机场共用自助旅客处理平台技术规范

Technical Specification of Civil Airport Common Use Self Service Passenger Processing Platform

目 次

前　言

本标准按照GB/T1.1-2009规则起草。

本标准版权归中国民用机场协会所有。

本标准起草单位：中国民航信息股份有限公司、中国民航大学。

本标准起草人员：

机场共用自助旅客处理平台技术规范

1 范围

本标准规定了机场共用自助服务的要求。本标准允许多个航空公司共享一个物理柜机，为乘客提供自助服务。这些服务包括,但不限于值机功能。该标准还允许航空公司开发符合CUSS的航司应用程序，这些航司应用程序能够在任何兼容CUSS平台的柜机上运行。

本标准适用于民用机场公共区域（包括军民合用机场内民航旅客使用的区域）。

2 规范性引用文件

下列文件对于本文件的航司应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

IATA推荐性惯例—2015 共用旅客自助服务规范版本1.4(Common Use Self Service (CUSS) Technical Specification, Revision 1.4)

AEA 技术规范—2002 2008 2009 ATB技术规范(ATB Technical specs- Amended August 2002, 2008, 2009)

AEA 技术规范—2002 2008 2009 参数化行李条数据概念(Parametric Baggage Tag Data Concept –2002, 2008, 2009)

AEA 技术规范—2001 自助服务规范(Self Service Specifications - 2001)

AEA 技术规范—2013 自助丢包(Self Bag Drop –2013)

3 术语和定义

下列术语和定义适用于本文件。

3.1

共用旅客自助规范　CUSS　Common Use Passenger Processing Systems

由国际航协定义，用于航空公司使用机场的终端设备的信息处理规范。

3.2

CUSS平台　CP　CUSSplatform

CUSS平台作为应用和硬件之间的中间件，负责应用的显示管理，应用对硬件的使用管理，远程应用和设备管理。CUSS平台提供一种区别于传统办理机场业务手续的方式，旅客通过此平台可获得更好的体验查询、订票、登机一站式的服务。

3.3

CUSS航司应用　CA　CUSSApplication

运行在CUSS平台上，使用CUPSS平台接口的航空公司航司应用程序，文中简称为航司应用。也可称为Kiosk Application, Self Service application, airline application。

3.4

通用航司应用启动界面　CLA　CommonLaunchApplication

CLA是指，柜机上的航司应用在未处于激活状态时，CUSS平台提供的航司应用选择界面。

3.5

柜机　Kiosk

包含硬件柜机，CUSS平台和CUSS航司应用。

3.6

进纸口　Feeder

打印机的组成部分之一。一个打印机，可以有多个进纸口，放置不同的登机牌。头等舱、公务舱和普通舱旅客，可以使用不同的登机牌。一般CUSS机上的打印机，只有一个进纸口。

3.7

出纸口　Dispenser

打印机的组成部分之一。

3.8

废纸槽　Capture

打印机的组成部分之一。若旅客打印登机牌后未取走登机牌，则登机牌被回收到废纸槽中。

3.9

纸箱　Bin

打印机的组成部分之一。进纸口、出纸口和废纸槽均属于纸箱。

3.10

登机牌打印机　ATB Printer　Automatic Ticketing and Boarding Printer

登机牌打印机。

3.11

通用打印机　GPP　General Purpose Printer

通用打印机，可用于打印行李条。

3.12

媒介　Media

一般指物理媒介的抽象。可以是各种护照、身份证、登机牌等等。护照阅读器，就是一个MediaInput，媒介就是护照。登机牌打印机是MediaOutput，媒介是登机牌。

3.13

CUSS航司应用管理　CAM　CUSSApplicaionManagement

CAM负责控制和调度所有运行在CUSS平台上的航司应用的状态。例如，CLA将用户选择的航司应用通知CAM时，CAM负责将航司应用设置为“激活”状态；夜间机场停运后，CAM可以将所有航司应用设置为“停用”或“挂起”状态。CAM负责将航司应用状态通知CLA。无论什么原因，若某航司应用状态变为“不可用”，那么CAM会通知CLA，CLA会将航司应用图标置灰或不显示，禁止旅客点选。

4 CUSS 总则

4.1 CUSS 简介

机场提供CUSS柜机，在柜机上安装CUSS平台，及航司应用，为提供旅客自助服务，例如选座，值机（Check-in）等，达到资源共享的目的。

常规情况下，CUSS柜机提供读设备（身份证阅读器，护照阅读器，扫描枪等）和打印设备（登机牌打印机，行李条打印机），并提供人机交互的触摸屏。

CUSS平台上可运行一个或多个航司应用，理论上数量没有限制。同一时间，只有一个航司应用处于激活状态，为旅客提供自助服务。其他航司应用处于“未激活”状态，在后台运行。

本标准规定CUSS平台和航司应用间的交互行为以及接口等技术内容。

典型CUSS架构如下图1所示：

4.2 CUSS 基本原则

CUSS平台提供商，可按需求自由选择操作系统。国内目前使用WinXP，Win7和Win10。

CUSS平台不应指定硬件。对柜机硬件的CPU架构没有要求；只要硬件符合CUSS标准的硬件接口要求，即应可用。

CUSS航司应用与平台相互独立，无耦合性。理论上，通过CUSS认证的航司应用可在任何机场提供的CUSS平台上运行。

4.3 CUSS 平台软件架构

4.3.1 架构

CUSS平台由以下部分组成(见图1)：

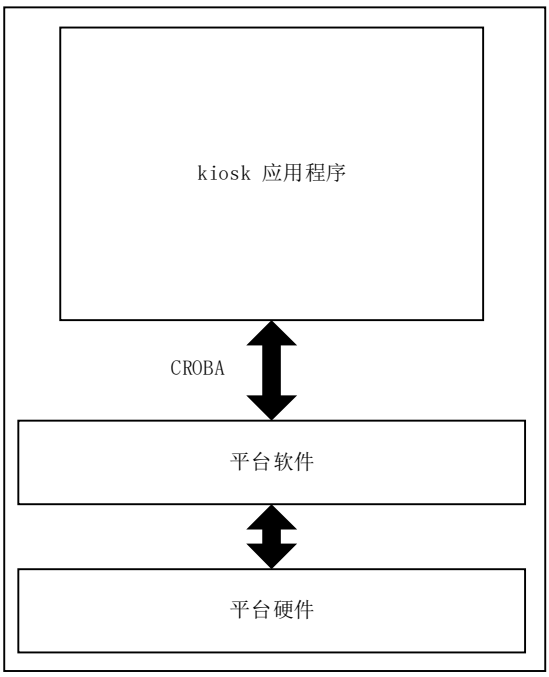

图 1 典型的 CUSS 架构图

4.3.2 CLA

航司应用航司应用航司应用航司应用通常，若航司应用状态变化，CLA状态相应变化。例如航司应用将自身设置成“不可用”状态，则CLA页面上该航司应用被置成灰色或不显示。若所有在CUSS平台上注册的航司应用均为“不可用”状态时，则CLA显示“柜机不可用”界面。

4.3.3 CAM

具体来讲，CAM有如下功能：

1.事件分发

CAM的“事件分发”功能，主要是指CAM提供公共事件通道（PublicEventChannel）。无论前台激活状态正在办理旅客，还是后台非激活状态的航司应用，均可以订阅这个公共事件通道里的事件。特别是非激活状态的航司应用，可以根据公共事件通道里发布的事件，决定自身的状态。

航司应用航司应用航司应用航司应用航司应用

2.提供平台环境参数和组件库

航司应用在启动时，通过CAM获得CUSS平台的运行环境参数和组件列表，由这些信息来判断自身能否在平台上运行。

平台运行环境参数包括：柜机位置（机场、航站楼、经纬度等），软件环境（浏览器版本，操作系统版本，CUSS平台版本等）等。

组件列表包括：登机牌打印机组件，护照阅读器组件，条码扫描枪组件等。

CAM除了提供组件列表，还提供各个组件的特性（attribute）。

航司应用可根据自身对各个外设的依赖程度，来决定自身是否在平台上可运行，或者是否全功能运行。

3 访问控制

航司应用对底层外设的访问权限，由CAM和设备组件（DC）来控制。

CAM有义务确保只有处于“激活”状态的航司应用，才有各种外设的访问权限。

同一时间，只有一个航司应用处于“激活”状态。

4.3.4 设备组件（DC）

航司应用不允许直接访问外设。为了实现这一目标，CUSS平台抽象并封装了所有外设组件，并对外提供接口。航司应用程序通过平台的组件接口来访问硬件设备。

一个实际的物理设备，可以根据其功能和特性被抽象成多个设备组件。

例如：一个登机牌打印机可以被抽象为：ATBFeeder（进纸口），ATBMediaOutput（打印）和ATBDispenser（出纸口）。

4.4 平台与航司应用通讯机制

4.4.1 CORBA 通讯

CORBA（Common Object Request Broker Architecture）是一种程序间的通讯机制。

实现：

——跨平台跨语言的数据传输。CUSS 平台和 CUSS 航司应用，可使用不同的编程语言编写。

——实现远程方法的本地调用。允许 CUSS 航司应用通过 CORBA 接口，获取平台提供的对象，并调用该对象提供的方法实现业务功能。

4.4.2 指令和事件

4.4.2.1 说明

指令与事件，是CUSS平台与航司应用之间的通讯机制，通过指令与事件实现设定的功能。

4.4.2.2 指令

航司应用可以通过指令，控制平台的航司应用管理（CAM）或DC（设备组件）做特定的动作。

指令模式见表1：

表 1 指令模式表

同步	指令执行处于阻塞状态，只有执行完成或超时才会返回
异步	指令执行处于异步状态，会立即返回。当指令功能完成后，执行结果会以事件的形式通知调用方

指令分类见表2：

表 2 指令分类表

共享	除了“被挂起”状态，航司应用在任何状态都可以调用。例如，通知 CAM 本航司应用进入“不可用”状态。
独占	只有“激活”状态的航司应用，才能使用。例如，打印登机牌。

