接口</b> .....	<b>18</b>
6.1 接口规范 .....	18
6.1.1 XML Schema 规范.....	19
6.1.2 XML Schema 规范中基础数据属性表.....	29
6.2 接口示例 .....	31
<b>7 文件传输规范</b> .....	<b>38</b>
7.1 文件大小 .....	38
7.2 文件拆分 .....	38

## 1 概述

加入码上放心追溯平台（以下简称“平台”）的企业在进行药品的入库、出库时，需要通过数据采集设备采集药品追溯数据，并上报到码上放心追溯平台。

本文档的内容包括如下两点：

- 1) 文档详细介绍了数据采集设备的功能需求，数据采集设备厂商可以参照功能需求在采集设备上开发相应的功能，满足企业与码上放心追溯平台的数据采集以及数据交互的需求。
- 2) 文档详细定义了数据采集设备与码上放心追溯平台进行数据交互的接口标准，具体如下：
  - ✓ 上游企业出库数据下载接口
  - ✓ 往来单位列表下载接口
  - ✓ 货主列表下载接口
  - ✓ 药品信息下载接口
  - ✓ 业务数据文件上传接口

以上均为文件形式的DataExchange接口。

上游企业出库数据下载接口、往来单位列表下载接口、货主列表下载接口、药品信息下载接口均采用指定的文本文件格式。

业务数据文件上传接口采用 XML1.0 的国际标准文件格式进行数据交换。

本接口标准兼容原药监网的数据采集器接口标准。

## 2 适用范围

- 适用的企业范围  
本文档适用的企业类型包括生产企业、批发企业、第三方物流企业。
- 本标准适用的采集设备范围  
支持任何可扫描药品追溯码的设备

## 3 术语和定义

**药品追溯码：**最小包装单元或最小销售单元的追溯码

**包装追溯码：**外包装（中包装以上）的药品追溯码

**包装级别：**1级为最小销售单元追溯码，即药品追溯码；2级为1级的上级包装码，依次类推。

码上放心客户端：码上放心追溯平台提供的客户端软件，企业可以下载并安装到本地，企业采集药品追溯数据后，可以通过客户端上报到码上放心追溯平台。

**往来单位：**与企业发生业务往来的单位，包含授权关系、药品代理关系、入出库关系、配送关系等。

#### • 关于货主的几点重要说明

货主：委托第三方物流企业进行仓储、药品配送的客户，一般是药品生产企业，也有批发企业、上市许可持有人。物流企业代货主上传的单据（包括与单据相关的数据）归货主所有。（原有概念）

当前版本新增“委托单位”、“配送单位”概念，委托单位即生产企业、批发企业或上市许可持有人，委托下游物流企业进行仓储、配送等工作。配送单位即被委托的物流企业。

取消“代上传”功能，物流企业上传的单据归自己所有。物流企业把委托单位写进单据信息中，作为单据的一个标签。委托单位可在自己的客户端中查询此类单据。

由生产/批发/上市许可持有人企业出库、物流企业负责运输的单据，生产/批发/上市许可持有人企业把配送单位写进单据信息中，作为单据的一个标签。配送单位可在自己的客户端中为此类单据维护温度数据等。

货主不再具有实际业务意义，可理解为单纯的单据所有者。但依然支持用户以老版代货主功能上传的历史单据数据。

## 4 采集器功能需求

### 4.1 业务功能

#### 4.1.1 核注入库

##### 【功能描述】

该功能用于企业入库药品时扫描药品上的追溯码生成入库单据。**需要维护订货单编号、发货单编号、发货单位、发货地址、收货地址、委托单位。应收明细：输入药品信息模糊搜索-选择药品，编辑最小追溯码数量。**

1. 入库操作时，用户需首先需要选择入库种类（生产入库、采购入库、退货入库、调拨入库、零头入库、召回入库、赠品入库、盘盈入库、报废入库、其它入库）。
2. 如果用户选择退货入库、采购入库和调拨入库、且用户下载了上游企业的出库信息用来做入库数据对比。则用户首先需要扫描一个入库的追溯码，终端根据该追溯码显示该药品的上游企业名称、药品名称和生产批号。**并将该药品的出库单作为当前的上游企业出库单号**，然后进行第3步操作。

如果对比追溯码，没有发现任何匹配的上游企业出库单，则提示用户是否继续入库，如果用户选择“确认”，则直接执行第3，如果用户选择“取消”，则重复第2步。

如果用户未下载上游企业出库数据，则直接进行第 3 步操作。

3. 输入企业入库单号，之后根据系统设置该功能是否有附加功能，如果有，则输入附加功能所需关键数据，然后开始扫描追溯码，扫描中，关键数据出现变更时，如生产日期变更，则需要重新输入，然后继续扫描。

为了防止用户输入错误的单号，系统要增加一次校验，当第一次输入确认后提示“请再次输入入库单号”，如果两次输入相同则进行后续操作，如果不同，则提示“两次输入的入库单号不一致，请重新输入”，用户按确认后回到该步骤起始处继续输入单号；如果两次输入相同，则判断输入的入库单号在终端的入库单数据库中是否已存在，如果存在则提示用户“该入库单号已经在 XXXX.XX.XX XX:XX 执行了入库扫描”，同时给出两个选项“重新输入”（默认）和“追加扫描”，如果选择重新输入则回到该步骤起始处，如果选择追加扫描则进行后续操作。

注：

追加扫描操作的意思是指一个单子的货物分多次扫描。

XXXX.XX.XX XX:XX 为时间格式，该时间为该单号起初扫描第一个码的时间

4. 设置药品类型。

药品类型包括：普药类型、特药类型、混合类型，系统自动读取设置中的药品类型；系统设置为普药类型，则当前药品类型为普药类型，系统设置为特药类型，则当前药品类型为特药类型，系统设置为混合类型时，用户需再次选择药品类型（普药类型、特药类型）。

为满足一单据中仅包含一种药品类型的需求，扫描过程中需校验追溯码是否匹配药品类型，如不匹配，则提示用户：“该码不符合当前药品类型，请重新扫描”，如匹配则继续扫描。（见 4.1 校验规则）

4.1 校验规则：校验追溯码类型时，可以根据码长度以及编码规则判断。

4.1.1 码长度—追溯码长度均为 20 位；

4.1.2 标志位

普药：追溯码前两位为标志位--81 或 83 或 84 或 85 或 86 或 87 或 88；

特药：追溯码前两位为标志位--89

5. 用户如选择结束本次入库扫描。则返回输入企业入库单号界面。扫描完成后，软件需要显示此次扫描多少产品，表示本次入库完成。

每次扫描的追溯码软件需判断该追溯码是否已经扫描过。如扫描过则报重码错误，用户确认后返回扫描界面继续扫描。重码判断在相同功能内判断，如生产入库，则在所有入库扫描的码中判断，入库扫描后，如果做出库扫描，则不算重码。

每次扫描追溯码时，系统需要检查剩余追溯码容量，当容量小于设定阈值时，需要报警。

每次扫描追溯码时，系统需要判断扫描进入的追溯码长度是否等于 16 位或 20 位数字，如果不为其中一种情况，则提示用户扫描错误，并重新扫描。

高端 PDA 专用需求：（第三方物流专用）

物流企业输入出库单号时，系统需要显示出当前委托单位名称（非必填）。如企业未设定委托单位，可继续设定。输入方式包括：**1、检索方式**，可通过客户名称、名称拼音简写进行模糊检索；**2、可手工或者扫描输入委托单位的唯一编码**，系统自动检索并进行填写。

