

顶尖标签秤及触控秤

编程手册

V2.0

2018/11/14

目录

一、简介	1
二、动态连接库接口	2
2.1 动态连接库调用流程	2
2.2 动态连接库函数说明	3
三、命令行接口	6
3.1 命令行参数说明	6
3.2 命令行使用范例	6
四、类型定义	8
4.1 协议类型 (ProtocolType)	8
4.2 操作类型 (ProcType)	8
4.3 数据类型 (DataType)	8
4.4 错误码 (ErrorCode)	9
4.5 设备信息结构定义 (TASSDKDeviceInfo)	9
五、数据文件格式	11
5.1 文件格式说明	11
5.2 文件范例	11
5.3 文件编码格式	11
5.4 文件格式详解	11
5.4.1 单品文件 (PLU)	12
5.4.2 备注文件 (Note1/2/3/4)	14
5.4.3 部门文件 (Department)	15
5.4.4 分组文件 (Group)	15
5.4.5 热键文件 (HotKey)	15
5.4.6 折扣排程文件 (Discount)	16
5.4.7 广告信息文件 (AdvertisementInfo)	16
5.4.8 标签文件 (Label)	16
5.4.9 操作记录 (SaleRecord)	17
5.4.10 溯源文件 (Trace)	18
5.4.11 信息条文 (Message1)	19

附录一、条形码编码对照表	20
--------------------	----

一、简介

本文档将介绍如何调用Aclas Scale SDK接口与顶尖条码标签秤进行数据交互。

支持的设备：顶尖LH51系列、LS M3系列条码标签秤及TS系列触控条码标签秤。

支持的操作系统：Windows及Linux

接口方式：动态连接库接口（DLL或SO）及命令行接口（Console）

文件列表：

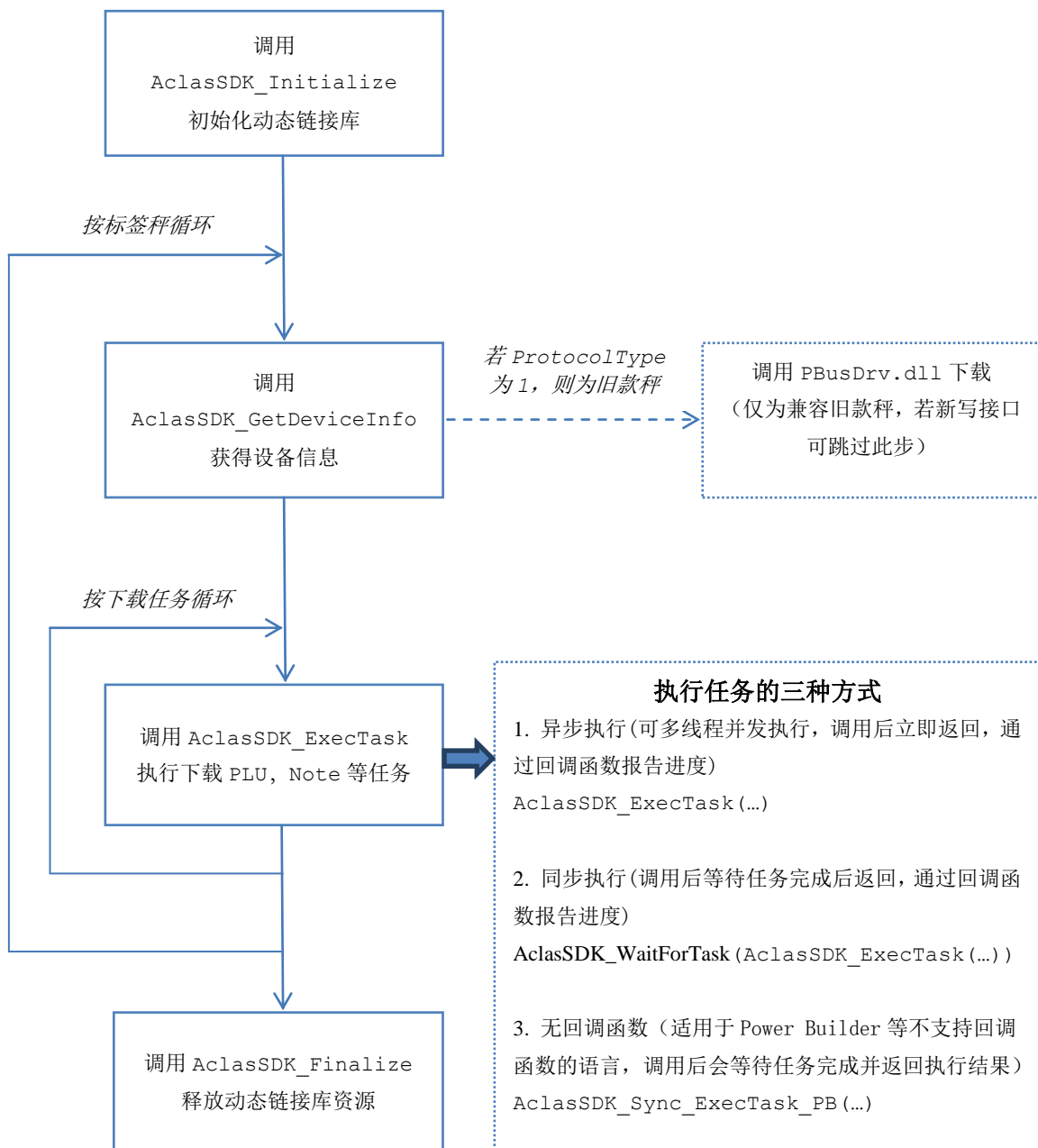
- | | |
|-----------------------|------------------|
| 1、AclasSDK.dll | ----Windows动态连接库 |
| 2、AclasSDKConsole.exe | ----Windows命令程序 |
| 3、AclasSDK.so | ----Linux动态连接库 |
| 3、AclasSDKConsole | ----Linux命令程序 |