4.4.2.3 事件

4.4.2.3.1 总则

CUSS 平台通过事件回调的方式，与航司应用进行交互。

4.4.2.3.2 事件原因

事件可能由以下原因导致：

——硬件故障
——软件故障
——现有错误确认
——错误修复
——任何正常情况变更都可能会修改航司应用或系统管理器的行为
——异步/同步接口调用完成或指令中止

4.4.2.3.3 事件来源

事件可以通过以下方式生成：

——任何一个航司应用程序向系统管理器发送事件
——航司应用管理器
——设备虚拟组件

4.4.2.3.4 事件模式

事件是异步的，由以下两种场景触发,见表 3。

表 3 事件触发场景表

指令关联的	异步指令执行完成后，通过事件向调用方反馈执行结果。
自发的	与指令无关。当航司应用或者平台某组件状态变化时，生成事件。

4.4.2.3.5 事件类别

事件分为三个类别：

——Normal：发生正常处理，而不是检测到错误
——Alert：发生异常情况，但不需要手动干预
——Alarm：需要立即注意（即需要手动干预）

所有的Alert和Alarm事件都必须在当其出现时，发送给系统管理软件。表4是Alarm和Alert的主要区别：

表 4 Alarm 和 Alert 的主要区别表

Alert	Alarm
若问题解决或消失，状态将自动恢复正常	状态保持警报状态，直到 SP 系统管理器确认并由人员解决
遵循严重性设置来解决问题	需要人**立即**做出反应

平台提供商和服务提供商可以通过SLA（Service Level Agreement）设定将事件分类为Alert或Alarm。

4.4.2.3.6 事件类型

事件分为四种类型：

——Public：所有的航司应用和系统管理航司应用都可以接收该类事件。

——Private：只有关联的航司应用和航司系统管理器（请求的事件）才能接收该类事件。

——Platform：只有 SP 系统管理器，CUSS 航司应用管理器，CLA 和 CUSS 组件接口可以接收该类事件。

——Invalid：若以异步模式回调该指令或拒绝同步调用，则返回的事件类型应始终是 invalid。

所有事件必须至少具有上述类型之一。实际类型取决于相关的上下文。

4.4.2.3.7 事件码（EventCode）

标示航司应用或者组件状态的变化。

4.4.2.3.8 状态码（StatusCode）

StatusCode是事件的重要组成部分。主要描述某组件的当前状态。

4.4.2.3.9 事件监听机制

CUSS航司应用监听事件有两种模式：

——模式一，通用事件监听。航司应用调用 RegisterEvent 指令，将可订阅平台发布给该航司应用的所有事件。

——模式二，组件事件监听。航司应用调用某组件的 Acquire 指令，将可订阅该组件生成的所有事件。平台允许航司应用对每个组件，都有一个监听者。

4.4.2.3.10 事件处理

必须按如下方式处理事件：

——所有事件都必须按其显示的方式根据事件类型发送到订阅的航司应用程序。

——SP 系统管理器组件必须记录所有事件和状态变更。

——航司应用程序负责注册事件并跟踪组件状态。

图2是事件转换概述：

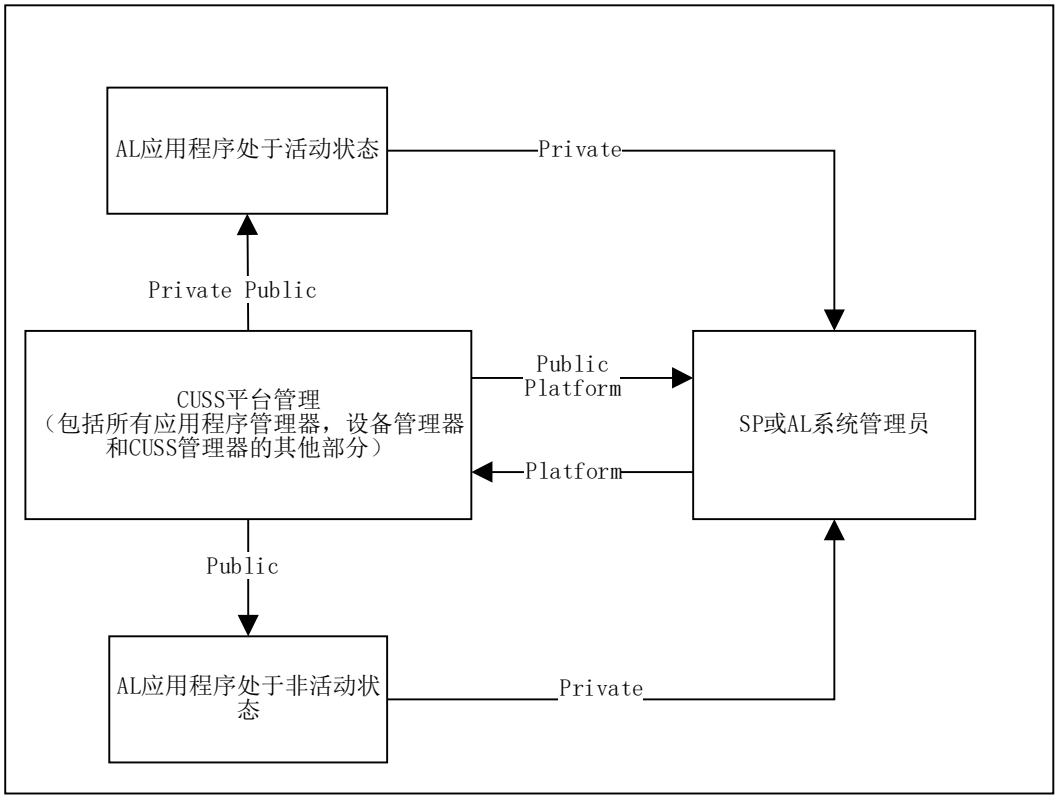

图 2 事件转换概述图

4.5 组件定义

4.5.1 总则

虚拟组件使用一组类定义,这些类已划分为层次结构类。下面定义这些类以及如何将每个虚拟组件映射到这些类。

4.5.2 组件类别

表5是组建表。

表 5 组件表

组件类	描述
Component	组成 CUSS 柜机平台的所有部件。所有组件都派生自此类。
组件	
ManagementInterface	航公航司应用程序管理器或系统管理器接口控制下的组件。
CUSSCntl	CUSS 设备组件控制的组件。
CUSSCntl	
NativeDevice	CUSS 环境使用的设备。只有在本章中提到(如：磁盘、屏幕、网络等)时,

	才能访问它们,而无需使用 CUSS 接口。所有这些本机设备都必须生成事件,以通知航司应用程序和系统管理员其状态和可用性。
ApplicationComponent	用于查询平台上配置的航空公司航司应用程序的状态和/或特征的组件。(不要与实际航司应用混淆)
Peripheral	能够生成要发送到航司应用程序的事件的输入/输出设备。
接口管理	
ApplicationManager	控制平台上所有航空公司航司应用程序的组件。
SystemManagerInterface	实现 SP/航空公司航司应用系统管理器接口的组件。
外围设备	
Input	向航司应用程序提供数据的组件。
Output	能够从航司应用程序接收数据的组件
User	与客户/用户直接交互的组件。
Userless	不与客户/用户交互的组件
Media	使用物理介质的组件(例如卡片、登机牌或纸质文档)
Medialess	不使用物理介质的组件(例如卡片、登机牌或纸质文档)
Data	传输数据的组件。
Dataless	不传输数据的组件。

读取表 5 的另一种方法是：

1 Component

2 ManagementInterface

3 ApplicationManager

4 SystemManagerInterface

5 CUSSCntl

6 NativeDevice

7 ApplicationComponent

8 Peripheral

9 Input

10 Output

11 User

12 Userless

13 Media

14 Medialess

15 Data

16 Dataless

若下列问题中某个答案为"是",则组件被视为与User类(用户输入或用户输出虚拟组件类型)相关:

——用户是否必须以任意方式干扰设备以使数据可用?

——航司应用程序出于任意原因将设备置于禁用状态是否有用?

4.5.3 虚拟组件定义

虚拟组件继承组成此组件的所有类的属性、指令和事件。例如,用户输入虚拟组件由这些类组成:用户、无媒介、数据和输入;因此,此虚拟组件能够处理为这些类或其超类定义的所有内容。此外,虚拟组件由实际组件及其CUSS接口组成。例如,媒介输入由真正的读卡器(硬件)、读卡器供应商驱动程序(软件)和关联CUSS接口组成。

表6显示了每个虚拟CUSSCtnl组件组成的组件类:

表 6 虚拟组建与组件类表

虚拟组件与组件类	
虚拟组件名称	组件类
Application	航司应用程序组件
废纸槽(能够保留介质的组件)	Userless+Media+Dataless
数据输入(用于数据传输的组件(例如数字输入))	Userless+Medialess+Data+input
数据输出(用于用户数据传输的组件(例如屏幕))	Userless+Medialess+Data+output
出纸口(从外围组件接收介质并将其提供给用户或其他外围组件的组件,例如,将 ATB 票据从打印机弹出至暂存设备)	Userless+Media+Dataless
显示设备(例如展台电脑屏幕)	NativeDevice
进纸口(保持介质的组件(例如 ATB 库存)并将其供应到另一个外围组件)	Userless+Media+Dataless
媒介输入 (用于从介质读取的组件(例如磁卡读卡器)	Userless+Media+Data+input
媒介输出 (用于写入介质的组件(例如收据打印机))	Userless+Media+Data+output
网络(处理网络访问的组件)	NativeDevice
存储设备(用于读取/写入存储的组件(例如硬盘))	NativeDevice

用户输入(用于入站用户数据传输的组件(例如声音设备))	Userless+Medialess+Data+input
用户输出(用于出站用户数据传输的组件(例如屏幕))	Userless+Medialess+Data+output

4.5.4 依赖于链接组件的组件

对于某些设备,由于链接组件中的错误,可能无法完成对组件的操作。例如,若链接的出纸口组件中存满了文档, 即使媒介输出组件没有错误，媒介输出组件可能也无法打印。

若调用的组件指令依赖于处于不允许该指令完成的状态的链接组件,则该指令将失败,使用HARDWARE_ERROR或其他故障状态代码(具体取决于链接组件的条件),但调用指令的组件将保持可用。

例如, 在媒介输出组件上打印,若当链接的出纸口组件位于MEDIA_FULL且无法接受更多登机牌时，则Send()请求在MEDIA_FULL时将失败,但媒介输出组件将保持可用。

5 航司应用管理接口规范（ApplicaionManagementInterface）

5.1 航司应用管理接口

5.1.1 总则

航司应用管理接口定义了所有航司应用访问CUSS平台CAM的接口。包括根据航司应用请求，平台将航司应用从一个状态转到另一个状态等等。

5.1.2 航司应用状态描述

以下描述航司应用状态的定义和航司应用状态的转换规则。

CUSS平台的CAM模块，负责管理航司应用状态，并提供初始化请求(initRequest)和通知(notify)两个指令供航司应用使用，操控自身状态。表7是AMI航司应用状态描述表。