**【前置条件】:**

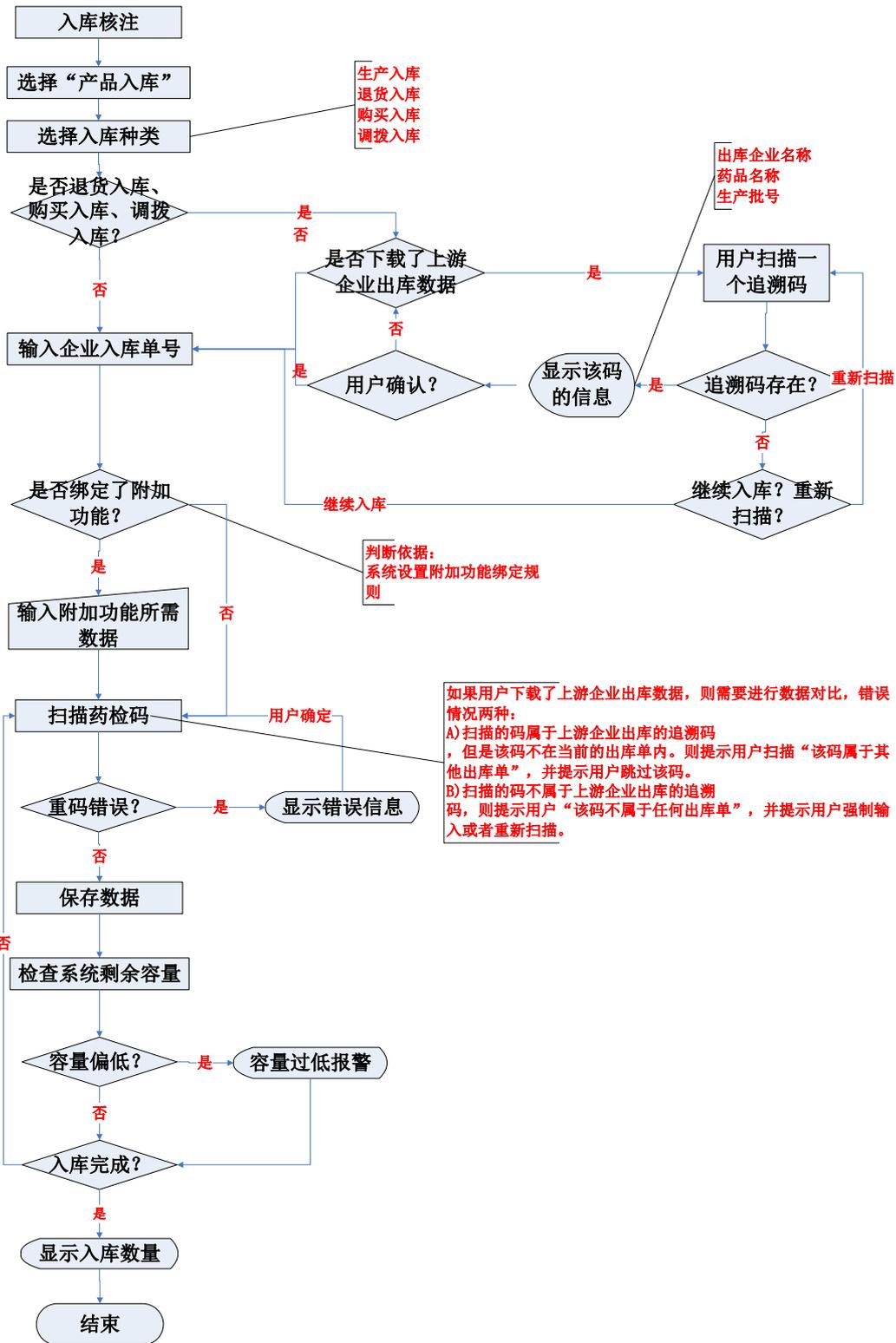
无

**【后置条件】:**

入库数据导出到企业客户端。

**【关键数据】:**

入库种类、企业入库单编号、追溯码、操作员编号、扫描时间、系统设置附加功能所需关键数据、上游企业出库数据。**【业务流程】:**



### 4.1.2 核销出库

**【功能描述】:**

该功能用于企业出库药品时扫描药品上的追溯码生成出库单据。

核销出库功能基本相同核注入库。区别在于用户首先需要选择出库种类（销售出库、退货出库、调拨出库、返工出库、销毁出库、抽检出库、供应出库、直调出库、零头出库、召回出库、赠品出库、盘亏出库、损坏出库、报废出库、其它出库），然后输入的是企业的出货单号。没有上游企业对比功能。

重码判断、系统容量判断、码长度判断同样包括在该功能内（该功能不在流程图中列出）。

高端 PDA 专用需求：（增值服务、第三方物流专用）

企业输入出库单号时，系统提供其输入收货方功能。**需要维护订单编号、收货单位、收货地址、委托单位、配送单位。**

**当出库单中收货方为疾控单位时，委托单位、配送单位为必填项。**

用户可在其往来单位列表中通过模糊检索输入收货方。检索条件：收货方名称关键字，收货方名称拼音头字母简写。也可输入或者扫描收货方唯一编码，系统自动检索并进行输入。**发货方信息输入方式与收货方信息输入方式一致。**

企业输入收货方后，系统须将该收货方编号记录，并写入上传业务数据中，具体位置和格式参加业务数据上传接口。

如企业未导入往来单位列表，该字段可为空，等同于通用核销出库流程。

高端 PDA 专用需求：（第三方物流专用）

物流企业输入出库单号时，系统需要显示出当前货主名称。如企业未设定当前货主，需要企业输入当前货主。输入方式包括：**1、检索方式**，可通过客户名称、名称拼音简写进行模糊检索；**2、可手工或者扫描输入货主的唯一编码**，系统自动检索并进行填写。