命令程序依赖于动态连接库，如采用控制台接口方式调用，需同时附带动态连接库。而采用动态连接库调用时无需附带命令程序。

一般来说，我们通过SDK接口实现单品文件和备注文件的下载就可以了，对于触控秤还要实现部门文件和分组文件的下载。其它的数据多为不常用或一次性的配置数据，建议使用我司标签秤上位机软件Link65来完成。

二、动态连接库接口

2.1 动态连接库调用流程



2.2 动态连接库函数说明

2.2.1 初始化

Pascal	function AclasSDK_Initialize(Adjuct: Pointer): Boolean; stdcall;		
描述	用于初始化动态连接库。		
	参数	类型	描述
输入	Adjuct	Pointer	保留字, 传nil或null
返回		Boolean	True为成功, False失败。

2.2.2 释放

Pascal	procedure AclasSDK_Finalize; stdcall;		
描述	用于释放动态连接库分配的资源。		

2.2.3 获取设备协议类型

Pascal	function AclasSDK_GetDeviceType(Addr, Port, ProtocolType: UInt32): Integer;		
描述	用于获得设备协议类型。返回值可用来区分秤的种类, 若为1则为顶尖旧款标签秤, 请调用旧款标签秤动态链接库PBusDrv.dll		
	参数	类型	描述
输入	Addr	UInt32	设备IP地址 (整型)
	Port	UInt32	设备端口号, 填0
	ProtocolType	UInt32	协议类型, 填0
返回		Integer	0:未识别 其它返回值见4.1

说明: 整型IP地址: 就是ip地址的整数 (UInt32) 表达形式如" 192.168.2.208" 转成整数形式为:

192*256*256*256+168*256*256+2*256+208=3232236240

2.2.4 获取设备信息

Pascal	function AclasSDK_GetDeviceInfo(Addr, Port, ProtocolType: UInt32): TASSDKDeviceInfo; C#可采用下面的函数 function AclasSDK_GetDeviceInfo(Addr, Port, ProtocolType: UInt32; var DeviceInfo TASSDKDeviceInfo):Boolean;		
描述	用于获得设备信息。		
	参数	类型	描述
输入	Addr	UInt32	设备IP地址 (整型)
	Port	UInt32	设备端口号, 填0
	ProtocolType	UInt32	协议类型, 详见4.1
返回		TASSDKDeviceInfo	详见4.5

2.2.5 执行任务

Pascal	function AclasSDK_ExecTask(Addr, Port, ProtocolType, ProcType, DataType: UInt32; FileName: PWideChar; OnProgress: TASSDKOnProgressEvent; UserData: Pointer): THandle; stdcall;		
--------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

	<pre>// Ansi版本 function AclasSDK_ExecTaskA(Addr, Port, ProtocolType, ProcType, DataType: UInt32; FileName: PAnsiChar; OnProgress: TASSDKOnProgressEvent; UserData: Pointer): THandle; stdcall;</pre>		
描述	<p>用来执行下载上传数据。这个函数是异步执行的，调用后会立即返回，具体的进度是通过回调函数报告的。</p> <ol style="list-style-type: none"> 1. 若希望取消任务，请调用AclasSDK_StopTask。 2. 若希望同步执行，请使用AclasSDK_WaitForTask(AclasSDK_ExecTask(...))的形式调用。 3. 若使用的语言不支持回调函数(如PowerBuilder)，请使用AclasSDK_Sync_ExecTask_PB。 		
	参数	类型	描述
输入	Addr	UInt32	设备IP地址（整型）
	Port	UInt32	设备端口号，填0
	ProtocolType	UInt32	协议类型，详见4.1
	ProcType	UInt32	操作类型，详见4.2
	DataType	UInt32	数据类型，详见4.3
	FileName	string	上传、修改、下载的文件绝对地址 AclasSDK_ExecTask为WideString AclasSDK_ExecTaskA为AnsiString
	OnProgress	TASSDKOnProgressEvent	进度回调函数
	UserData	Pointer	用户自定义数据指针，SDK不对此值进行任何操作，回调时原封不动返回。 此参数的主要目的是在多线程异步执行多个任务时，能在回调函数中区分任务。
返回		THandle	任务句柄，返回为-1时，任务执行失败。

2.2.6 停止任务

Pascal	procedure AclasSDK_StopTask(TaskHandle: THandle = 0); stdcall;		
描述	用于取消一个任务		
	参数	类型	描述
输入	TaskHandle	THandle	任务句柄，为0时停止所有任务。

2.2.7 等待任务完成

Pascal	procedure AclasSDK_WaitForTask(TaskHandle: THandle); stdcall;		
描述	用于等待任务完成后再返回。即同步执行任务。		
	参数	类型	描述
输入	TaskHandle	THandle	任务句柄

2.2.8 进度回调函数

Pascal	TASSDKOnProgressEvent = procedure(nErrorCode, Index, Total: UInt32; lpUserData: Pointer); stdcall;		
	参数	类型	描述
输入	nErrorCode	UInt32	错误代码，详见4.4

	Index	UInt32	当前进度
	Total	UInt32	总数

2.2.9 执行任务(适用于Power Builder等无回调函数的编程语言，其他语言可参考使用)