表 7 AMI 航司应用状态描述表

状态	描述
未启动状态（STOPPED）	只有 CUSS 的 CAM 能合法启动航司应用。
初始化状态（INITIALIZE）	CUSS 平台启动航司应用后，航司应用进入的第一个状态。 航司应用被启动后，应做如下操作： 1、调用级别(level)指令，获取航司应用令牌(token)。 2、调用注册事件(registerEvent)指令，设置平台时间监听。 3、调用初始化请求(initRequest)指令，向平台请求进入初始化(INITIALIZE)状态。 由于航司应用初始化动作可能耗时很长，并且可能占用硬件资源，因此同一时间，平台只允许一个航司应用处于初始化(INITIALIZE)状态。 航司应用调用初始化请求(initRequest)指令是同步指令，航司应用执行此指令处

	于阻塞状态。只有平台允许航司应用进入初始化(INITIALIZE)状态时，才可以返回。 4、航司应用调用组件指令，拿到平台提供的组件列表以及特性。 5、航司应用结束初始化需要的操作后，调用通知(notify)指令，进入下一个状态。 不可用(UNAVAILABLE)状态：初始化成功，进入到不可用状态。 未启用(STOPPED)状态：初始化失败，航司应用结束。
不可用状态（UNAVAILABLE）	航司应用结束初始化（INITIALIZE）状态后，进入该状态。在此状态下，航司应用检查运行环境，来决定自身是否可进入可用（AVAILABLE）状态。
可用状态（AVAILABLE）	CUSS 平台提供的运行环境满足航司应用要求，航司应用进入可用（AVAILABLE）状态。 在 CLA 界面上，航司应用 logo 可供旅客点选，航司应用处于后台运行，随时待命的状态。 1、旅客点击 CLA 上的 logo，航司应用进入激活（ACITVE）状态。 2、若平台某组件故障（例如打印机缺纸），运行环境无法满足航司应用要求，航司应用会进入不可用状态（UNAVAILABLE）状态。
激活状态（ACTIVE）	旅客点击 CLA 上航司应用 logo，航司应用进入激活状态，给旅客提供服务。 在激活（ACTIVE）状态，航司应用可访问所有组件。 旅客办理完成后，航司应用会重新进入 AVAILABLE 状态。
挂起状态（SUSPENDED）	由于管理要求，平台可将航司应用设置为挂起（SUSPENDED）状态，旅客将无法使用。 国内航司应用场景为，某航空公司的航司应用只有某一个时间段为旅客提供服务，其余时间处于挂起状态。
禁用状态（DISABLE）	由于航司应用非法操作，平台可禁用航司应用。

5.1.3 航司应用状态图

航司应用状态图(见图3)描述的是航司应用如何从一个状态转换为另一个状态。航司应用本身(服务提供者系统管理器）或是航司应用提供者系统管理器要求航司应用状态变更。这些变更伴随一个事件发生，该事件是作为未经请求的事件或作为航司应用程序本身调用的通知（notify）指令的返回事件，由航司应用管理器发送到相关航司应用程序的。注意到，状态转换的数量反映了EventCode之间的响应。粗线条意味着状态转换的发生需要人为干预。

5.1.4 航司应用状态转换描述

5.1.4.1 加载转换（STOPPED 到 INITIALIZE，或 DISABLED 到 INITIALIZE）

用于在系统中加载或重新加载航司应用程序：

——航司应用管理器根据它自己的 AL 或者 SP 系统管理器的要求,用加载指令加载航司应用。

——航司应用管理器在系统重启时加载航司应用程序或航司应用管理器在有人为干预时加

载禁用航司应用。

——在航司应用管理器允许时，航司应用进入初始化（INITIALIZE）状态。

——航司应用正进行初始化时，通用启动航司应用程序会在屏幕中显示“暂时不可用”。

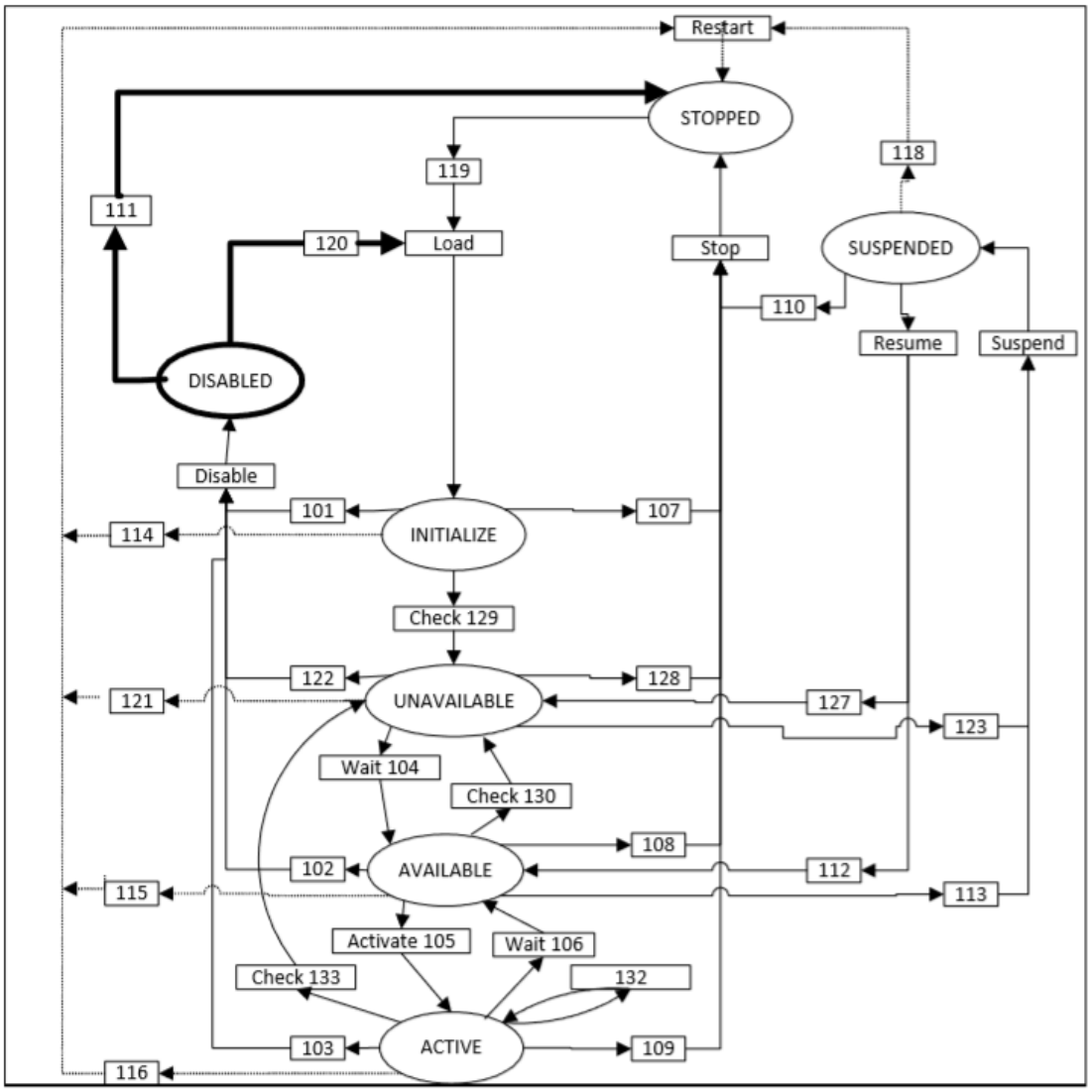

图 3 航司应用状态转换图

5.1.4.2 检查转换（INITIALIZE 到 UNAVAILABLE，或 AVAILABLE 到 UNAVAILABLE 或 ACTIVE 到 UNAVAILABLE）

航司应用已经完成初始化并发现CUSS环境并不适于它的正常运行。因此航司应用让航司应用程序管理员将其状态改为不可用（UNAVAILABLE）状态。

CLA将从屏幕上删除航司应用程序图标或将其显示为不可选择。

5.1.4.3 等待转换（UNAVAILABLE 到 AVAILABLE，或 ACTIVE to 到 AVAILABLE）

航司应用确定CUSS环境可以完成其正确执行或它已完成会话。因此，它请求改变状态为AVAILABLE状态。通用启动航司应用程序显示航司应用图标为可选择。

5.1.4.4 启动转换（AVAILABLE 到 ACTIVE，或 ACTIVE 到 ACTIVE）

用户选择航司应用程序并开始会话。CUSS航司应用管理器根据通用启动航司应用程序的要求控制状态转换。航司应用程序管理器将航司应用程序窗口置于前端。CLA继续显示航司应用程序图标。

5.1.4.5 暂停转换（到 SUPSENDED）

暂停航司应用执行程序；CUSS航司应用管理员根据SP或AL系统管理器的要求控制状态转换。通用航司应用启动程序移除航司应用图标并显示为不可选择[1]。若这是被移除的最后一个图标（即是最后一个被暂停的航司应用），通用航司应用启动程序应该在屏幕上显示“柜机无服务”。

5.1.4.6 恢复转换（返回 PRE-SUSPENDED 状态）

航司应用可执行以下操作：

——只有已经暂停航司应用的系统管理器（SP 或 AL）可以申请这个状态转换。

——航司应用会回到先前的状态（这个状态是被暂停之前的那个状态）。

——若 SP 和 AL 系统管理器暂停了航司应用，它需要在返回其先前状态之前由它们两者恢复（这是为了解决 SP 和 AL 系统管理器操作规则中的潜在冲突）。

——若状态是可用 AVAILABLE 状态，通用航司应用启动程序显示航司应用图标为可选；若航司应用程序状态为 UNAVAILABLE 状态或挂起(SUSPENDED)状态，从屏幕上删除航司应用图标或将其显示为不可选。

5.1.4.7 禁用转换（到 DISABLED）

用于禁用航司应用程序（将航司应用置于禁用区）直到有人为干预：

CUSS航司应用程序管理器将会把航司应用程序置于禁用(DISABLED)状态，由于不正确的行为，例如：

——超出会话时间限制

——航司应用程序阈值错误等等

CUSS航司应用程序管理器停止航司应用程序执行（即卸载它）。

通用航司应用启动程序移除航司应用图标并显示其为不可选择。

5.1.4.8 停止转换（到 STOPPED）

用于停止航司应用执行：

——CUSS 航司应用管理器将航司应用置于未启用(STOPPED)状态，根据航司应用本身和它自有的 AL 或 SP 系统管理器要求。

——CUSS 航司应用程序管理器停止航司应用程序执行（即卸载它）。

通用航司应用启动程序移除航司应用图标并显示其为不可选择[2]。若这是被移除的最后一个图标（即是最后一个被停止的航司应用），通用航司应用启动程序应该在屏幕上显示“柜机无服务”。