**【前置条件】:**

无；

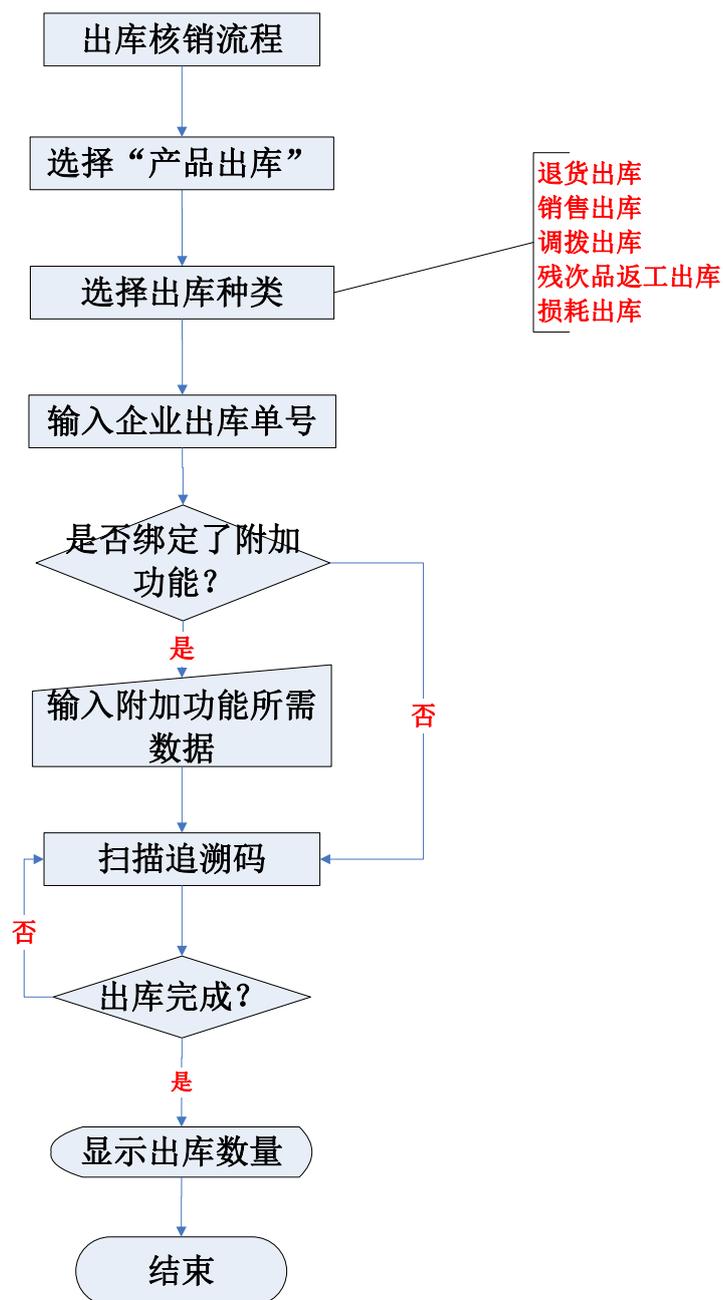
**【关键数据】:**

企业出货单编号、追溯码、操作员编号、扫描时间、系统设置附加功能所需关键数据。

**【关键数据】:**

出库数据上传到 PC 客户端。

**【业务流程图】:**（**出库类型-字典表更新**）



输入企业出库单号步骤：为了防止用户输入错误的单号，在输入单号后系统要增加一次校验，当第一次输入确认后提示“请再次输入出库单号”，如果两次输入相同则进行后续操作，如果不同，则提示“两次输入的单号不一致，请重新输入”，用户按确认后回到该步骤起始处继续输入单号；如果两次输入相同，则判断输入的出库单号在终端的出库单数据库中是否已存在，如果存在则提示用户“该出库单号已经在 xxxx.xx.xx xx:xx 执行了出库扫描”，同时给出两个选项“重新输入”（默认）和“追加扫描”，如果选择重新输入则回到该步骤起始处，如果选择追加扫描则进行后续操作。追加扫描操作的意思是指一个单号的货物分多次扫描。

注：

追加扫描操作的意思是指一个单子的货物分多次扫描。

XXXX.XX.XX XX:XX 为时间格式，该时间为该单号起初扫描第一个码的时间  
设置药品类型。

药品类型包括：普药类型、特药类型、混合类型，系统自动读取设置中的药品类型；

系统设置为普药类型，则当前药品类型为普药类型，系统设置为特药类型，则当前药品类型为特药类型，系统设置为混合类型时，用户需再次选择药品类型（普药类型、特药类型）。

为满足一单据中仅包含一种药品类型的需求，扫描过程中需校验追溯码是否匹配药品类型，如不匹配，则提示用户：“该码不符合当前药品类型，请重新扫描”，如匹配则继续扫描。（见下面的校验规则）

校验规则：校验追溯码类型时，可以根据码长度以及编码规则判断。

码长度—普药、特药码长度均为 20 位；特药包含 16、20 位两种长度

标志位--普药：追溯码前两位为标志位--81 或 83 或 85 或 86 或 87 或 88

特药：16 位特药追溯码第一位为标志位—1；20 位特药追溯码追溯码前两位为标志位—89

## 4.2 系统功能

### 4.2.1 用户登录

#### 【功能描述】:

该功能用于操作员在终端登陆。

### 4.2.2 数据通讯

#### 4.2.2.1 上游企业出库数据下载

##### 【功能描述】:

该功能用于从企业客户端上下载上游企业的出库数据使用。该数据下载后，企业在做入库时，需要先跟下载的上游企业出库数据进行对比，如果发现错误需要提示。

下载接口定义见：5.1 上游企业出库数据下载接口。

##### 【关键数据】:

上游企业名称、上游企业出库单号、追溯码

#### 4.2.2.2 往来单位列表下载

##### 【功能描述】:

该功能用于下载企业在客户端维护的往来单位列表。下载后，企业进行入出库操作时，可以通过检索的方式在往来单位列表中选择上游发货企业或者下游收货企业

下载接口见 5.2 数据下载接口—>往来单位下载接口

**【关键数据】:**

往来单位名称、名称拼音简写、往来单位编码

区别老版与新版，老版如上（不变），新版：往来单位名称、名称拼音缩写、往来单位自定义编码、地址信息（用于创建订单时维护收发货地址）

#### 4.2.2.3 委托单位列表下载（第三方物流企业 PDA 专用）

**【功能描述】:**

该接口用于物流企业从客户端下载委托单位列表到手持终端。用于物流企业进行当前委托单位设置使用。

下载接口见 5.3 数据下载接口—委托单位列表下载接口

**【关键数据】:**

委托单位名称、委托单位名称拼音简写、委托单位编码

#### 4.2.2.4 药品信息下载

**【功能描述】:**

该功能用于企业从客户端下载药品产品编码信息。下载后，企业进行入出库操作时，可以通过检索的方式查询码对应的药品信息。

下载接口见 5.4 数据下载接口—>药品信息下载接口

**【关键数据】:**

药品类型、生产企业、药品通用名、剂型、制剂规格、包装规格、批准文号、商品名、产品编码、包装比例、包装层级

#### 4.2.2.5 企业业务数据上传

**【功能描述】:**

该功能用于将手持终端内的数据整合成规定格式的 XML 文件，然后上传到企业客户端指定的目录。上传成功后，用户可选择清空或者继续保留手持终端数据。

考虑到有可能文件会比较大，传输时间较长。上传控件必须提供检索当前文件传输百分比功能。

上传数据时，需要将数据分成三部分，关联关系相关数据（建立关联、删除关联、重建关联和查验关联）生成一个 XML 文件，普药、特药分别生成不同的 XML 文件。

关联关系数据上传格式见通讯接口—>关联关系数据上传接口

其他数据上传格式见通讯接口—>业务数据上传接口

### 4.2.3 系统设置

#### 4.2.3.1 系统时间设置

**【功能描述】:**

该功能用于用户设定手持终端的系统时间。用户选择该功能后，显示系统当前时间。用户可手工输入时间，更改系统时间。

**【关键数据】:**

系统时间

#### 4.2.3.2 系统最低容量报警设置

**【功能描述】:**

该功能用于每次出库扫描时，检查系统容量，计算出剩余空间可容纳的追溯码数量，显示。当剩余追溯码容量小于系统预警设置的阈值时，提示用户上传并清空数据。

**【关键数据】:**

系统容量，追溯码长度

#### 4.2.3.3 使用企业功能设置

**【功能描述】:**

该功能用于设置当前使用企业，备选项为生产企业、运营企业、使用单位。

根据该项的设置，手持终端需要屏蔽相应的功能项，以免用户误操作。

**生产企业、运营企业、使用单位各自具备的功能项如下表:**

功能点	生产企业	批发企业	使用单位	第三方物流
生产入库	√			√
退货入库	√	√		√
采购入库	√	√	√	√
调拨入库	√	√	√	√
零头入库	√	√	√	√
召回入库	√	√	√	√
赠品入库	√	√	√	√
盘盈入库	√	√	√	√
报废入库	√	√	√	√
其它入库	√	√	√	√