Pascal	<pre>function AclSDK_Sync_ExecTask_PB(Addr: PWideChar; Port, ProtocolType, ProcType, DataType: UInt32; FileName: PWideChar): Integer; stdcall; // Ansi版本 function AclSDK_Sync_ExecTaskA_PB(Addr: PAnsiChar; Port, ProtocolType, ProcType, DataType: UInt32; FileName: PAnsiChar): Integer; stdcall; //Power Builder请用Ansi版本，下面是Power Builder函数声明 function int AclSDK_Sync_ExecTaskA_PB(ref string Addr, uint Port, uint ProtocolType, uint ProcType, uint DataType, ref string FileName)</pre>		
	参数	类型	描述
输入	Addr	string	设备IP地址如: 192.168.0.2
	Port	UInt32	设备端口号, 填0
	ProtocolType	UInt32	协议类型, 详见4.1
	ProcType	UInt32	操作类型, 详见4.2
	DataType	UInt32	数据类型, 详见4.3
	FileName	string	上传、修改、下载的文件绝对地址 AclSDK_ExecTask为WideString AclSDK_ExecTaskW为WideString AclSDK_ExecTaskA为AnsiString
返回		Integer	0: 执行成功。其它错误代码, 详见4.4

三、命令行接口

采用命令行调用方式，调用 AclasSDKConsole.exe 并带入参数执行操作

3.1 命令行参数说明

参数	作用	值	功能
-h	设备 IP 地址	如：“192.168.2.208”	指定设备的 IP 地址
-p	协议类型	默认为 None, 即自动侦测协议。详见 4.1	
-t	操作类型	详见 4.2	
-b	数据类型	详见 4.3	
-d	数据序号	主要用于传递标签序号，默认为 0。详见 5.4.8	
-n	文件名	如：“d:\plu.txt”	文件名

返回值 (PostQuitMessage):

- 0: 执行成功
- 1: 设备离线
- 2: 参数错误
- 3: 缺少参数
- 4: 执行错误

3.2 命令行使用范例

功能：下载 PLU 数据 d:\PLU.txt 到 192.168.2.208 的设备上

命令：AclasSDKConsole.exe -h "192.168.2.208" -t Down -b PLU -n "d:\PLU.txt"

功能：从 192.168.2.208 的设备上传 PLU 数据到 d:\PLU.txt

命令：AclasSDKConsole.exe -h "192.168.2.208" -t Up -b PLU -n "d:\PLU.txt"

功能：清除 192.168.2.208 的设备上的 PLU 数据

命令：AclasSDKConsole.exe -h "192.168.2.208" -t Del -b PLU -n "*"

功能：下载 Note1 数据 d:\Note1.txt 到 192.168.2.208 的设备上

命令：AclasSDKConsole.exe -h "192.168.2.208" -t Down -b Note1 -n "d:\Note1.txt"

功能：下载热键数据 d:\hotkey.txt 到 192.168.2.208 的设备上

命令：AclasSDKConsole.exe -h "192.168.2.208" -t Down -b Hotkey -n "d:\hotkey.txt"

功能：下载热键图片 d:\900001.jpg 到 192.168.2.208 的触控秤上

命令：AclasSDKConsole.exe -h "192.168.2.208" -t Down -b KeyImage -n "d:\900001.jpg"

功能：同步 192.168.2.208 中的设备时间

命令：AclasSDKConsole.exe -h "192.168.2.208" -t Down -n "2008-08-08 08:08:08" -b Time

功能：下载标签格式文件 d:\label.lf 到 192.168.2.208 的设备上的第 1 号标签。

命令：AclasSDKConsole.exe -h "192.168.2.208" -t Down -b LabelFormat -d 0 -n "d:\label.lf"

功能：下载标签底纹文件 d:\label.lm 到 192.168.2.208 的设备上的第 1 号标签。

命令：AclasSDKConsole.exe -h "192.168.2.208" -t Down -b LabelMap -d 0 -n "d:\label.lm"

功能：下载标签设计文件 d:\label.tbl 到 192.168.2.208 的设备上的第 1 号标签。

命令：AclasSDKConsole.exe -h "192.168.2.208" -t Down -b LabelDesign -d 0 -n "d:\label.tbl"

功能：从 192.168.2.208 的设备上传第 1 号标签的标签设计文件到 d:\label.tbl。

命令：AclasSDKConsole.exe -h "192.168.2.208" -t Up -b LabelDesign -d 0 -n "d:\label.tbl"

四、类型定义

4.1 协议类型（ProtocolType）

ProtocolType(-p)		
DLL值	控制台命令	描述
0	None	自动检测
1	Pecr	Pecr协议
2	Hecr	Hecr协议
3	TSecr	TSecr协议
Pecr协议用于LH51或M3的LH51模式 Hecr协议用于M3系列 TSecr协议用于TS系列 设置为None，SDK会自动检测机型协议，而预设置协议参数可省去传输前的检测时间		

4.2 操作类型（ProcType）

ProcType(-t)		
DLL值	控制台命令	描述
0	Down	下载
1	UP	上传
2	Edit	修改
3	Del	删除
4	List	TS机型获取图片、视频、文件列表
设置任务的操作类型		

4.3 数据类型（DataType）

DataType(-b)			
DLL值（10进制）	DLL值（16进制）	控制台命令	描述
0	0x0000	PLU	单品资料
12	0x000C	Note1	备注1
13	0x000D	Note2	备注2
14	0x000E	Note3	备注3
28	0x001C	Note4	备注4
2	0x0002	Department	部门