5.1.4.9 重启转换

[1]要删除按钮或使其无法选择，则由平台提供商和航司应用程序提供商之间的 SLA 协议决定。 建议平台将此选项配置为可配置。

重启柜机或重启CUSS平台可以激活这个状态转换。CUSS航司应用管理器会把所有的航司应用程序（除了在禁用(DISABLED)状态的航司应用）置于未启用(STOPPED)状态并开始加载。参阅加载转换。

5.2 航司应用管理指令

5.2.1 总则

航司应用以下CORBA地址，获得平台ApplicationManager对象。
corbaloc:<kiosk-IP address>:20000/ApplicationManager或
corbaloc:<kiosk-host name>:20000/ApplicationManager。
ApplicationManager提供以下指令，供航司应用使用。

5.2.2 级别(Level)指令

表8是Level指令表。

表 8 Level 指令表

描述	这是第一个由柜机航司应用程序或系统管理器发布的指令,用于获取有关特定 CUSS 平台实现的基本信息。调用航司应用程序可以验证环境以检查它是否能正确执行到此特定环境实现。若航司应用程序由平台知道(通过平台配置),则此调用将返回航司应用程序引用(令牌)。
执行者	航司应用程序管理类
调用者	INITIALIZE 状态的航司应用 服务提供商的系统管理器 航司应用程序提供程序的系统管理器
访问权限	共享，本地/远程，同步
输入参数	**ApplicationID**：是柜机航司应用程序身份验证的一部分，7.11 节定义结构
执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	Structure of { **SessionTimeout**：number in milliseconds（航司应用程序会话的超时值)。会话是航司应用程序处于 ACTIVE 状态的时间段。当此超时延迟时,将发送"SESSION_TIMEOUT"事件 **KillTimeout**：number in milliseconds（航司应用程序被终止前所剩时间(移动到禁用状态)。在会话超时过期时开始"KillTimeout"。KILL_TIMEOUT 事件将在 KillTimeout 超时延迟后发送。） **KioskLocation**：7.8 节中定义结构 **KioskGPSLocation**：柜机 GPS 坐标，7.9 中定义结构 **KioskID**：柜机航司应用程序身份验证的一部分，7.11 节中定义结构 **CUSSVersion**：Name，包含所有支持的 CUSS 版本的逗号分隔字符串 **CUSSinterfaceVersionSupportedMinimumLevel**：Name

	CUSSinterfaceVersionSupportedMaximumLevel：Name **JVMName**:：Name，使用的 JAVA 虚拟机的名称 **JVMVersion**:：Name，使用的 JAVA 虚拟机版本 **BrowserName**:：Name，已安装的互联网浏览器的名称 **BrowserVersion**:：Name，已安装的互联网浏览器的版本 **OSName**:：Name，已安装的操作系统的名称 **OSVersion**：Name，已安装的操作系统的版本 **ApplicationToken**：7.5 节中定义 }

在异地部署柜机的柜机服务提供商必须将此信息传达给在异地柜机上运行的航司应用程序。这是为允许航空公司在必要时选择退出这些柜机(出于法律、监管或其他原因),航空公司提供商可以调整其航司应用程序以支持场外和机场代码指示器,以调整其航司应用程序的业务逻辑。

5.2.3 组件(Component)指令

表9是Component指令表。

表 9 Component 指令表

描述	这是航司应用程序发出的第二个指令,允许它获取所有已实现虚拟组件的列表、其特征及其 CORBA 对象引用。这将允许航司应用程序检查它所需的所有组件是否已实现。
执行者	航司应用程序管理类
调用者	INITIALIZE 或 ACTIVE 状态的航司应用 服务提供商的系统管理器 航司应用程序提供程序的系统管理器
访问权限	共享，本地/远程，同步
输入参数	**ApplicationToken**：在 7.5 节中定义
执行结果	**FunctionReturnCode**：附录 A.1 节中定义

数据返回	Table [0..**number of virtual components-1**] of structure of {
	VirtualComponentName：Name，在 7.5 节中定义
	VirtualComponentObjectReference：Reference(计算 IOR)
	Real component name：Name(这是仅唯一和特定外围设备必须用于比较的组件，它是用于映射到许多虚拟组件的特定外围设备)
	LinkedComponents:：structure of (
	LinkTable：Table [0..**number of links-1**] of {
	VirtualComponentTableIndex：number number {from 0 to number of

	virtual component-1}（仅限双向直接组件链接） } }

可通过虚拟组件对象引用访问组件特征。所有调用方都将获得所有本机设备和外围组件的完整列表,以及：若调用方是特定航空公司航司应用的所有航司应用程序组件组件的SP系统管理器列表(若调用方是关联的航空公司航司应用系统管理器),则获得与平台上所有配置的航空公司航司应用程序相关的所有航司应用程序组件组件的列表。若调用方是航空公司航司应用程序,则获得其自己的航司应用程序组件组件。

5.2.4 生成事件(GenerateEvent)指令

表10是GenerateEvent指令表。

表 10　　GenerateEvent 指令表

描述	向系统管理器生成事件。航司应用程序不应使用 GenerateEvent 与航司应用程序管理器通信。它应该改为通知。
执行者	管理接口类
调用者	航司应用程序 服务提供商的系统管理器 航司应用程序提供程序的系统管理器
访问权限	共享，本地/远程，同步
输入参数	**ApplicationToken**：7.5 节中定义 **Event**：将要生成的：7.12 节中定义结构
执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：7.12 节中定义结构

5.2.5 查询事件(QueryEvent)指令

表11是QueryEvent指令表。

表 11　　QueryEvent 指令表

描述	返回事件的描述。
执行者	航司应用程序管理类 "CUSSCntl"类所有获取的虚拟组件
调用者	航司应用程序 服务提供商的系统管理器 航司应用程序提供程序的系统管理器
访问权限	共享，本地/远程，同步

输入参数	Structure of{
	ApplicationToken：7.5 节中定义
	EventListSelection：7.12 节中定义结构
	}

执行结果	**FunctionReturnCode**：附录 A.1 节中定义

数据返回	Structure of {
	ListType: set of {CODE, COMPONENT, TYPE} (返回的列表类型将与请求中的列表类型相同)
	List1: Case **list type** of {
	CODE, TYPE: structure of {
	NumberOfCodes: number
	List: Table of [1..**number of codes**] of structure of {
	EventCode: number
	EventType: set of {PRIVATE, PUBLIC, PLATFORM}
	EventDescription: chain of characters
	NumberOfComponent: *number* (代码所航司应用的组件)
	List2: Table [1..**number of components**] of name (虚拟组件名称)
	}
	}
	COMPONENT: structure of {
	NumberOfComponent: number
	List: structure of {
	ComponentName: name *(*虚拟组件名称*)*
	NumberOfCode: number
	List2: Table of [1..**number of codes**] of structure of (
	EventCode: number
	EventType: set of {PRIVATE, PUBLIC, PLATFORM}
	EventDescription: chain of characters
	}
	}
	}
	}
	}

5.2.6 注册事件(RegisterEvent)指令

表12是RegisterEvent指令表。

表 12　　RegisterEvent 指令表

描述	订阅或丢弃接收任何相关事件通知。使用此指令具有附加作用,即调用不会取代以前的事件,而是订阅以前的事件列表加上当前调用中的事件列表。使用此指令完成的所有订阅都将在航司应用程序中通过单个侦听器接收。
执行者	航司应用程序管理类 "CUSSCntl"类所有获取的虚拟组件
调用者	航司应用程序 服务提供商的系统管理器 航司应用程序提供程序的系统管理器
访问权限	共享，本地/远程，同步

输入参数	Structure of {
	ApplicationToken: 7.5节中定义
	Action: set of {SUBSCRIBE, DISCARD} (订阅接收事件,放弃不接收事件)
	EventListSelection: 7.13中定义结构(指定要注册的事件)
	Listener: reference (侦听器的对象引用)
	Correlation: 7.6节中定义(随每个事件发送到侦听器而提交的用户数据)
	}

执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：7.12 节中定义结构

5.2.7 等待事件(WaitEvent)指令

表13是WaitEvent指令表。

表 13　　WaitEvent 指令表

描述	航司应用程序正在等待事件发生。要等待事件,航司应用程序必须通过 **Acquire** (第 6.2.2)或 **RegisterEvent** (第 5.2.6 节)指令订阅它。指令将在事件发生时(列表中的任何或全部)或 timeout 过期时完成。
执行者	航司应用程序管理类 "CUSSCntl"类所有获取的虚拟组件
调用者	航司应用程序
访问权限	共享，本地/远程，同步

输入参数	Structure of {
	Timeout：7.4节中定义
	Application Token：7.5节中定义
	Event list selection：7.13节中定义结构
	}

执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：7.12 节中定义结构

5.2.8 初始化请求(InitRequest)指令

表14是InitRequest指令表。

表 14　　InitRequest 指令表

描述	航司应用程序现在要初始化/重新初始化。这是一个阻塞调用（BlockingCall）。在此指令返回后,允许航司应用程序初始化。此处理可确保对所有航司应用程序进行初始化序列化。这是必要的,因为航司应用程序可能会加载 PECTAB 到 ATB 打印机,一次只能由一个航司应用程序完成。
执行者	航司应用程序管理类
调用者	航司应用程序管理器加载后处于"已停止"状态的航空公司航司应用程序
访问权限	共享，本地/远程，同步
输入参数	**ApplicationToken**：7.5 节中定义
执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：7.12 节中定义结构

5.2.9 通知(Notify)指令

表15是Notify指令表。

表 15　　Notify 指令表

描述	航司应用程序使用此指令从 CUSS 航司应用程序管理器请求状态更改,若请求获得批准,它将更改航司应用程序状态。
执行者	航司应用程序管理类
调用者	邻域状态（NeighborhoodState）的航司应用程序
访问权限	共享，本地/远程，同步

输入参数	Structure of {
	ApplicationToken：7.5节中定义，请求航司应用程序的令牌
	KioskApplicationID: 7.11节中定义，状态改变的柜机航司应用程序.
	State Transition: number (附录A.2中定义的EventCode[101 到 130])
	}

执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：7.12 节中定义结构

5.3 回调(Callback)指令与事件

5.3.1 总则

航司应用必须使用航司应用管理者提供的RegisterEvent指令在平台注册一个事件监听（EventListener），平台调用EventListener的Callback指令与航司应用通讯。

5.3.2 Callback 指令

表16是Callback指令表。

表 16　　Callback 指令表

描述	向之前注册的监听器发送事件
执行者	航司应用程序 服务提供商系统管理器 提供航司应用程序系统管理器
调用者	航司应用程序管理类 所有“外围设备”类的虚拟组件
访问权限	本地/远程，同步
输入参数	**Event**：7.12 节中定义结构
数据返回	**None**