销售出库	√	√	√	√
销毁出库	√	√		√
抽检出库	√	√	√	√
退货出库	√	√	√	√
调拨出库	√	√	√	√
返工出库	√			
供应出库	√	√		√
直调出库	√	√	√	√
零头出库	√	√	√	√
召回出库	√	√	√	√
赠品出库	√	√	√	√
盘亏出库	√	√	√	√
损坏出库	√	√	√	√
报废出库	√	√	√	√
其它出库	√	√	√	√
库存盘点	√	√	√	√
追溯码替换	√	√	√	√
追溯码注销	√	√	√	√
上游企业出库单下载	√	√	√	√
业务数据上传	√	√	√	√
往来单位列表下载	√	√	√	√
委托单位列表下载				√
委托单位设置				√
配送单位设置	√	√	√	√

#### 4.2.3.4 药品类型设置

##### 【功能描述】:

该功能用于设置采集器可支持的药品类型（药品类型包括：普药类型、特药类型、混合

类型), 每次制定单据时系统自动读取该设置。

**【类型说明】:**

- 1 普药类型: 系统设置为普药类型, 创建单据时系统自动设定为普药类型单据, 扫描过程中仅可扫描属于普药类型的码。
- 2 特药类型: 系统设置为特药类型, 创建单据时系统自动设定为特药类型单据, 扫描过程中仅可扫描属于特药类型的码。
- 3 混合类型: 系统设置为混合类型, 创建单据时系统需提示用户设定药品类型 (此时可选类型为: 普药类型、特药类型), 扫描过程中仅可扫描符合当前单据设定的药品码。

**【关键数据】:** 药品类型

#### 4.2.3.5 委托单位设置 (第三方物流 PDA 专用)

**【功能描述】:**

该功能用于物流企业用户选择其当前操作的业务单据属于哪家客户。用户可在客户列表中设定其当前货主。

企业选择当前委托单位的方式包括: 1、检索方式, 可通过客户名称、名称拼音简写进行模糊检索; 2、可手工或者扫描输入货主的唯一编码, 系统自动检索并进行填写。

设定委托单位后, 该客户名称需要 PDA 的显著位置随时可见。

**【关键数据】:**

委托单位名称、委托单位名称拼音简写

#### 4.2.4 数据删除

**【功能描述】:**

该功能用于将用户已经输入的数据。包括所有的主要功能和补充功能数据。

用户删除数据时, 可按照企业的出/入库单批量删除, 也可按照追溯码逐个删除。

删除关联关系数据时, 删除包装码时, 同时删除下属产品码。

删除数据要求可以扫描删除。

## 5 数据下载接口

### 5.1 上游企业出库数据下载接口

**接口描述:**

该接口用于从企业客户端下载上游企业的出库数据使用。

**接口标准:**

\$<企业名称>, <企业出库单号>

<追溯码>, <药品名称>, <生产批号>, <生产日期>

<追溯码>, <药品名称>, <生产批号>, <生产日期>

.....

<追溯码>, <药品名称>, <生产批号>, <生产日期>

#

说明:

\$符号: 出库单开始符

#符号: 出库单结束符

**数据属性:**

名称	类型	长度
企业名称	字符	20 (汉字)
出库单号	字符	20
药品名称	字符	10(汉字)
生产日期	字符	6
生产批号	字符	20
追溯码	字符	16 或 20

**接口示例:**

\$青海制药厂, 10991213-C

123456789012345, 青海药品 1, 1234567890, 060901

123456789012345, 青海药品 1, 1234567890, 060901

#

\$北京制药厂, 293210

123456789012346, 北京药品 1, 1234567890, 060901

123456789022345, 北京药品 1, 1234567890, 060901

#

**5.2 往来单位列表下载接口(需更新)****接口描述:**

该接口用于从客户端下载往来单位列表到手持终端。用于企业入出库时, 选择上游发货

企业或者下游收货企业。文件名称 Customers.txt。

**接口标准:**

<往来单位名称>,<往来单位拼音头字母简写>,<往来单位编码>,<注册地址>,<仓库地址 1>,...<参考地址 n> (回车换行)

<往来单位名称>,<往来单位拼音头字母简写>,<往来单位编码>,<注册地址>,<仓库地址 1>,...<参考地址 n> (回车换行)

.....

<往来单位名称>,<往来单位拼音头字母简写>,<往来单位编码>,<注册地址>,<仓库地址 1>,...<参考地址 n> (回车换行)

**数据属性: (长度定义为最大长度, 不是指定长度)**

名称	类型	长度
往来单位名称	字符	30
往来单位拼音头字母简写	字符	30
往来单位编码	字符	12
注册地址	字符	500
仓库地址 1	字符	500
...		
仓库地址 n	字符	500

**接口示例:**

北京天坛生物制品有限公司, bjttswzpysgs, BJ001,北京市朝阳区,北京市东城区

北京积水潭医院,bjjstyy,00002,北京市朝阳区,北京市东城区

....

上海医药股份有限公司,shyyyxgs, SH230,北京市朝阳区,北京市东城区

### 5.3 委托单位列表下载接口 (第三方物流 PDA 专用)

**接口描述:**

该接口用于物流企业从客户端下载委托单位列表到手持终端。用于企业进行当前委托单位设置使用。文件名称 Owner.txt。

**接口标准:**

<委托单位>,<委托单位拼音头字母简写>,<委托单位编码> (回车换行)

<委托单位>,<委托单位拼音头字母简写>,<委托单位编码> (回车换行)

.....

<委托单位名称>,<委托单位拼音头字母简写>,<委托单位编码> (回车换行)

**数据属性：(长度定义为最大长度，不是指定长度)**

名称	类型	长度
委托单位名称	字符	30
委托单位拼音头字母简写	字符	30
委托单位编码	字符	12

**接口示例：**

北京天坛生物制品有限公司, bjttswzpysgs, BJ001

北京积水潭医院, bjstyy, 00002

....

上海医药股份有限公司, shyyyxgs, SH230

## 5.4 药品信息下载接口

**接口描述：**

该接口用于企业从客户端下载药品产品编码信息到手持终端。下载后，企业进行入出库操作时，可以通过检索的方式查询码对应的药品信息。

文件名称：Drugcode.txt

**接口标准：**

<药品类型>,<生产企业>,<药品通用名><剂型>,<制剂规格>,<包装规格>,<批准文号>,<商品名>,<产品编码>,<包装比例>,<包装级别> (回车换行)

.....

<药品类型>,<生产企业>,<药品通用名><剂型>,<制剂规格>,<包装规格>,<批准文号>,<商品名>,<产品编码>,<包装比例>,<包装级别> (回车换行)

**数据属性：(长度定义为最大长度，不是指定长度)**

名称	类型	长度
药品类型	字符	30
生产企业	字符	60

药品通用名	字符	60
剂型	字符	60
制剂规格	字符	60
包装规格	字符	60
批准文号	字符	60
商品名	字符	120
药品本位码 (SDC)	字符	30
产品编码	字符	12
包装比例	字符	20
包装层级	字符	1

**接口示例:**

第二类精神药品,北京天坛生物制品有限公司,甘草片,片剂, 0.25mg, 袋 200 片, 国药准字 Z1120100999,复方甘草片,000000000000, 89300002,1:10:100,1

....