4	0x0004	Group	分组
3	0x0003	HotKey	热键
5	0x0005	Discount	折扣排程
21	0x0015	AdvertisementInfo	广告信息
32	0x0020	LabelFormat	标签格式文件
33	0x0021	LabelMap	标签底纹文件
34	0x0022	LabelDesign	标签设计文件
261	0x0105	Message1	信息条文
263	0x0107	SaleRecord	操作记录
8193	0x2001	Time	时间
以下为LS独有			
288	0x0120	TraceStatus	溯源配置文件
289	0x0121	TraceMap	溯源索引文件
290	0x0122	TraceData	溯源数据文件
以下为TS独有			
4101	0x1005	KeyImage	热键图片
4357	0x1105	BatchKeyImage	批量热键图片（zip压缩包）

4.4 错误码（ErrorCode）

ErrorCode		
DLL值（10进制）	DLL值（16进制）	描述
0	0	正常
1	0x0001	进度事件
2	0x0002	手动停止
256	0x0100	已初始化
257	0x0101	未初始化
258	0x0102	设备不存在
259	0x0103	不支持的协议类型
260	0x0104	该数据类型不支持此操作
261	0x0105	该数据类型不支持
264	0x0108	无法打开输入文件
265	0x0109	字段数与内容数不匹配
266	0x010A	通讯数据异常
267	0x010B	解析数据异常
268	0x010C	CodePage错误
269	0x010D	无法创建输出文件

4.5 设备信息结构定义（TASSDKDeviceInfo）

TASSDKDeviceInfo

参数	类型	大小	
ProtocolType	UInt32	4	协议类型
Addr	UInt32	4	地址
Port	UInt32	4	端口
Name	AnsiChar	16	设备名称
ID	UInt32	4	设备ID
Version	UInt32	4	设备软件版本
Country	UInt8	1	国家类别
DepartmentID	UInt8	1	部门号
KeyType	UInt8	1	键盘类型
PrinterDot	UInt64	8	打印头已打印点数
PrnStartDate	TDateTime	8	打印头启用时间
LabelPage	UInt32	4	打印头已打印标签张数
PrinterNo	UInt32	4	打印头序列号
PLUStorage	UInt16	2	PLU可存储数量
HotKeyCount	UInt16	2	支持热键数量
NutritionStorage	UInt16	2	营养信息可存储数量
DiscountStorage	UInt16	2	打折排程可存储数量
Note1Storage	UInt16	2	Note1可存储数量
Note2Storage	UInt16	2	Note2可存储数量
Note3Storage	UInt16	2	Note3可存储数量
Note4Storage	UInt16	2	Note4可存储数量
Adjunct	AnsiChar	177	保留字段
整个结构大小为256			

五、数据文件格式

5.1 文件格式说明

- 第一行为字段名称，从第二行开始为正式的数据。字段顺序及字段个数无强制要求，但数据需与字段一一对应。
- 每个字段或者数据均采用 TAB (0x09) 隔开。每行之间用回车换行 (0x0D 0x0A) 隔开
- 数据中出现上述分隔符请按以下规则转义。

原字符 替换为

0x09 <->{\$09}

0x0A <->{\$0A}

0x0D <->{\$0D}

如原字符串中存在 {\$09}、{\$0A}、{\$0D} 等，请自行转义处理。

5.2 文件范例

以下为单品文件(PLU)的范例。

ID	DepartmentID	GroupID	Name1	Price	BarcodeType1
90001	20 1	Apple	9.99	40	
90002	20 1	Pear	8.88	40	
90003	20 2	Banana	3.00	40	

5.3 文件编码格式

输入文件: 根据文本文件编码头进行判断, 如为Ansi格式, CodePage以当前操作系统默认CodePage进行解析, 推荐采用Unicode格式。

输出文件: 默认Ansi格式, 可通过在文件名后加上 “?编码格式” 的方式指定。

Ansi: 0

Unicode: 1200

BigEndianUnicode: 1201

UTF8: 65001

如: plu.txt?65001 会创建文件名为plu.txt编码格式为UTF8的文本文件。

5.4 文件格式详解

加粗为主键。

5.4.1 单品文件 (PLU)