5.3.3 CUSS 航司应用管理 Callback 事件

CUSS平台的CAM会生成航司应用状态变化事件，发送给所有注册的监听（航司应用），或者直接返回给调用者。表17是CUSS航司应用管理Callback EventCode表。

表 17　　CUSS 航司应用管理 Callback EventCode 表

EventCode		
EventCode	描述	
000	状态转换暂停，恢复，停止=EC_OK 在返回事件中用于调用暂停、恢复、停止等指令。	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	004	SOFTWARE_ERROR
	802	SP_SYSTEM_MANAGER_REQUEST
	803	AL_SYSTEM_MANAGER_REQUEST

航司应用程序状态转换		
101	状态转换禁用 = INITIALIZE_DISABLED 由 CUSS 航司应用程序管理者请求	
	相关的 StatusCode	
	303	CRITICAL_SOFTWARE_ERROR
	305	NOT_RESPONDING
	306	THRESHOLD_ERROR
	801	CUSS_MANAGER_REQUEST
102	状态转换禁用 = AVAILABLE_DISABLED 由 CUSS 航司应用程序管理者请求	
	相关 StatusCode	
	303	CRITICAL_SOFTWARE_ERROR
	305	NOT_RESPONDING
	306	THRESHOLD_ERROR
	801	CUSS_MANAGER_REQUEST
103	状态转换禁用 = ACTIVE_DISABLED 由 CUSS 航司应用程序管理者请求	
	相关 StatusCode	
	303	CRITICAL_SOFTWARE_ERROR
	305	NOT_RESPONDING
	306	THRESHOLD_ERROR
	310	KILL_TIMEOUT
	801	CUSS_MANAGER_REQUEST
104	状态转换等待= UNAVAILABLE_AVAILABLE 通过一个航司应用程序请求	
	相关 StatusCode	
	805	AL_APPLICATION_REQUEST
105	状态转换激活= AVAILABLE_ACTIVE 由 CUSS 航司应用程序管理器根据来自通用启动航司应用程序的信息请求	
	相关 StatusCode	
	804	AL_APPLICATION_REQUEST

106	状态转换等待= ACTIVE_AVAILABLE 通过一个航司应用程序请求	
	相关 StatusCode	
	804	AL_APPLICATION_REQUEST
107	状态转换停止= INITIALIZE_STOPPED_STOP 由航司应用程序，CUSS 航司应用程序管理者或系统管理者请求	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	004	SOFTWARE_ERROR
	801	CUSS_MANAGER_REQUES
	802	SP_SYSTEM_MANAGER_REQUEST
	803	AL_SYSTEM_MANAGER_REQUEST
	804	AL_APPLICATION_REQUEST
108	状态转换停止= AVAILABLE_STOPPED _STOP 由航司应用程序，CUSS 航司应用程序管理者或系统管理者请求	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	004	SOFTWARE_ERROR
	801	CUSS_MANAGER_REQUEST
	802	SP_SYSTEM_MANAGER_REQUEST
	803	AL_SYSTEM_MANAGER_REQUEST
	804	AL_APPLICATION_REQUEST
109	状态转换停止=ACTIVE_STOPPED _STOP 由航司应用程序，CUSS 航司应用程序管理者或系统管理者请求	
	相关 StatusCode	
	000	OK
	001	TIMEOUT

	002	WRONG_STATE
	004	SOFTWARE_ERROR
	801	CUSS_MANAGER_REQUEST
	802	SP_SYSTEM_MANAGER_REQUEST
	803	AL_SYSTEM_MANAGER_REQUEST
	804	AL_APPLICATION_REQUEST
110	状态转换停止= = SUSPENDED_STOPPED_STOP 由 AL 或 SP 系统管理请求(能够请求此状态更改的只有已将航司应用程序置于挂起状态的那个)	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	004	SOFTWARE_ERROR
	802	SP_SYSTEM_MANAGER_REQUEST
	803	AL_SYSTEM_MANAGER_REQUEST
111	状态转换停止= = DISABLED_STOPPED_STOP 由 CUSS 航司应用程序管理者或 SP 系统管理者请求	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	004	SOFTWARE_ERROR
	801	CUSS_MANAGER_REQUEST
	802	SP_SYSTEM_MANAGER_REQUEST
112	状态转换恢复= SUSPENDED_AVAILABLE 由 SP 或航司系统管理员请求	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE

	004	SOFTWARE_ERROR
	802	SP_SYSTEM_MANAGER_REQUEST
	803	AL_SYSTEM_MANAGER_REQUEST
113	状态转换挂起=AVAILABLE_SUSPENDED 由 SP 或航司系统管理员请求	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	004	SOFTWARE_ERROR
	802	SP_SYSTEM_MANAGER_REQUEST
	803	AL_SYSTEM_MANAGER_REQUEST
114	状态转换重启=INITIALIZE_STOPPED_RESTART 由 CUSS 航司应用程序管理者或 SP 系统管理者或航司应用程序请求	
	相关 StatusCode	
	801	CUSS_MANAGER_REQUEST
	802	SP_SYSTEM_MANAGER_REQUEST
115	状态转换重启=AVAILABLE_STOPPED_RESTART 由 SP 系统管理者或者 CUSS 航司应用程序管理者或者航司应用程序请求	
	相关 StatusCode	
	801	CUSS_MANAGER_REQUEST
	802	SP_SYSTEM_MANAGER_REQUEST
	804	AL_APPLICATION_REQUEST
116	状态转换重启=ACTIVE_STOPPED_RESTART 由 CUSS 航司应用程序管理者或 SP 系统管理者或航司应用程序请求	
	相关 StatusCode	
	801	CUSS_MANAGER_REQUEST
	802	SP_SYSTEM_MANAGER_REQUEST
	804	AL_APPLICATION_REQUEST
118	状态转换重启=SUSPENDED_STOPPED_RESTART 由 CUSS 航司应用程序管理者或 SP 系统管理者或航司应用程序请求	

	相关 StatusCode	
	801	CUSS_MANAGER_REQUEST
	802	SP_SYSTEM_MANAGER_REQUEST
	804	AL_APPLICATION_REQUEST
119	状态转换加载=STOPPED_INITIALIZE 由 CUSS 航司应用程序管理者或系统管理者请求	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	003	CANCELLED
	004	SOFTWARE_ERROR
	303	CRITICAL_SOFTWARE_ERROR
	305	NOT_RESPONDING
	801	CUSS_MANAGER_REQUEST
	802	SP_SYSTEM_MANAGER_REQUEST
	803	AL_SYSTEM_MANAGER_REQUEST
120	状态转换加载=DISABLED_INITIALIZE 在人的介入后由 CUSS 航司应用程序管理者或 SP 系统管理者请求	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	003	CANCELLED
	004	SOFTWARE_ERROR
	303	CRITICAL_SOFTWARE_ERROR
	305	NOT_RESPONDING
	801	CUSS_MANAGER_REQUEST
	802	SP_SYSTEM_MANAGER_REQUEST
121	状态转换重启=UNAVAILABLE_STOPPED_RESTART 由 CUSS 航司应用程序管理者或 SP 系统管理者或航司应用程序请求	

	相关 StatusCode	
	801	CUSS_MANAGER_REQUEST
	802	SP_SYSTEM_MANAGER_REQUEST
	804	AL_APPLICATION_REQUEST
122	状态转换禁用=UNAVAILABLE_STOPPED_RESTART 由 CUSS 航司应用程序管理者或 SP 系统管理者或航司应用程序请求	
	相关 StatusCode	
	303	CRITICAL_SOFTWARE_ERROR
	305	NOT_RESPONDING
	306	THRESHOLD_ERROR
	801	CUSS_MANAGER_REQUEST
123	状态转换挂起=UNAVAILABLE_SUSPENDED 由 SP 或者航司系统管理员请求	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	003	CANCELLED
	004	SOFTWARE_ERROR
	802	SP_SYSTEM_MANAGER_REQUEST
	803	AL_SYSTEM_MANAGER_REQUEST
127	状态转换恢复=SUSPENDED_UNAVAILABLE 由航司应用程序或 SP 系统管理员请求	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	003	CANCELLED
	004	SOFTWARE_ERROR
	802	SP_SYSTEM_MANAGER_REQUEST
	803	AL_SYSTEM_MANAGER_REQUEST

128	状态转换停止=SUSPENDED_UNAVAILABLE 由航司应用程序或 SP/航司系统管理员请求	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	004	SOFTWARE_ERROR
	802	SP_SYSTEM_MANAGER_REQUEST
	803	AL_SYSTEM_MANAGER_REQUEST
	804	AL_APPLICATION_REQUEST
129	状态转换检查=INITIALIZE_UNAVAILABLE 由航司应用程序请求	
	相关 StatusCode	
	804	AL_APPLICATION_REQUEST
130	状态转换检查=AVAILABLE_UNAVAILABLE 由航司应用程序请求	
	相关 StatusCode	
	804	AL_APPLICATION_REQUEST
132	状态转换等待=ACTIVE_ACTIVE 由航司应用程序请求	
	相关 StatusCode	
	804	AL_APPLICATION_REQUEST
133	状态转换等待=ACTIVE_UNAVAILABLE 由航司应用程序请求	
	相关 StatusCode	
	804	AL_APPLICATION_REQUEST
202	状态：UNAVAILABLE 无状态转换。航司应用程序是 UNAVAILABLE 状态	
	相关 StatusCode	
	000	OK
	001	TIMEOUT

	002	WRONG_STATE
	004	SOFTWARE_ERROR
204	状态：STOPPED 无状态转换。航司应用程序是未启用(STOPPED)状态	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	004	SOFTWARE_ERROR
206	状态：DISABLED 无状态转换。航司应用程序是禁用(DISABLED)状态	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	004	SOFTWARE_ERROR
	303	CRITICAL_SOFTWARE_ERROR
	305	NOT_RESPONDING
	306	THRESHOLD_ERROR
	310	KILL_TIMEOUT
207	状态：INITIALIZE 无状态转换。航司应用程序是初始化（INITIALIZE）状态。	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	004	SOFTWARE_ERROR
208	状态：AVAILABLE 无状态转换。航司应用程序是可用（AVAIABLE）状态。	
	相关 StatusCode	
	000	OK

	001	TIMEOUT
	002	WRONG_STATE
	004	SOFTWARE_ERROR
209	状态：ACTIVE 无状态转换。航司应用程序是激活(ACTIVE)状态。	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	004	SOFTWARE_ERROR
	309	SESSION_TIMEOUT