国家基本药物,北京华兰生物制品有限公司, 注射用青霉素, 注射剂, 10mg, 盒 50 支, 国药准字 Z1120100888, 注射用青霉素,000000000000, 8100001,1:10:50,1

国家基本药物,北京华兰生物制品有限公司, 注射用青霉素, 注射剂, 10mg, 盒 50 支, 国药准字 Z1120100888, 注射用青霉素,000000000000, 8800001,1:10:20,2

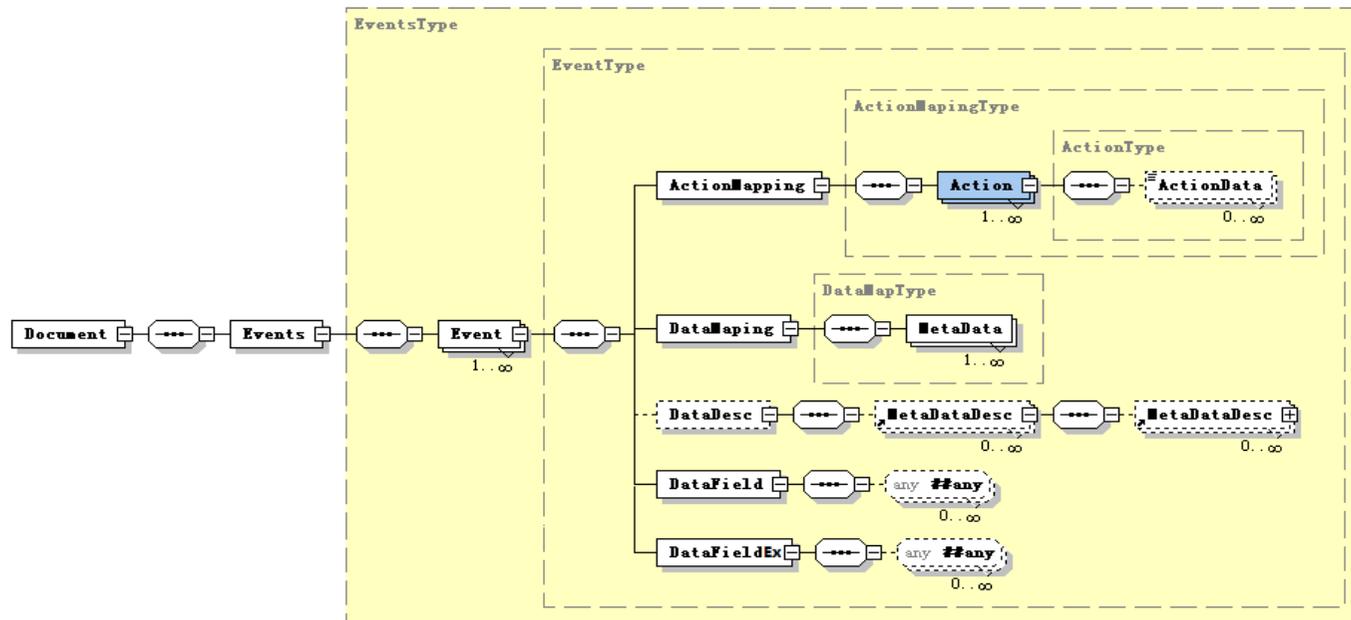
## 6 业务数据上传接口

### 6.1 接口规范

该接口规范定义了手持终端内的除关联关系数据外所有数据上传格式,上传时需要将数据按照接口规范整合成一个 XML 文件,然后上传到企业客户端的制定目录。

### 6.1.1 XML Schema 规范

手持终端生成的业务数据 XML 文件必须符合我方提供的 Schema 规范，结构如下：



各节点注释如下表：

节点名称	节点含义	节点属性	下级节点	举例：
Document	文档根节点	License: 可选	Events	
Events	Document 子节点，只可出现一次，用于区分多种事件产生的数据，如出库事件、码替换事件	Comment 扩展属性，可选	Event ActionMapping、 DataMaping DataDesc DataField	
Event	Events 子节点，定义当前事件发生哪些动作。	Name: 事件名称，可选 PreAction: 前置动作名称，可选 MainAction: 主动作名称，必选 AfterAction: 后置动作名称，可选 Comment: 扩展属性，可选	ActionMapping、 DataMaping、 DataDesc、 DataField	<pre>&lt;Event Name="SalesWareHouseOut" MainAction="WareHouseOut" PreAction=" " AfterAction=" " /&gt;</pre>
ActionMapping	Event 子节点，定义当前事件激发的动作所需要的数据。例如出库动作所需数据		Action	<pre>&lt;ActionMapping&gt;   &lt;Action Name="SalesWareHouseOut"&gt;     &lt;ActionData&gt;CorpOrderID&lt;/ActionData&gt;     &lt;ActionData&gt; Actor&lt;/ActionData&gt;     &lt;ActionData&gt; ActDate&lt;/ActionData&gt;     &lt;ActionData&gt;Code&lt;/ActionData&gt;   &lt;/Action&gt;</pre>
Action	ActionMapping 子节点，可出现多次，定义每一个动作所需	Name: Action 描述的动作名称，必选	ActionData	

	要的数据			</ActionMapping>
ActionData	Action 子节点, 可出现多次, 表示 Action 描述动作所需要要的一个数据名称			
DataMapping	Event 子节点, 当前事件所需所有数据的数据描述和列表	MetaData		<DataMapping> <MetaData Name="CorpOrderID" Type="String"/> <MetaData Name="Actor" Type="String"/> <MetaData Name="ActDate" Type="Date"/> <MetaData Name="Code" Type="String"/> </DataMapping>
MetaData	DataMapping 子节点, 描述每一个数据的名称和属性		Name: 所需数据的名称 Type: 所需数据的类型	
DataDesc	Event 子节点, 描述该事件所需所有数据的关系, 该子节点可选	MetaDataDesc		<DataDesc> <MetaDataDesc MetaName="CorpOrderID" IsParent="true"> <MetaDataDesc MetaName="Code" IsParent="false"/> </MetaDataDesc> <MetaDataDesc MetaName=" ActDate" IsParent="true"> <MetaDataDesc MetaName="Code" IsParent="false"/> </MetaDataDesc> </DataDesc>
MetaDataDesc	DataDesc 子节点, 描述每一个数据与其他数据的关系, 可嵌套自身节点	MetaDataDesc	IsParent: 该数据是否父属性 MetaName: 该数据的名称	
DataField	Event 子节点, 包含具体的数据, 应收药品及数量	具体格式见接口示例	具体格式见接口示例	<DataField> <Code value="123456789012345" CorpOrderID="010101" Actor="test01" ActDate="2006-07-08 14:32:40"/> <Code value="123456789012345" CorpOrderID="010101" Actor="test01"

				ActDate="2006-07-08 14:32:40"/> </DataField>
DataFieldEx	Event 子节点，包含具体的数据	具体格式见接口示例	具体格式见接口示例	<DataFieldEx> <Data CommDrugId="协同平台药品标识" PhysicInfo="药品信息" PkgSpec="包装规格" ProduceEntName="生产企业名" ProduceBatchNo="药品批次号" ProduceDate="生产日期" ExprieDate="有效期至" CodeCount="码数量" SaleCount="最小销售单位数量"/> <Data CommDrugId="" PhysicInfo="" PkgSpec="" ProduceEntName="" ProduceBatchNo="" ProduceDate="" ExprieDate="" CodeCount="" SaleCount=""/> </DataFieldEx>

源代码如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 2 U (http://www.xmlspy.com) by ELEVEN (ELEVEN) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <!-- 文档根节点document -->
  <xs:element name="Document">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Events" type="EventsType"/>
      </xs:sequence>
      <xs:attribute name="SN" type="xs:string" use="optional"/>
      <xs:attribute name="License" type="xs:string" use="optional"/>
      <xs:attribute name="Version" type="xs:string" use="optional"/>
      <xs:attribute name="Comment" type="xs:string" use="optional"/>
      <!-- 二级节点事件节点Events，类型EventsType -->
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
</xs:element>
<!--一级节点Events类型定义-->
<xs:complexType name="EventsType">
  <xs:sequence>
    <xs:element name="Event" type="EventType" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="Comment" type="xs:string"/>
</xs:complexType>
<xs:complexType name="MetaDataDesc">
  <xs:sequence>
    <xs:element name="MetaData" type="DataDescType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DataDescType"/>
<xs:simpleType name="ParentType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Yes"/>
    <xs:enumeration value="No"/>
  </xs:restriction>
</xs:simpleType>
<!--二级节点 Event类型定义-->
<xs:complexType name="EventType">
  <xs:sequence>
    <xs:element name="ActionMapping" type="ActionMappingType"/>
    <xs:element name="DataMapping" type="DataMappingType"/>
    <xs:element name="DataDesc" minOccurs="0">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="DataDescType">
            <xs:sequence>
              <xs:element ref="MetaDataDesc" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="DataField">
```