带下划线的字段为基本字段，请在单品文件中包含这些字段。

PLU																								
字段名	类型	范围	解释																					
ID	Int	999999	生鲜码, 唯一识别 PLU																					
ItemCode	Str	16	货号, 主要用于构造条码之用, 一般与生鲜码相同。请参见附录一、条形码编码对照表																					
DepartmentID	Int	0..99	部门号。此处的部门号主要用于构造条码之用。请参见附录一、条形码编码对照表 若一位部门建议用 2, 二位部门建议用 20 或 22																					
GroupID	Int	0..9999	分组号。此字段对触控秤是必须的, 关联分组 (Group) 表中的 ID。对普通按键式标签秤可不包含此字段。																					
Name1	Str	40	名称 1																					
Name2	Str	40	名称 2																					
Name3	Str	40	名称 3																					
Price	Float		单价, 最多小数点后两位, 最大为 999999.99																					
UnitID	Int		如不了解此部分, 请使用建议值: 称重用4(即kg), 计数用10(PCS) 称重单位编号, 值的意义为 0-50g;1-g;2-10g;3-100g;4-kg;5-oz;6-lb;7-500g; 8-600g;9-pcs(g);10-pcs(kg);11-pcs(oz);12-pcs(lb)																					
BarcodeType1	Int	0..255	条码类型 1。0-149: 内置; 150-255: 自定义。请参见附录一、条形码编码对照表 如不了解此部分, 请参照以下建议:																					
			<table><tr><td>货号位数</td><td>条码类型</td><td>建议值</td></tr><tr><td>5</td><td>18 码</td><td>94</td></tr><tr><td>5</td><td>13 码带总价</td><td>2</td></tr><tr><td>5</td><td>13 码带重量</td><td>7</td></tr><tr><td>6</td><td>18 码</td><td>30</td></tr><tr><td>6</td><td>13 码带总价</td><td>22</td></tr><tr><td>6</td><td>13 码带重量</td><td>27</td></tr></table>	货号位数	条码类型	建议值	5	18 码	94	5	13 码带总价	2	5	13 码带重量	7	6	18 码	30	6	13 码带总价	22	6	13 码带重量	27
			货号位数	条码类型	建议值																			
			5	18 码	94																			
			5	13 码带总价	2																			
			5	13 码带重量	7																			
			6	18 码	30																			
6	13 码带总价	22																						
6	13 码带重量	27																						
BarcodeType2	Int	0..255	条码类型 2。0-149: 内置; 150-255: 自定义																					
Label1ID	Int	0..32	标签 1 编号。0: 不打印																					
Label2ID	Int	0..32	标签 2 编号。0: 不打印																					
ProducedDate	Str		生产日期。根据系统时间格式, 如: YYYY/MM/DD hh:mm:ss																					
FreshnessDate	Int	0..999	保鲜天(小时)数。保鲜期计算方式详见[FreshnessDateFrom]定义																					
ValidDate	Int	0..999	保质期。保质期计算方式详见[ValidDateFrom]定义																					
PackageType	Int	0..4	包装类型。0:正常; 1:定重; 2:定价; 3:定重定价; 4:拼盘																					
PackageWeight	Float		包装重量, 或者限制销售重量																					
PackagePrice	Float		包装价格。																					
PackageRange	Int	0..99	包装误差。0-99%																					

PackageDays	Int	0..999	包装天数。包装日期计算方式详见[PackageDateFrom]
PackageHours	Int	0..99	包装小时数。包装日期计算方式详见[PackageDateFrom]
DiscountID	Int	0..99	折扣表 ID。0: 无折扣; 1-99: 关联 Discount 表中的 ID DiscountID 与 DiscountRate 二选一
DiscountRate	Float	0..100	折扣; 90.05%—>9005
TareID	Int	0..99	皮重。0: 无效; 1-99 关联 Tare 表中的 ID。 TareID 与 TareValue 二选一
TareValue	Float		皮重值。
LimitPrice	Float		最高单价(如果为 0 也不做最高限价)
Flag1	Byte		如不了解此部分, 请使用建议值60 (16进制:0x3C, 二进制:00111100) Bit0:是否允许变价。 0: 允许; 1: 不允许 Bit1: 保鲜, 保质, 销售天数或小时数。 0: 按天; 1: 按小时 Bit2: 是否打印包装日期。0: 不打印; 1: 打印 Bit3: 是否打印保鲜日期。0: 不打印; 1: 打印 Bit4: 是否打印保质期。 0: 不打印; 1: 打印 Bit5: 是否打印Note3。 0: 不打印; 1: 打印 Bit6: N/A, 默认为0 Bit7: N/A, 默认为0
Flag2	Byte		如不了解此部分, 请使用建议值240 (16进制:0xF0, 二进制: 11110000) Bit0: 是否打印单品文本1。 0: 不打印; 1: 打印 Bit1: 是否打印单品文本2。 0: 不打印; 1: 打印 Bit2: 是否打印单品文本3。 0: 不打印; 1: 打印 Bit3: 是否打印单品文本4。 0: 不打印; 1: 打印 Bit4: 是否打印营养信息。 0: 不打印; 1: 打印 Bit5: 是否打印Note1。 0: 不打印; 1: 打印 Bit6: 是否打印Note2。 0: 不打印; 1: 打印 Bit7: 是否打印生产日期。 0: 不打印; 1: 打印
Flag3	Byte		Bit0: 溯源商品。 0: 否; 1: 是 Bit1: 追溯信息顺序。 0: 采用先进先出; 1: 采用后进先出 Bit2: 是否最高限价。 0: 不限制; 1: 不许超过最高单价 Bit3: N/A, 默认为0 Bit4: 是否打印单品文本5。 0: 不打印; 1: 打印 Bit5: 是否打印单品文本6。 0: 不打印; 1: 打印 Bit6: 是否打印单品文本7。 0: 不打印; 1: 打印 Bit7: 是否打印单品文本8。 0: 不打印; 1: 打印
ProducedDateRule	Int		生产日期定义规则 0:生产日期 = 系统日期 1:生产日期 = [ProducedDate]
FreshnessDateFrom	Int		保鲜期的基准时间 0:保鲜期 =系统日期 + [FreshnessDate] 1:保鲜期 = 生产日期+ [FreshnessDate] 2:保鲜期 = 包装日期+ [FreshnessDate]

ValidDateFrom	Int		保质期的基准时间 0:保质期 = 系统日期 + [ValidDate] 1:保质期 = 生产日期 + [ValidDate] 2:保质期 = 包装日期 + [ValidDate]
PackageDateFrom	Int		包装日期的基准时间 0:包装日期 = 系统日期 + [PackageDays] + [PackageHours] 1:包装日期 = 生产日期 + [PackageDays] + [PackageHours]
Message1	Int		引用的信息条文号

Flag 字段的说明

Flag 字段一般是一组开关标志。我们知道一个 byte(字节)有 8 个 bit(位),我们称之为 bit0~bit7, 每一个 bit 可对应一个开关标志, 故一个 flag 字段最多可包含 8 个标志。Flag 字段的值可通过下面的方法计算:

Flag 字段的值

$$= \text{bit0} * 2^0 + \text{bit1} * 2^1 + \text{bit2} * 2^2 + \text{bit3} * 2^3 + \text{bit4} * 2^4 + \text{bit5} * 2^5 + \text{bit6} * 2^6 + \text{bit7} * 2^7$$

其中每个 bit 只有 0 和 1 两个值

^代表乘方

5.4.2 备注文件 (Note1/2/3/4)

- 每条 PLU 可以有 4 条备注, 每条备注最多可容纳 1000 字符。
- 备注通过单独的命令下载, 并用 PLUID 与 plu 的生鲜码关联

Note1			
字段名	类型	大小	解释
PLUID	Int		生鲜码
Value	Str	1000	材料信息

Note2			
字段名	类型	大小	解释
PLUID	Int		生鲜码
Value	Str	1000	附加信息

Note3			
字段名	类型	大小	解释
PLUID	Int		生鲜码
Value	Str	1000	附加信息

Note4			
字段名	类型	大小	解释
PLUID	Int		生鲜码
Value	Str	1000	附加信息

5.4.3 部门文件 (Department)

部门的概念对应于 Mis 后台的部门/类别组/大类别的概念。

Department			
字段名	类型	大小	解释
ID	Int		部门 ID
Name	Str	30	名称

PLU 文件中有个部门栏位，仅用于构成条码之用，与此处的部门没有关联关系。

5.4.4 分组文件 (Group)

分组的概念对应于 Mis 后台的类别/小类别的概念，它的上级是部门。

通过 DepartmentID 与部门关联起来，即每个分组属于某个部门。

Group			
字段名	类型	大小	解释
ID	Int		组 ID
Name	Str	30	名称
DepartmentID	Int		所属部门 ID
ShowPosition	Int		分组显示位置

PLU 中有一栏 GroupID 将 PLU 与分组关联起来，即每条 PLU 属于某个分组。

5.4.5 热键文件 (HotKey)

热键文件有两种格式：

标签秤(*.key)：

HotKey			
字段名	类型	大小	解释
ButtonIndex	Int		热键号，从 1 开始，总数为热键数（如 LS2 为 224）
ButtonValue	Int		热键值，填入 PLU 的生鲜码

触控秤(*.tsk)：

HotKey			
字段名	类型	大小	解释
PLUID	Int		生鲜码
GroupID	Int		分组号
IsShowTouch1	Int		是否显示
Position1	Int		显示位置
IsShowTouch2	Int		保留
Position2	Int		保留

5.4.6 折扣排程文件 (Discount)

Discount			
字段名	类型	大小	解释
ID	Int		折扣排程号
Name	Str	50	折扣排程名称，触控秤会用到这个字段
ItemIndex	int	0..2	折扣排程子项序号
Mode	Int	1..99	折扣模式
DiscountRate	Float		折扣率
DiscountAmt	Float		折扣值
BeginDateTime	Datetime		开始时间
EndDateTime	Datetime		结束时间
MinWeight	Float		最小重量
MaxWeight	Float		最大重量
MinTotalPrice	Float		最小总价
MaxTotalPrice	Float		最大总价
Flag1	Byte		Bit0: N/A, 默认为0 Bit1: N/A, 默认为0 Bit2: N/A, 默认为0 Bit3: 时间范围模式。0:每日该时间段内; 1:开始时间到结束时间 Bit4: 是否启用总价范围。0:不启用; 1:启用 Bit5: 是否启用重量范围。 0:不启用; 1:启用 Bit6: 是否启用时间范围。 0:不启用; 1:启用 Bit7: 是否启用排程子项。 0:不启用; 1:启用

5.4.7 广告信息文件 (AdvertisementInfo)

用于在屏幕上显示广告信息，只有一行数据。

AdvertisementInfo			
字段名	类型	大小	解释
Value	Str	256	广告信息

5.4.8 标签文件 (Label)

标签文件是二进制文件，一套标签文件分为3个文件，分别是

标签底纹文件(LabelMap), 文件类型为*.LM

标签格式文件(LabelFormat), 文件类型为*.LF

标签设计文件(LabelDesign), 文件类型为*.TBL

在标签设计器中保存为*.TBL 时，会自动保存同名的*.LM 和*.LF 文件。

标签设计器中也可以保存为*.TBZ 格式，这其实是将上述三个文件打包成一个 ZIP 文件。

下载到标签秤时需同时下载这三个文件，上传时只要上传标签设计文件(LabelDesign)即可。

顶尖条码标签秤支持多张标签，用标签号区分。一般只需使用第 1 号标签即可满足需要。

请注意：下载标签的接口中使用的标签序号=标签号-1，即第 1 号标签的标签序号为 0，第 2 号标签的标签序号为 1，依此类推。

如何传递标签序号呢？

➤ 动态链接库接口

标签序号是通过操作类型(ProcType)的高位字节来表示的。

如下载第 8 号标签的 ProcType 的值为：

ProcType=7*256+0；

如上传第 5 号标签的 ProcType 的值为：

ProcType=4*256+1；

➤ 命令行接口

通过-d 参数指定标签序号。标签序号为 0 是可省略此参数。

请参见 3.2 命令行使用范例

5.4.9 操作记录 (SaleRecord)