6 设备组件接口规范（DeviceComponentInterface）

6.1 设备组件接口

6.1.1 总则

设备组件接口DCI是基于为航司应用定义的虚拟环境。下文描述此虚拟环境，包括虚拟组件可以处于的所有状态的描述及其关联的状态转换。

6.1.2 虚拟组件概念

CUSS平台提供虚拟组件与真实设备的映射，航司应用通过虚拟组件实现自自助功能。航司应用CUSS组件管理概念还包括虚拟组件链接。所有这些虚拟组件都可以在一个或许多实体设备中实现。与同一个实体外设相关的所有虚拟组件必须具有相同的实体组件名称，从而允许航司应用（若需要）知道该事实。以相同的方式，所有链接的虚拟组件将通过虚拟组件链表在表中交叉链接，允许航司应用程序在内部构建虚拟组件链，而无需知道这些虚拟组件是在一个还是在多个实体设备中实现的，一个实体组件（例如，外设）可以映射到一个或多个虚拟组件。

根据定义，每种媒介类型的每个功能每个实体组件应该有一个虚拟组件（后者仅适用于基于媒介的外设）。请参阅Appx B 组建影射以检查其他实体组件如何映射到虚拟组件。

从虚拟环境到真实环境的映射由CUSS实体：对象，方法（指令）等完成。航司应用通过接口调用访问这些实体。这些接口使用基于CORBA的面向对象编程技术，以实现为航司应用定义CUSS工作环境的功能和事件。

一个虚拟组件上的一个指令，一次只允许对一种库存类型执行一个操作。

由于接口确切地知道哪个功能用于哪个组件的哪个目的，接口将负责调整需求以确保接口调用正确行为。

事实上，虚拟组件被作为对象网络实现，由表示航司应用于实体组件和状态的指令和事件的方法处理。这允许通过仅修改所需的CUSS接口实现（底层方法）来更改任何柜机实现，而无需更改任何设计良好的航司应用。

所有虚拟组件都有给定设备类型的标准化特征。CUSS平台为航司应用程序提供了这些特性，并允许它在必要时更改其中一些特性。有关每个虚拟组件的特征列表，请参阅第8章：虚拟组件属性。

6.1.3 设备组件接口规则

所有ATB和GPP打印机接口（其虚拟组件都被指定为登机牌打印机）必须支持AEA数据流（GPP不包括磁条）。所有指定为行李条打印机的虚拟机都必须支持AEA数据流。

所有GPP必须支持SVG或W3C标准格式化消息。

任何文件都可以打印：

——在 ATB2 设备上：使用 AEA 数据流

——在 GPP 打印机上：使用 AEA 或 SVG 数据流；必须在柜机中没有实现 ATB 打印机时实现前者，否则它是可选的（对于该柜机）

只有ATB2设备才会使用AEA数据流进行读取。

非ATB读卡器（媒介输入）使用MSG数据流进行读取。

非ATB打印机（媒介输出）支持用于打印的标准SVG数据流。即使实体设备没有这样的实体功能，每个媒介输出组件也与出纸口组件相关联。在这种情况下，打印机输出路径被认为是出纸口。由于出纸口组件不是实体的，因此不需要Offer指令将文档传递给用户。

前一段也适用于必须将媒介返回给用户的媒介输入组件。

航司应用不能使用本机命令，组件接口也不支持本机命令。

打印机/读卡器内存管理：

——若 n 个航司在打印机或阅读器上有一个航司应用程序，则每个航司应用程序将具有 1/n 的打印机或读卡器内存。

——每个航司将具有至少 120KB 的可用内存（这提供 4K 的 6 个 PECTAB，2K 的 4 个模板，10K 的 8 个 logo）。

——若航司应用需要打印机或读卡器上下载 PECTAB、logo 或任何其他文件，但没有更多可用内存，则航司文件请求者之一将从打印机或阅读器中的上下文中卸载。

——项目只会下载一次，CUSS 接口将会在航司应用程序每次下载项目时缓存，若航司应用程序需要时未加载到打印机或读卡器上，CUSS 接口将检索该项目。

字符集：

相关文件

ISO/IEC 10646-1 第二版

Unicode 3.0.0 (http://www.unicode.org)

CUSS标准是为航司应用程序提供商提供一个平台，用以编写航司应用程序，并在全球任何一个符合CUSS标准的柜机系统上运行该航司应用程序，为获得世界各地的客户支持和为了符合国家标准，航司应用程序提供者必须以这种方式编写程序，并且可以支持不同的语言和不同的字符集。这并不意味着平台的所有设备都必须支持多种语言或字符集，只有与用户交互的那些必须支持这一点。因为航空业中的大多数数据流都是基于ASCII的，另一个标准（IATA，AEA ...）已经涵盖的设备或其数据流不必支持除ASCII之外的其他字符集。但是，仅由CUSS标准涵盖的操作系统和设备必须至少支持ISO 8859-1（Latin 1）和双字节。那些必须支持其他字符集的必须使用Unicode 3.0.0。出于兼容性原因，ASCII必须使用Unicode UTF-8编码。UTF-16可以转换为UTF-8而不会丢失信息。

6.1.4 设备组件状态描述

表18是设备组件状态表。

表 18　　设备组件状态表

节点描述	
状态	**描述**
释放状态（RELEASED）	这是设备组件的初始状态。在RELEASED状态下任何航司应用程序都可以获取该组件(供使用)。一旦组件被航司应用程序获取，它不会阻止其他航司应用程序同时获取它。这意味着多个航司应用程序可以同时获取相同的组件。 在此状态下不允许使用组件 Query 指令。
就绪状态（READY）	READY 状态告诉航司应用程序，组件现在已准备好接收并执行航司应用程序给出的任何函数或指令。在接收指令时，组件变为 BUSY 状态并保持，直到指令返回错误或 OK。 只有持有活动令牌的航司应用程序才能执行任何独占指令。所有其他航司应用程序都无权这样做。 从 READY 状态直接进入 EVENTHANDLING 状态，然后进入 UNAVAILABLE 状态（例如手动关闭打印机）时，会产生自发事件事件。 航司应用程序可以释放组件。
繁忙状态（BUSY）	完全瞬态状态，表明组件正在执行由航司应用指令（例如，读或写）。 在此状态下调用组件 Query 指令，应返回上次已知状态。若请求是有效的，则所有其他指令将排队。
事件正在处理状态（EVENTHANDLING）	也是一个瞬态状态。定义组件在处理事件。 事件可能由航司应用调用组件指令或组件自发的内部检查产生。
不可用状态（UNAVAILABLE）	定义不可恢复的错误条件，该条件不允许在组件上执行任何指令。 航司应用可以根据某组件在转换到 UNAVAILABLE 状态时产生的事件，来决定自身是否需要在 CLA 上显示。 组件可以通过接收硬件驱动消息，或者自身检查，或者人为干预（例如移除卡纸）得知硬件恢复，此时将进入 READY 状态。 若航司应用在之前获得了 UNAVAILABLE 状态组件，那么航司应用可以释放该组件。 在此状态下允许组件 Query 指令。

6.1.5 设备组件状态图

图4是设备组件状态图（航司应用视图），定义了CUSS组件的常见行为，而不是它的实现。但它确切地定义了必须发送事件的时间以及必须发生的事件类型。它还定义了事件发生的顺序。航司应用程序查看的状态仅用于定义转换和事件。

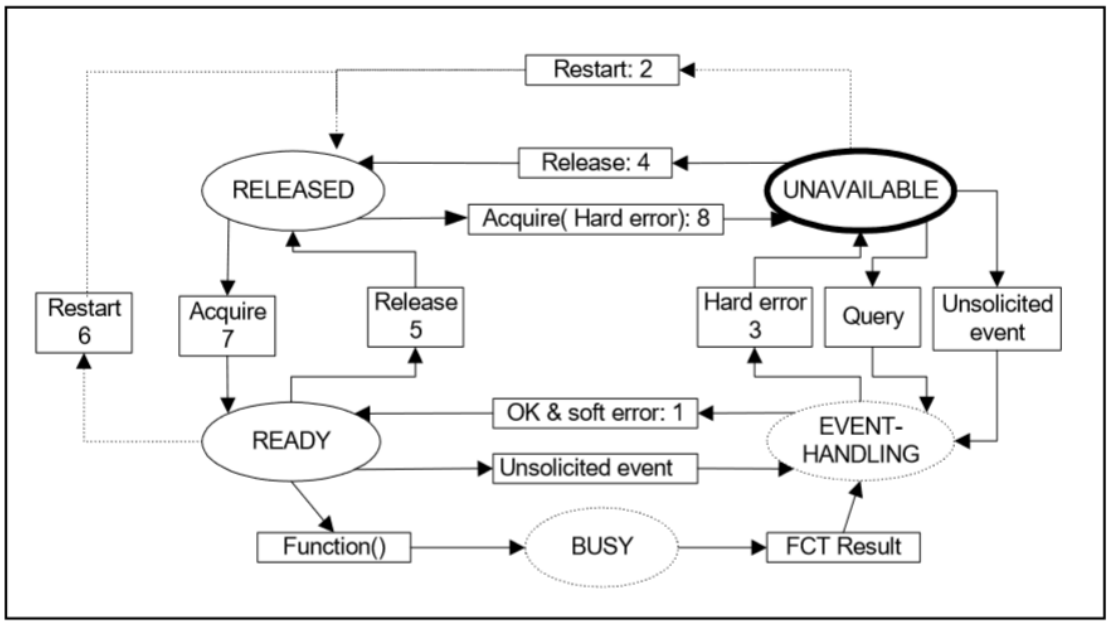

图 4 设备组件状态图（航司应用视图）

在该图中，虚线椭圆表示瞬态。此外，此图中的EventCode编号与事件可能发生的顺序无关。编号仅用于标识不同的事件。有关这些事件的更多详细信息，。

无论虚拟组件状态的航司应用程序视图是什么，平台行为都是相同的。若航司应用程序认为组件处于READY状态，而它实际上是处于UNAVAILABLE状态。这是为了确保航司应用程序可以使用状态转换来了解方法调用是成功还是不成功。

6.1.6 设备组件状态转换描述

表19是设备组件状态转换表。

表 19　　设备组件状态转换表

状态转换	
转换	**描述**
获取（Acquire）	组件的 Acquire 指令由航司应用程序调用，组件或者转换为 READY 状态或者为 UNAVAILABLE 状态。
释放（Release）	组件的 Release 指令由航司应用程序调用，组件将进入 RELEASED 状态。
功能（Function）	这可以是以下功能之一，具体取决于组件： Retain, Offer 基于媒介的组件 Receive 输入组件 Send 输出组件 Enable, Disable 基于用户的组件 Test, Setup, Cancel 继承自 Peripheral 类的所有组件