```
<xs:complexType>
  <xs:complexContent>
    <xs:extension base="DataFieldType">
      <xs:sequence>
        <xs:any namespace="##any" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
</xs:element>
<xs:element name="DataFieldEx">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="DataFieldType">
        <xs:sequence>
          <xs:any namespace="##any" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="Name" type="EventList"/>
<xs:attribute name="PreAction" type="ActionList" use="optional"/>
<xs:attribute name="MainAction" type="ActionList"/>
<xs:attribute name="AfterAction" type="ActionList" use="optional"/>
<xs:attribute name="Comment" type="xs:string" use="optional"/>
</xs:complexType>
<!--二级节点ActionMapping类型定义-->
<xs:complexType name="ActionMappingType">
  <xs:sequence>
    <xs:element name="Action" type="ActionType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<!--三级节点Action类型定义-->
<xs:complexType name="ActionType">
  <xs:sequence>
```

```
<xs:element name="ActionData" type="MetaDataList" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="Name" type="ActionList"/>
</xs:complexType>
<!--二级节点DataMap类型定义-->
<xs:complexType name="DataMapType">
  <xs:sequence>
    <xs:element name="MetaData" type="MetaDataType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<!--三级节点MetaData类型定义-->
<xs:complexType name="MetaDataType">
  <xs:attribute name="Type" type="DataTypeList" use="optional"/>
  <xs:attribute name="Length" type="xs:short" use="optional"/>
  <xs:attribute name="Name" type="MetaDataList" use="required"/>
</xs:complexType>
<!--二级节点数据区DataField类型定义-->
<xs:complexType name="DataFieldType"/>
<xs:element name="MetaDataDesc">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="MetaDataDesc" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="MetaName" type="MetaDataList" use="required"/>
    <xs:attribute name="IsParent" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
<!--可选动作列表-->
<xs:simpleType name="ActionList">
  <xs:restriction base="xs:string">
    <xs:enumeration value="WareHouseIn"/>
    <xs:enumeration value="WareHouseOut"/>
    <xs:enumeration value="CodeReplace"/>
    <xs:enumeration value="CodeDestory"/>
  </xs:restriction>
</xs:simpleType>
<!--可选事件列表-->
```

```
<xs:simpleType name="EventList">
  <xs:restriction base="xs:string">
    <xs:enumeration value="PurchaseWareHouseIn"/>
    <xs:enumeration value="SalesWareHouseOut"/>
    <xs:enumeration value="ReturnWareHouseIn"/>
    <xs:enumeration value="ReturnWareHouseOut"/>
    <xs:enumeration value="ProduceWareHouseIn"/>
    <xs:enumeration value="AllocateWareHouseIn"/>
    <xs:enumeration value="AllocateWareHouseOut"/>
    <xs:enumeration value="ReworkWareHouseOut"/>
    <xs:enumeration value="DestoryWareHouseOut"/>
    <xs:enumeration value="CheckWareHouseOut"/>
    <xs:enumeration value="DirectAllocateWareHouseOut"/>
    <xs:enumeration value="CodeReplace"/>
    <xs:enumeration value="CodeDestory"/>
    <!-- 赠品入库 -->
    <xs:enumeration value="GiftHouseIn"/>
    <!-- 盘盈入库 -->
    <xs:enumeration value="IncomeHouseIn"/>
    <!-- 报废入库 -->
    <xs:enumeration value="ScrapHouseIn"/>
    <!-- 赠品出库 -->
    <xs:enumeration value="GiftHouseOut"/>
    <!-- 盘亏出库 -->
    <xs:enumeration value="LossHouseOut"/>
    <!-- 损坏出库 -->
    <xs:enumeration value="DamageHouseOut"/>
    <!-- 报废出库 -->
    <xs:enumeration value="ScrapHouseOut"/>
    <!-- 其他入库 -->
    <xs:enumeration value="OtherHouseIn"/>
    <!-- 其他出库 -->
    <xs:enumeration value="OtherHouseOut"/>
  </xs:restriction>
</xs:simpleType>
<!-- 可选基础数据列表 -->
<xs:simpleType name="MetaDataList">
```