SaleRecord			
字段名	类型	大小	解释
TypeID	Str	1	当字段值为'D'时表示此段数据为标签销售单品； 当字段值为'E'时表示此段数据为收据销售的单品；
IP	Int		秤的 IP 地址
ReceiptNO	Int		当字段值为'E'时此值为收据的流水号
PLUID	Int		单品的生鲜码
UnitPrice	Int		单品销售单价
TotalPrice	Int		单品销售总价
DiscountAmt	Int		单品优惠金额
UnitID	Int		称重单位编号,值的意义为 0-50g; 1-g; 2-10g; 3-100g; 4-kg; 5-oz; 6-lb; 7-500g; 8-600g; 9-pcs(g); 10-pcs(kg); 11-pcs(oz); 12-pcs(lb)
Weight	Int		当 pcs 时此值件数如 1 件值为 1000; 当非 pcs 时此值为实际的称重重量
SaleTime	Datetime		销售日期时间
OnlineTime	Datetime		最后一次清销售数据的日期时间
Clerk	Str	8	收银员号
Tracecode	Str	33	追溯码
Pcode	Int		追溯码批次码
LinkF	Int		0x10:联网变价, 0x11:联网不变价 0x01:不联网变价 0x00:不联网不变价

5.4.10 溯源文件 (Trace)

溯源文件包含三个相互关联的文件，分别是

溯源配置文件(TraceStatus)，建议文件名为 TraceStatus.txt

溯源索引文件(TraceMap)，建议文件名为 TraceMap.txt

溯源数据文件(TraceData)，建议文件名为 TraceData.txt

- 溯源数据是有时效性的，故每次下载均为完整覆盖下载。
- 生鲜码和溯源码之间是多对多关系。即一个生鲜码可能有多个溯源码，一个溯源码可能被多个生鲜码共用。
- 溯源配置文件(TraceStatus)只有一行数据。
- 对于简单的溯源码应用场景，如一个生鲜码只有一个溯源码的情形，也可以直接将溯源码放在备注(Notel/2/3/4)中的方式来实现。

TraceStatus			
字段名	类型	大小	解释
MapCount	Int		对应 TraceMap.txt 里面的数据笔数
DataCount	Int		对应 TraceData.txt 里面的数据笔数
Http	Str	128	溯源码网址

范例

MapCount DataCount Http

2 2 http://www.xxxtrace.com/search.html?code=

TraceMap			
字段名	类型	大小	解释
PLUID	Int		生鲜码
Index	Int		索引号(即此生鲜码对应的溯源码在 TraceData.txt 文档中行号，行号是从 0 开始计数的)

范例

PluID Index

601 0

602 1

TraceData			
字段名	类型	大小	解释
GroupID	Int		保留，填 0 即可
TraceCode	Str	30	溯源码

范例

GroupID TraceCode

0 41110321000000013407

0 41110321000000013507

5.4.11 信息条文 (Message1)

- 与备注不同，信息条文是单独维护的，即不是和 PLU 一一对应的
- PLU 中有个 Message1 字段，填的就是这里的信息条文号 (ID)，表示该 PLU 引用这条信息条文

Message1			
字段名	类型	大小	解释
ID	Int		信息条文号
Val	Str	245	信息内容

附录一、条形码编码对照表

条形码类型	部门	货号	总价	重量	校验和
00~09: Ean13 码, 前两码印部门码:					
00	DD(2)	IIIIIIII(10)	X	X	C
01	DD(2)	IIIII(6)	PPPP(4)	X	C
02	DD(2)	IIII(5)	PPPPP(5)	X	C
03	DD(2)	IIII(4)	PPPPPP(6)	X	C
04	DD(2)	III(3)	PPPPPPP(7)	X	C
05	DD(2)	IIIII(6)	X	W.WWW(4)	C
06	DD(2)	IIIII(6)	X	WW.WW(4)	C
07	DD(2)	IIII(5)	X	WW.WWW(5)	C
08	DD(2)	IIII(5)	X	WWWW.W(5)	C
09	DD(2)	IIII(5)	X	WWWWW(5)	C
10~19: Ean13 码, 前两码印固定码:					
10	20(2)	IIIIIIII(10)	X	X	C
11	21(2)	IIIII(6)	PPPP(4)	X	C
12	22(2)	IIII(5)	PPPPP(5)	X	C
13	23(2)	IIII(4)	PPPPPP(6)	X	C
14	24(2)	III(3)	PPPPPPP(7)	X	C
15	25(2)	IIIII(6)	X	W.WWW(4)	C
16	26(2)	IIIII(6)	X	WW.WW(4)	C
17	27(2)	IIII(5)	X	WW.WWW(5)	C
18	28(2)	IIII(5)	X	WWWW.W(5)	C
19	29(2)	IIII(5)	X	WWWWW(5)	C
20: 不打印条形码					
21~29: Ean13 码, 前一码印部门码:					
21	D(1)	IIIII(7)	PPPP(4)	X	C
22	D(1)	IIIII(6)	PPPPP(5)	X	C
23	D(1)	IIII(5)	PPPPPP(6)	X	C
24	D(1)	IIII(4)	PPPPPPP(7)	X	C
25	D(1)	IIIII(7)	X	W.WWW(4)	C
26	D(1)	IIIII(7)	X	WW.WW(4)	C
27	D(1)	IIIII(6)	X	WW.WWW(5)	C
28	D(1)	IIIII(6)	X	WWWW.W(5)	C
29	D(1)	IIIII(6)	X	WWWWW(5)	C
30~35: 18 码, 前一码印部门码					
30&33	D(1)	IIIII(6)	PPPPP(5)	WW.WWW(5)	C
31&34	D(1)	IIIII(6)	PPPPP(5)	WWWW.W(5)	C
32&35	D(1)	IIIII(6)	PPPPP(5)	WWWWW(5)	C