	Query 所有组件 组件将保持 READY 状态或进入 UNAVAILABLE 状态。
未请求的事件（UnsolicitedEvent）	外部事件，例如：卡/票已插入、卡纸、设备无法访问、设备现已正常、打印机缺纸、纸张不足、纸张现在正常等等。 根据事件，生成的组件状态或者为 READY 状态或者为 UNAVAILABLE 状态。
重启（Restart）	组件由授权平台组件发布。当系统重新启动时，或当 CUSS 航司应用程序管理器将航司应用程序置于禁用(DISABLED)状态时，或航司应用程序停止并且航司应用程序本身未释放其获取的虚拟组件，可能会发生这种情况。 该组件将进入 RELEASED 状态。

6.2 设备组件接口(DCI)指令

6.2.1 总则

CUSS控制着DCI虚拟组件管理指令集。它们分为四类:组件指令集、数据指令集、文档指令集和事件指令集。

组件指令：Acquire, Disable, Enable, Query, Release, Setup 和Test。

数据指令：Receive 和Send。

文档指令：Offer 和Retain。

事件指令：Cancel。

6.2.2 获取（Acquire）指令

表20是Acquire指令表。

表 20　　Acquire 指令表

描述	使虚拟组件可用于航司应用程序。航司应用程序可以同时申请特定的组建监听器，这些监听器与获取的组件关联。
执行者	“外部”或“本地设备”类的所有已发布虚拟组件。这些虚拟组件的初始状态为“已释放”。
调用者	在 INITIALIZE、UNAVAILABLE、AVAILABLE 或者 ACTIVE 状态的航司应用程序 服务提供商系统管理器 提供航司应用程序的系统管理器
访问权限	共享，本地/远程，同步
输入参数	Structure of { **Timeout:** 调用的Timeout值；第7.4节中定义 **ApplicationToken:** 在第7.5节中定义 **EventListSelection:** 在第7.13节定义中结构（指定要注册的组件事件，即事件筛选） **Listener**: reference (监听器的对象引用)

	Correlation: 在第 7.6 节中定义(随每个事件发送到侦听器而提交的用户数据) }
执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：7.12 节中定义结构

返回事件中的StatusCode，代表设备的状态。对于不同的虚拟组件，可能返回的结果不同。详见表21。

表 21　　Acquire 指令 StatusCode 表

Acquire	虚拟组件类型											
StatusCode	Capture	DataInput	DataOutput	UserInput	UserOutput	Dispenser	Feeder	MediaInput	MeidaOutput	Display	Network	Storage
OK	×	×	×	×	×	×	×	×	×	×	×	×
TIMEOUT	×	×	×	×	×	×	×	×	×	×	×	×
WRONG_STATE	×	×	×	×	×	×	×	×	×	×	×	×
CANCELLED	×	×	×	×	×	×	×	×	×	×	×	×
SOFTWARE_ERROR	×	×	×	×	×	×	×	×	×	×	×	×
MEDIA_JAMMED						×	×	×	×			
MEDIA_MISPLACED						×	×	×	×			
MEDIA_PRESENT						×		×	×			
MEDIA_ABSENT						×		×	×			
MEDIA_HIGH	×					×						×
MEDIA_FULL	×					×						×
MEDIA_LOW							×					
MEDIA_EMPTY							×					
MEDIA_DAMAGED												×
MEDIA_INCOMPLETELY_INSERTED								×				
CONSUMABLES									×			
HARDWARE_ERROR	×	×	×	×	×	×	×	×	×	×	×	×

CRITICAL_SOFTWARE_ERROR	×	×	×	×	×	×	×	×	×	×	×	×
NOT_REACHABLE	×	×	×	×	×	×	×	×	×	×	×	×
NOT_RESPONDING	×	×	×	×	×	×	×	×	×	×	×	×
THRESHOLD_ERROR	×	×	×	×	×	×	×	×	×	×	×	×
THRESHOLD_USAGE	×	×	×	×	×	×	×	×	×	×	×	×
CONFIGURATION_ERROR		×	×	×	×			×	×	×	×	×

6.2.3 禁用（Disable）指令

表22是Disable指令表。

表 22　　Disable 指令表

描述	使虚拟组件对用户不可用(例如,从文档插入中禁用读取设备)。
执行者	"用户"类所有获取和启用的虚拟组件(不包括“本地设备”组件)。
调用者	处于 ACTIVE 状态的航司应用程序。 服务提供商系统管理器
访问权限	独占，本地/远程，同步/异步
输入参数	**Timeout**：调用的 Timeout 值;第 7.4 节中定义 **ApplicationToken:**：在第 7.5 节中定义
执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：7.12 节中定义结构

无论处理多少文档，平台都将确保在启用状态下，航司应用程序可以调用enable()的任何组件,直到该航司应用程序调用disable()。

在某些情况下,组件实际上会因其物理限制而禁用。例如,电动读卡器在提供、删除或保留该卡之前，无法读取其他卡，即使航司应用程序在逻辑上启用该卡也是如此。若由于此或任何其他原因，物理组件在读取文档后(例如由于其固件逻辑)而禁用,则平台必须自动重新启用它(除非航司应用程序显式的直接立即禁用该组件阅读)。

在返回事件中的StatusCode表示函数调用状态,并依赖于此指令航司应用于的虚拟组件,如表23所示:

表 23　　Disable 指令 StatusCode 表

Disable	虚拟组件类型				
StatusCode	UserInput	UserOutput	Dispenser	MediaIuput	MediaOutput

OK	×	×	×	×	×
TIMEOUT	×	×	×	×	×
WRONG_STATE	×	×	×	×	×
CANCELLED	×	×	×	×	×
SOFTWARE_ERROR	×	×	×	×	×
OUT_OF_SEQUENCE	×	×	×	×	×
DATA_PRESENT	×	×			
CONSUMABLES					×
HARDWARE_ERROR	×	×	×	×	×
CRITICAL_SOFTWARE_ERROR	×	×	×	×	×
NOT_REACHABLE	×	×	×	×	×
NOT_RESPONDING	×	×	×	×	×
THRESHOLD_ERROR	×	×	×	×	×
THRESHOLD_USAGE	×	×	×	×	×
CONFIGURATION_ERROR	×	×	×	×	×

6,2.4 启用（Enable）指令

表24是Enable指令表。

表 24　　Enable 指令表

描述	使虚拟组件可供用户使用(例如,为文档插入启用读取器设备)。
执行者	“用户”类所有获取和禁用的虚拟组件(不包括“本地设备”组件)。默认情况下,获取所有虚拟组件时将禁用它们。
调用者	处于激活(ACTIVE)状态的航空公司航司应用程序。 服务提供商系统管理器
访问权限	独占，本地/远程，同步/异步
输入参数	**Timeout**：调用的 Timeout 值；在第 7.4 节中定义 **ApplicationToken**：在第 7.5 节中定义
执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：在 7.12 节中定义结构

无论处理多少文档，平台都将确保在启用状态下，航司应用程序可以调用enable()的任何组件,直到该航司应用程序调用disable()。

若一个组件(出纸口)的媒介状态依赖于链接组件的动态(例如媒介输出打印组件)，那么平台将要产生一个MEDIA事件(PRESENT,HIGH,FULL,ABSENT,etc)，即使这个航司应用程序没有启用出纸口组件。例如，若出纸口有具有能够检测打印文档的传感器，那么出纸口打印机就会向活动的航司应用程序广播一个专有的MEDIA_PRESENT事件，即使这个航司应用程序没有调用出纸口上的enable()。

在某些情况下,组件实际上会因其物理限制而禁用。例如,电动读卡器在提供、删除或保留该卡之前，无法读取其他卡，即使航司应用程序在逻辑上启用该卡也是如此。若由于此或任何其他原因，物理组件在读取文档后(例如由于其固件逻辑)而禁用,则平台必须自动重新启用它(除非航司应用程序显式的直接立即禁用该组件阅读)。

在返回事件中的StatusCode表示函数调用状态,并依赖于此指令航司应用于的虚拟组件,如表25所示:

表 25　　Enable 指令 StatusCode 表

Enable	虚拟组件类型				
StatusCode	UserInput	UserOutput	Dispenser	MediaIuput	MediaOutput
OK	×	×	×	×	×
TIMEOUT	×	×	×	×	×
WRONG_STATE	×	×	×	×	×
CANCELLED	×	×	×	×	×
SOFTWARE_ERROR	×	×	×	×	×
OUT_OF_SEQUENCE	×	×	×	×	×
MEDIA_JAMMED			×	×	×
MEDIA_MISPLACED			×	×	×
MEDIA_PRESENT			×	×	×
MEDIA_ABSENT			×	×	×
MEDIA_HIGH			×		
MEDIA_FULL			×		
MEDIA_INCOMPLETELY_INSERTED				×	
DATA_PRESENT	×	×			
CONSUMABLES					×
HARDWARE_ERROR	×	×	×	×	×
CRITICAL_SOFTWARE_ERROR	×	×	×	×	×
NOT_REACHABLE	×	×	×	×	×

NOT_RESPONDING	×	×	×	×	×
THRESHOLD_ERROR	×	×	×	×	×
THRESHOLD_USAGE	×	×	×	×	×
CONFIGURATION_ERROR	×	×	×	×	×

6.2.5 查找（Query）指令

表26是Query指令表。

表 26　　Query 指令表

描述	返回虚拟组件的状态（state）/状况(status)
执行者	所有获取的虚拟组件：“外围设备”类，“本机设备”类和“航司应用程序组件”类的虚拟组件。
调用者	处于 INITIALIZE、UNAVAILABLE、AVAILABLE 或 ACTIVE 状态的航司应用程序 服务提供商系统管理器 航司应用程序提供系统管理器
访问权限	共享，本地/远程，同步/异步
输入参数	**Timeout**：调用的 Timeout 值；在第 7.4 节中定义 **ApplicationToken**：在第 7.5 节中定义
执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：在 7.12 节中定义结构

在返回事件中的StatusCode表示设备状态(或设备组件忙时的最后一个已知设备状态)，并依赖于此指令航司应用于的虚拟组件，如表27所示:

表 27　　Query 指令 StatusCode 表

Query	虚拟组件类型											
StatusCode	Capture	DataInput	DataOutput	UserInput	UserOutput	Dispenser	Feeder	MediaInput	MeidaOutput	Display	Network	Storage
OK	×	×	×	×	×	×	×	×	×	×	×	×
TIMEOUT	×	×	×	×	×	×	×	×	×	×	×	×
WRONG_STATE	×	×	×	×	×	×	×	×	×	×	×	×
CANCELLED	×	×	×	×	×	×	×	×	×	×	×	×
SOFTWARE_ERROR	×	×	×	×	×	×	×	×	×	×	×	×