```
<xs:restriction base="xs:string">
  <xs:enumeration value="CorpOrderID"/>
  <xs:enumeration value="FromCorpID"/>
  <xs:enumeration value="ToCorpID"/>
  <xs:enumeration value="OwnerCorpID"/>
  <xs:enumeration value="Actor"/>
  <xs:enumeration value="ActDate"/>
  <xs:enumeration value="CorpBatchNo"/>
  <xs:enumeration value="Code"/>
  <xs:enumeration value="SourceCode"/>
  <xs:enumeration value="ReplaceCode"/>
  <xs:enumeration value="WrongCode"/>
  <xs:enumeration value="UpperCorpOrderID"/>
  <!--用于满足协同平台标准 -->
  <xs:enumeration value="FromAddress"/>
  <xs:enumeration value="ToAddress"/>
  <xs:enumeration value="FromBillCode"/>
  <xs:enumeration value="OrderCode"/>
  <xs:enumeration value="DisCorpID"/>
  <xs:enumeration value="AssCorpID"/>
  <xs:enumeration value="FromPerson"/>
  <xs:enumeration value="ToPerson"/>
  <!--用于满足协同平台标准，DataFieldEx节点 -->
  <xs:enumeration value="CommDrugId"/>
  <xs:enumeration value="PhysicInfo"/>
  <xs:enumeration value="PkgSpec"/>
  <xs:enumeration value="ProduceEntName"/>
  <xs:enumeration value="CodeCount"/>
  <xs:enumeration value="SaleCount"/>
</xs:restriction>
</xs:simpleType>
<!--基础数据可选数据类型列表-->
<xs:simpleType name="DataTypeList">
  <xs:restriction base="xs:string">
    <xs:pattern value="String|Date|Number|DateTime|Boolean"/>
  </xs:restriction>
</xs:simpleType>
```

</xs:schema>

### 6.1.2 XML Schema 规范中基础数据属性表

	类型	最大长度	举例	备注
企业出/入库单号 (CorpOrderID)	String	20	1234567890; BN-01-20060707	
上游发货企业编号 (FromCorpID)	String	20	10192	增值服务、第三方物流 PDA 使用字段
下游收货企业编号 (ToCorpID)	String	20	10192	增值服务、第三方物流 PDA 使用字段
货主编号 (OwnerCorpID)	String	20	10192	第三方物流（最新版客户端已不支持） 委托单位/货主维护功能 中的自定义编码
操作人 (Actor)	String	12	Test01	
操作时间 (ActDate)	Date	19	2006-07-08 14:32:40	
是否错误码 (WrongCode)	Boolean	5	false、true	

上游企业出库单号 (UpperCorpOrderID)	String	20	1234567890; BN-01-20060707	
手持终端硬件编号	String	30	SN-19293213; 23239323	
发货地址 (FromAddress)	String	50	北京东四环福祿大厦 1 层 305 室	
收货地址 (ToAddress)	String	50	北京东四环福祿大厦 1 层 305 室	
发货单编号 (FromBillCode)	String	20	10001	
订货单编号 (OrderCode)	String	20	BC0003	
运输企业(配送企业) (DisCorpID)	String	20	10192;BJ001	取法和 FromCorpID 或 ToCorpID 一样
委托企业 (AssCorpID)	String	20	10192	委托单位/货主维护功能 中的自定义编码
发货人 (FromPerson)	String	10	张三	
收货人 (ToPerson)	String	10	李四	

协同平台药品标识 (CommDrugId)	String	13	1231234567890	
药品信息 (PhysicInfo)	String	20	盐酸曲马多片 片剂 50mg	
包装规格 (PkgSpec)	String	6	10片/盒	
生产企业名 (ProduceEntName)	String	50	河北制药有限公司	
码数量 (CodeCount)	String	5	100	
最小销售单位数量 (SaleCount)	String	5	200	

## 6.2 接口示例

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="终端接口XML Schema-3.0.xsd"
SN="323232323" Version="3.0" License="00120-100290-002" >
  <Events>
    <!--经营企业普通销售出库，收货方ID默认为空-->
    <Event Name="SalesWareHouseOut" MainAction="WareHouseOut" PreAction="" AfterAction="" >
      <ActionMapping>
        <Action Name="WareHouseOut">
          <ActionData>CorpOrderID</ActionData>
        </Action>
      </ActionMapping>
    </Event>
  </Events>
</Document>
```

```

        <ActionData>Actor</ActionData>
        <ActionData>ActDate</ActionData>
        <ActionData>ToCorpID</ActionData>
        <ActionData>Code</ActionData>
    </Action>
</ActionMapping>
<DataMapping>
    <MetaData Name="CorpOrderID" Type="String"/>
    <MetaData Name="Actor" Type="String"/>
    <MetaData Name="ActDate" Type="Date"/>
    <MetaData Name="ToCorpID" Type="String"/>
    <MetaData Name="Code" Type="String"/>
</DataMapping>
<DataDesc>
    <MetaDataDesc MetaName="Actor" IsParent="true">
        <MetaDataDesc MetaName="Code" IsParent="false"/>
    </MetaDataDesc>
    <MetaDataDesc MetaName="CorpOrderID" IsParent="true">
        <MetaDataDesc MetaName="Code" IsParent="false"/>
    </MetaDataDesc>
</DataDesc>
<DataField>
    <Data Code="123456789012345" CorpOrderID="010101" Actor="test01" ActDate="2006-07-08 14:32:40" ToCorpID=""
FromAddress="望京A地下一层" ToAddress="望京A地下二层" OrderCode="C001" DisCorpID="BJ001" FromPerson="张三" />
    <Data Code="123456789012345" CorpOrderID="010101" Actor="test01" ActDate="2006-07-08 14:32:40" ToCorpID=""
FromAddress="望京A地下一层" ToAddress="望京A地下二层" OrderCode="C001" DisCorpID="BJ001" FromPerson="张三" />
</DataField>
</Event>
<!--经营企业使用增值服务终端销售出库-->
<Event Name="SalesWarehouseOut" MainAction="WareHouseOut" PreAction="" AfterAction="" >
    <ActionMapping>
        <Action Name="WareHouseOut">
            <ActionData>CorpOrderID</ActionData>
            <ActionData>Actor</ActionData>
            <ActionData>ActDate</ActionData>
            <ActionData>ToCorpID</ActionData>
            <ActionData>Code</ActionData>
        </Action>
    </ActionMapping>

```

```
</Action>
</ActionMapping>
<DataMapping>
  <MetaData Name="CorpOrderID" Type="String"/>
  <MetaData Name="Actor" Type="String"/>
  <MetaData Name="ActDate" Type="Date"/>
  <MetaData Name="ToCorpID" Type="String"/>
  <MetaData Name="Code" Type="String"/>
</DataMapping>
<DataDesc>
  <MetaDataDesc MetaName="Actor" IsParent="true">
    <MetaDataDesc MetaName="Code" IsParent="false"/>
  </MetaDataDesc>
  <MetaDataDesc MetaName="CorpOrderID" IsParent="true">
    <MetaDataDesc MetaName="Code" IsParent="false"/>
  </MetaDataDesc>
</DataDesc>
<DataField>
  <Data Code="123456789012345" CorpOrderID="010101" Actor="test01" ActDate="2006-07-08 14:32:40" ToCorpID="BJ001"
  FromAddress="望京A地下一层" ToAddress="望京A地下二层" OrderCode="C001" DisCorpID="BJ001" FromBillCode="C001" AssCorpID="BJ001"
  FromPerson="张三" />
  <Data Code="123456789012345" CorpOrderID="010101" Actor="test01" ActDate="2006-07-08 14:32:40" ToCorpID="BJ002"
  FromAddress="望京A地下一层" ToAddress="望京A地下二层" OrderCode="C001" DisCorpID="BJ001" FromBillCode="C001" AssCorpID="BJ001"
  FromPerson="张三" />
</DataField>
</Event>
<!--第三方物流企业PDA销售出库-->
<Event Name="SalesWareHouseOut" MainAction="WareHouseOut" PreAction="" AfterAction="" >
  <ActionMapping>
    <Action Name="WareHouseOut">
      <ActionData>CorpOrderID</ActionData>
      <ActionData>Actor</ActionData>
      <ActionData>ActDate</ActionData>
      <ActionData>ToCorpID</ActionData>
      <ActionData>OwnerCorpID</ActionData>
      <ActionData>Code</ActionData>
    </Action>
  </ActionMapping>
</Event>
```

```

</ActionMapping>
<DataMapping>
  <MetaData Name="CorpOrderID" Type="String"/>
  <MetaData Name="Actor" Type="String"/>
  <MetaData Name="ActDate" Type="Date"/>
  <MetaData Name="ToCorpID" Type="String"/>
  <MetaData Name="OwnerCorpID" Type="String"/>
  <MetaData Name="Code" Type="String"/>
</DataMapping>
<DataDesc>
  <MetaDataDesc MetaName="Actor" IsParent="true">
    <MetaDataDesc MetaName="Code" IsParent="false"/>
  </MetaDataDesc>
  <MetaDataDesc MetaName="CorpOrderID" IsParent="true">
    <MetaDataDesc MetaName="Code" IsParent="false"/> FromBillCode
  </MetaDataDesc>
</DataDesc>
<DataField>
  <Data Code="123456789012345" CorpOrderID="010101" Actor="test01" ActDate="2006-07-08 14:32:40" ToCorpID="BJ001"
OwnerCorpID="BJ001" FromAddress="望京A地下一层" ToAddress="望京A地下二层" OrderCode="C001" FromBillCode="C001" AssCorpID="BJ001"
FromPerson="张三" />
  <Data Code="123456789012345" CorpOrderID="010101" Actor="test01" ActDate="2006-07-08 14:32:40" ToCorpID="BJ001"
OwnerCorpID="BJ001" FromAddress="望京A地下一层" ToAddress="望京A地下二层" OrderCode="C001" FromBillCode="C001" AssCorpID="BJ001"
FromPerson="张三" />
</DataField>
</Event>
<!--批发(物流)企业普通采购入库举例，该举例适用于采购入库、调拨入库、退货入库。生产企业生产入库按照上游企业出库单不存在处理,FromCorpID为空-->
<Event Name="PurchaseWareHouseIn" MainAction="WareHouseIn" >
  <ActionMapping>
    <Action Name="WareHouseIn">
      <ActionData>CorpOrderID</ActionData>
      <ActionData>FromCorpID</ActionData>
      <ActionData>WrongCode</ActionData>
      <ActionData>UpperCorpOrderID</ActionData>
      <ActionData>Actor</ActionData>
      <ActionData>ActDate</ActionData>
      <ActionData>Code</ActionData>
    </Action>
  </ActionMapping>
</Event>