条形码类型	部门	货号	单价	重量	校验和
40~45: 18 码, 前一码印部门码					
40&43	D(1)	IIIII(6)	UUUUU(5)	WW. WWW(5)	C
41&44	D(1)	IIIII(6)	UUUUU (5)	WWWW.W(5)	C
42&45	D(1)	IIIII(6)	UUUUU (5)	WWWWW(5)	C

条形码类型	部门	货号	总价	重量	校验和
46: 18 码					
46	DD(2)	IIIII(6)	PPPPP(5)	WWWWW(5)	X
50~55: 8 码					
50	X	IIIII(7)	X	X	C
51	D(1)	IIIII(6)	X	X	C
52	DD(2)	IIII(5)	X	X	C
53	X	IIIIII(8)	X	X	X
54	D(1)	IIIII(7)	X	X	X
55	DD(2)	IIIII(6)	X	X	X

条形码类型	部门	生鲜码	批次号	折扣	重量
36-38: 18 码, 前一码印部门码, 为生鲜批次管理专用					
36	D(1)	LLLLL(6)	BBBB(4)	RR(2)	WW.WWW(5)
37	D(1)	LLLLL(6)	BBBB(4)	RR(2)	WWWW.W(5)
38	D(1)	LLLLL(6)	BBBB(4)	RR(2)	WWWWW(5)

条形码类型	部门	生鲜码	批次号	折扣	重量	校验和
66~68: 18 码, 为生鲜批次管理专用						
66	D(1)	LLLLL(5)	BBBB(4)	RR(2)	WW.WWW(5)	C
67	D(1)	LLLLL(5)	BBBB(4)	RR(2)	WWWW.W(5)	C
68	D(1)	LLLLL(5)	BBBB(4)	RR(2)	WWWWW(5)	C

条形码类型	部门	货号	总价或单价	重量	校验和
60~65: ISBN 码, 18 码的一种					
60	D(1)	IIIII(6)	PPPPP(5)	WW.WWW(5)	C
61	D(1)	IIIII(6)	PPPPP(5)	WWWW.W(5)	C
62	D(1)	IIIII(6)	PPPPP(5)	WWWWW(5)	C
63	D(1)	IIIII(6)	UUU.UU(5)	WW.WWW(5)	C
64	D(1)	IIIII(6)	UUU.UU(5)	WWWW.W(5)	C
65	D(1)	IIIII(6)	UUU.UU(5)	WWWWW(5)	C

标签类型	部门	货号	数(重)量	总价或单价	校验和
79~85, 87: 18 码					


79	DD	IIIII(6)	WW.WWW	PPPPP	X
80	D	IIIII(6)	WW.WWW	PPPPP	C
81	D	IIIII(6)	WWWW.W	PPPPP	C
82	D	IIIII(6)	WWWWW	PPPPP	C
83	D	IIIII(6)	WW.WWW	UUU.UU	C
84	D	IIIII(6)	WWWW.W	UUU.UU	C
85	D	IIIII(6)	WWWWW	UUU.UU	C
87	DD	IIII(5)	WW.WWW	PPPPP	C


条形码类型	部门	货号	总价	数(重)量	单价	校验和
90~95: 18 码, 前两码印部门码:						
90	DD(2)	IIIII(6)		WW.WWW	UU.UU(4)	C
91	DD(2)	IIIII(6)		WWWW.W	UU.UU(4)	C
92	DD(2)	IIIII(6)		WWWWW	UU.UU(4)	C
93	D(1)	IIIII(6)	PPPPP(5)		UUU.UU(5)	C
94	DD(2)	IIII(5)	PPPPP(5)	WW.WWW		C
95	DD(2)	IIIII(6)		WWWWW	UU.UU(4)	C
条形码类型	部门	货号	数(重)量		总价	单价
96	DD(2)	IIIII(6)	W.WWW	PPPPP(5)		C
条形码类型	部门	货号	总价	数(重)量	单价	校验和
97	DD(2)	IIII(4)	PPPPP(6)	WW.WWW		C
条形码类型	部门	货号	单价	数(重)量	总价	校验和
98	DD(2)	IIII(5)	UUUUU(5)	WW.WWW		C


表格中字母含义如下:


C: 校验和(CHECKSUM) D: 部门码 2: 固定数字“2” I: 货号 L: 生鲜码
P: 总价 U: 单价 R: 折扣 W: 重量 X: 无此项


 Ean13 码= DEPARTMENT+ CODE+ [TOTAL PRICE]+ [WEIGHT]+C 其中: 带[] 项表示在某些编码方式中无此项。


 对 Ean13 码来说, 若采用价格条形码, 则条形码类型常选用 2(或 22), 若采用重量条形码, 则条形码类型常选用 7(或 27)。

 CHECKSUM 由标签秤自动计算, 用户在 PLU manager 中无需输入此项。


 30-32 与 33-35 格式的区别在于 CHECKSUM 的计算方法不同。

 40-45 与 30-35 格式的区别在于 30-35 中价格为总价, 40-45 中价格为单价。

 36-38, 66~68 为生鲜批次管理可采用的条形码类型, 商品批次号填入货号字段, 商品货号填入生鲜码字段。

 60-65 为 ISBN 码。

 EAN13 码校验码 Z 的计算方法: (偶位数之和*3) + 奇位数之和 + Z=10 的倍数

 18 码校验码 Z 的计算方法二种:

方法一、与 13 码相同 (条码类型: 30-32, 40-42)

方法二、(奇位数之和*3) + 偶位数之和 + Z=10 的倍数
(条码类型: 33-35, 43-45)