```

```
</Action>
</ActionMapping>
<DataMapping>
  <MetaData Name="CorpOrderID" Type="String"/>
  <MetaData Name="FromCorpID" Type="String"/>
  <MetaData Name="Actor" Type="String"/>
  <MetaData Name="WrongCode" Type="Boolean"/>
  <MetaData Name="UpperCorpOrderID" Type="String"/>
  <MetaData Name="ActDate" Type="Date"/>
  <MetaData Name="Code" Type="String"/>
</DataMapping>
<DataField>
  <!--正常入库扫描-->
  <Data Code="123456789012345" CorpOrderID="010101" Actor="test01" ActDate="2006-07-08 14:32:40" WrongCode="false"
UpperCorpOrderID="020101" FromCorpID="" FromAddress="望京A地下一层" ToAddress="望京A地下二层" FromBillCode="b001" OrderCode="C001"
DisCorpID="BJ001" AssCorpID="BJ002" ToPerson="张三" />
  <Data Code="123456789012346" CorpOrderID="010101" Actor="test01" ActDate="2006-07-08 14:32:40" WrongCode="false"
UpperCorpOrderID="020101" FromCorpID="" FromAddress="望京A地下一层" ToAddress="望京A地下二层" FromBillCode="b001" OrderCode="C001"
DisCorpID="BJ001" AssCorpID="BJ002" ToPerson="张三" />
  <!--入库扫描时，对比数据发现追溯码不存在，由用户选择强制输入时，数据记录-->
  <Data Code="123456789012347" CorpOrderID="010101" Actor="test01" ActDate="2006-07-08 14:32:40" WrongCode="true"
UpperCorpOrderID="020101" FromCorpID="" FromAddress="望京A地下一层" ToAddress="望京A地下二层" FromBillCode="b001" OrderCode="C001"
DisCorpID="BJ001" AssCorpID="BJ002" ToPerson="张三" />
  <!--入库扫描时，上游企业出库单不存在，或者用户没有下载上游企业出库单做对比的情况下，数据记录-->
  <Data Code="123456789012348" CorpOrderID="010102" Actor="test01" ActDate="2006-07-08 14:32:40" WrongCode="false"
UpperCorpOrderID="" FromCorpID="" ToPerson="张三" />
  <Data Code="123456789012349" CorpOrderID="010102" Actor="test01" ActDate="2006-07-08 14:32:40" WrongCode="false"
UpperCorpOrderID="" FromCorpID="" ToPerson="张三" />
  <Data Code="123456789012340" CorpOrderID="010102" Actor="test01" ActDate="2006-07-08 14:32:40" WrongCode="false"
UpperCorpOrderID="" FromCorpID="" ToPerson="张三" />
</DataField>
<!--应收数量-->
<DataFieldEx>
  <Data CommDrugId="协同平台药品标识" PhysicInfo="药品信息" PkgSpec="包装规格" ProduceEntName="生产企业名" CodeCount="码数量" SaleCount="
最小销售单位数量"/>
  <Data CommDrugId="" PhysicInfo="" PkgSpec="" ProduceEntName="" CodeCount="" SaleCount="" />
</DataFieldEx>
</DataField>
</DataMapping>
</ActionMapping>
</Action>
```

```

</DataFieldEx>

</Event>
<!--增值服务采购入库举例，该举例适用于采购入库、调拨入库、退货入库。生产企业生产入库按照上游企业出库单不存在处理-->
<Event Name="PurchaseWareHouseIn" MainAction="WareHouseIn" >
  <ActionMapping>
    <Action Name="WareHouseIn">
      <ActionData>CorpOrderID</ActionData>
      <ActionData>FromCorpID</ActionData>
      <ActionData>OwnerCorpID</ActionData>
      <ActionData>WrongCode</ActionData>
      <ActionData>UpperCorpOrderID</ActionData>
      <ActionData>Actor</ActionData>
      <ActionData>ActDate</ActionData>
      <ActionData>Code</ActionData>
    </Action>
  </ActionMapping>
  <DataMapping>
    <MetaData Name="CorpOrderID" Type="String"/>
    <MetaData Name="FromCorpID" Type="String"/>
    <MetaData Name="OwnerCorpID" Type="String"/>
    <MetaData Name="Actor" Type="String"/>
    <MetaData Name="WrongCode" Type="Boolean"/>
    <MetaData Name="UpperCorpOrderID" Type="String"/>
    <MetaData Name="ActDate" Type="Date"/>
    <MetaData Name="Code" Type="String"/>
  </DataMapping>
  <DataField>
    <!--正常入库扫描-->
    <Data Code="123456789012345" CorpOrderID="010101" Actor="test01" ActDate="2006-07-08 14:32:40" WrongCode="false"
UpperCorpOrderID="020101" FromCorpID="BJ001" OwnerCorpID="BJ001" ToPerson="张三" />
    <Data Code="123456789012346" CorpOrderID="010101" Actor="test01" ActDate="2006-07-08 14:32:40" WrongCode="false"
UpperCorpOrderID="020101" FromCorpID=" BJ001" OwnerCorpID="BJ001" ToPerson="张三" />
    <!--入库扫描时，对比数据发现追溯码不存在，由用户选择强制输入时，数据记录-->
    <Data Code="123456789012347" CorpOrderID="010101" Actor="test01" ActDate="2006-07-08 14:32:40" WrongCode="true"
UpperCorpOrderID="020101" FromCorpID=" BJ001" OwnerCorpID="BJ001" ToPerson="张三" />
    <!--入库扫描时，上游企业出库单不存在，或者用户没有下载上游企业出库单做对比的情况下，数据记录-->

```

```
<Data Code="123456789012348" CorpOrderID="010102" Actor="test01" ActDate="2006-07-08 14:32:40" WrongCode="false"
UpperCorpOrderID="" FromCorpID=" BJ001" OwnerCorpID="BJ001" ToPerson="张三" />
<Data Code="123456789012349" CorpOrderID="010102" Actor="test01" ActDate="2006-07-08 14:32:40" WrongCode="false"
UpperCorpOrderID="" FromCorpID=" BJ001" OwnerCorpID="BJ001" ToPerson="张三" />
<Data Code="123456789012340" CorpOrderID="010102" Actor="test01" ActDate="2006-07-08 14:32:40" WrongCode="false"
UpperCorpOrderID="" FromCorpID=" BJ001" OwnerCorpID="BJ001" ToPerson="张三" />
</DataField>
</Event>
</Events>
</Document>
```

## 7 文件传输规范

### 7.1 文件大小

文件大小必须在 4M 以内。

所有向码上放心追溯平台提交的数据文件，其文件大小要小于或等于 4M，否则码上放心追溯平台不予接收。

### 7.2 文件拆分

对于文件内容超出限制的情况，原则上首先根据数据的业务关系进行拆分，如将一个大包装内的所有关联数据放置在一个文件包中，如一个大包装的关联数据文件大于 4M，则可将该大包装的数据拆分为两个文件上传。其次对于已经拆分后的数据文件，其文件大小仍超出限制的情况，需由文件传输软件进行拆分，保证文件大小满足要求。

拆分文件时，生成的文件名必须包括拆分顺序编号和总文件数，如原文件名\_总文件数\_拆分顺序编号。