MEDIA_JAMMED	×					×	×	×	×			
MEDIA_MISPLACED						×	×	×	×			
MEDIA_PRESENT						×		×	×			
MEDIA_ABSENT						×		×	×			
MEDIA_HIGH	×					×						×
MEDIA_FULL	×					×						×
MEDIA_LOW							×					
MEDIA_EMPTY							×					
MEDIA_DAMAGED												×
MEDIA_INCOMPLETELY_INSERTED								×				
CONSUMABLES									×			
HARDWARE_ERROR	×	×	×	×	×	×	×	×	×	×	×	×
CRITICAL_SOFTWARE_ERROR	×	×	×	×	×	×	×	×	×	×	×	×
NOT_REACHABLE	×	×	×	×	×	×	×	×	×	×	×	×
NOT_RESPONDING	×	×	×	×	×	×	×	×	×	×	×	×
THRESHOLD_ERROR	×	×	×	×	×	×	×	×	×	×	×	×
THRESHOLD_USAGE	×	×	×	×	×	×	×	×	×	×	×	×
CONFIGURATION_ERROR		×	×	×	×			×	×	×	×	×

6.2.6 释放（Release）指令

表28是Release指令表。

表 28　　Release 指令表

描述	使虚拟组件对航司应用程序不可用,并取消订阅事件侦听器对于组件的释放。若航司应用程序本身未取消所有挂起的异步指令,将自动取消。
执行者	所有获取的虚拟组件：“外围设备”类和“本机设备”类
调用者	处于初始化、不可用、可用或 ACTIVE 状态的航司应用程序 服务提供商系统管理器 航司应用程序提供系统管理器
访问权限	共享，本地/远程，同步/异步
输入参数	**Timeout**：调用的 Timeout 值；在第 7.4 节中定义 **ApplicationToken**：在第 7.5 节中定义

执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：7.12 节中定义结构

在返回事件中的StatusCode表示调用状态功能，并依赖于此指令航司应用于的虚拟组件，如表29所示：

表 29　　Release 指令 StatusCode 表

Release	虚拟组件类型											
StatusCode	Capture	DataInput	DataOutput	UserInput	UserOutput	Dispenser	Feeder	MediaInput	MeidaOutput	Display	Network	Storage
OK	×	×	×	×	×	×	×	×	×	×	×	×
TIMEOUT	×	×	×	×	×	×	×	×	×	×	×	×
WRONG_STATE	×	×	×	×	×	×	×	×	×	×	×	×
SOFTWARE_ERROR	×	×	×	×	×	×	×	×	×	×	×	×

6.2.7 设置（Setup）指令

表30是Setup指令表。

表 30　　Setup 指令表

描述	设置虚拟组件，设置航司应用程序其配置文件
执行者	所有获取的“外围设备”类的虚拟组件(不包括“本机设备”组件)。
调用者	服务提供商系统管理器(仅在柜机未使用时) 航司应用程序提供系统管理器
访问权限	独占，本地/远程，同步/异步

输入参数	Structure of {
	Timeout: 调用的Timeout值；在第7.4节中定义
	Application Token: 在第7.5节中定义
	Data: 在第 7.10 节中定义结构
	}

执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：7.12 节中定义结构

当航司应用程序变为ACTIVE状态时,平台可确保选择由设置指令中使用的命令设置的航司应用程序的正确上下文。

若数据输入参数为 aeaDataType,则只有以下 AEA 命令才能接受:CT、PT、PC、PS、LT、LC、LS、FT、FC、FS、FA、FR、TT、TC、TA、AV17、ZS18、PV、RI、RC、ES 和 EP。对于LT、LC、LS，标志应采用PCX格式(参见附录 D)。

允许使用不带参数的 BT 命令。使用BT发送的任何参数(尝试设置bin)都将被忽略。其他任何命令会一定在 RC_UNAUTHORIZED有结果。有关在 CUSS 中使用 AEA 标准的更多信息，请参阅附录D。标签打印机应支持 BTT 请求。

当航司应用程序变为活动状态时,平台可确保使用设置指令命令,选择正确的设置航司应用程序的上下文。

返回事件中的StatusCode取决于此指令航司应用于的虚拟组件，如表31所示:

表 31　　Setup 指令 StatusCode 表

Setup	虚拟组件类型								
StatusCode	Capture	DataInput	DataOutput	UserInput	UserOutput	Dispenser	Feeder	MediaInput	MeidaOutput
OK	×	×	×	×	×	×	×	×	×
TIMEOUT	×	×	×	×	×	×	×	×	×
WRONG_STATE	×	×	×	×	×	×	×	×	×
CANCELLED	×	×	×	×	×	×	×	×	×
SOFTWARE_ERROR	×	×	×	×	×	×	×	×	×
OUT_OF_SEQUENCE				×	×			×	×
FORMAT_ERROR	×	×	×	×	×	×	×	×	×
LENGTH_ERROR		×	×	×	×			×	×
DATA_MISSING		×	×	×	×			×	×
CONSUMABLES									×
HARDWARE_ERROR	×	×	×	×	×	×	×	×	×
CRITICAL_SOFTWARE_ERROR	×	×	×	×	×	×	×	×	×
NOT_REACHABLE	×	×	×	×	×	×	×	×	×
NOT_RESPONDING	×	×	×	×	×	×	×	×	×
THRESHOLD_ERROR	×	×	×	×	×	×	×	×	×

6.2.8 测试（Test）指令

表32是Test指令表。

表 32　　Test 指令表

描述	尽可能深入地测试虚拟组件和实际组件。若组件是物理设备，则应访问设备驱动程序，但不应执行物理设备。
执行者	所有获取的“外围设备”类的虚拟组件(不包括“本机设备”组件)。
调用者	服务提供商系统管理器(仅在柜机未使用时)
访问权限	独占，本地/远程，同步/异步
输入参数	**Timeout**：调用的 Timeout 值；在第 7.4 节中定义 **ApplicationToken**：在第 7.5 节中定义
执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：7.12 节中定义结构

返回事件中的StatusCode表明设备状态，并取决于此指令航司应用的虚拟组件，如表33所示:

表 33　　Test 指令 StatusCode 表

Test	虚拟组件类型								
StatusCode	Capture	DataInput	DataOutput	UserInput	UserOutput	Dispenser	Feeder	MediaInput	MeidaOutput
OK	×	×	×	×	×	×	×	×	×
TIMEOUT	×	×	×	×	×	×	×	×	×
WRONG_STATE	×	×	×	×	×	×	×	×	×
CANCELLED	×	×	×	×	×	×	×	×	×
SOFTWARE_ERROR	×	×	×	×	×	×	×	×	×
OUT_OF_SEQUENCE	×			×	×	×		×	×
MEDIA_JAMMED	×					×	×	×	×
MEDIA_MISPLACED						×	×	×	×
MEDIA_PRESENT						×		×	×
MEDIA_ABSENT	×					×		×	×
MEDIA_HIGH	×					×			
MEDIA_FULL	×					×			
MEDIA_LOW							×		
MEDIA_EMPTY							×		
MEDIA_INCOMPLETELY_INSERTED								×	
DATA_PRESENT		×		×				×	

CONSUMABLES									×
HARDWARE_ERROR	×	×	×	×	×	×	×	×	×
CRITICAL_SOFTWARE_ERROR	×	×	×	×	×	×	×	×	×
NOT_REACHABLE	×	×	×	×	×	×	×	×	×
NOT_RESPONDING	×	×	×	×	×	×	×	×	×
THRESHOLD_ERROR	×	×	×	×	×	×	×	×	×
THRESHOLD_USAGE	×	×	×	×	×	×	×	×	×
CONFIGURATION_ERROR		×	×	×	×			×	×

6.2.9 数据指令集

6.2.9.1 总则

下面列出的两个数据指令集，即接收和发送，是与数据处理相关联的。 它们都是通过航司应用程序可用的同步和异步接口调用实现的。

航司应用程序在指令执行产生的事件中接收的所有消息必须与发出指令时航司应用程序使用的消息的类型(格式)相同。

6.2.9.2 接收（Receive）指令

表34是Receive指令表。

表 34　　Receive 指令表

描述	使来自虚拟组件的数据可供航司应用程序使用。航司应用程序必须调用此指令才能从虚拟组件获取未经请求的数据。
执行者	所有获取的“输入”类的虚拟组件(不包括“本地设备”组件)。若组件也属于“用户”类,则必须启用它。
调用者	处于 ACTIVE 状态的航司应用程序 服务提供商系统管理器
访问权限	独占，本地/远程，同步/异步
输入参数	**Timeout**：调用的 Timeout 值；在第 7.4 节中定义 **ApplicationToken**：在第 7.5 节中定义
执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：7.12 节中定义结构

假设航司应用程序在之前启用了设备组件，平台有责任确保设备组件准备好接收数据。

航司应用该平台将不包含任何特定于设备的分隔符(如轨道分隔符或哨兵字符)，填充/错误字符，并且仅返回标准数据。若物理读取器替换字符(例如,对于不可读的 OCR 字符位置)，则数据记录状态将指示读取错误，但返回尽可能多的信息。

若正在读取的媒体具有逻辑多轨排列，则每个轨道将作为单独的 msgData 数据”轨道”返回。例如，2或3轨道磁卡、双轨MRZ数据标准护照、3轨国民身份证OCR数据等(任何有效的多轨文档数据都可以以这种方式接收。CUSS 不仅限于某些类型的文档)。

航司应用程序开发人员应注意，任何媒体输入组件(护照读取器、读卡器等)都可能链接到出纸口组件。若是这种情况，那么CUSS航司应用程序必须在链接的出纸口组件上调用Offer()，才能将文档返回给用户。若航司应用程序不处理此情况，则平台可能会捕获文档(卡或护照等)，并且不会返回给用户。

例如在电动读卡器中,若航司应用程序通过Receive() 获取卡数据,并且不调用Offor()来弹出卡，则后续调用Receive()将不会再次返回数据(DATA_MISSING)。仅当卡弹出,并且客户再次插入卡时，才会再次调用Receive()从新卡中提供数据。

例如在扫卡式读卡器中，航司应用程序调用Enable(), 然后Receive()具有一个长的超时值。该Receive()超时或返回带有卡数据的DATA_PRESENT事件。若航司应用程序再次调用Receive()，因为设备仍然处于启用状态,则该调用将阻塞,直到刷掉新卡。

例如航司应用程序调用Enable()，然后在其事件监听器上等待DATA_PRESENT，然后调用Receive()来获取数据。另一个Receive()的调用不会再次返回数据(若没有调用Offor()，它将为电动读取器提供DATA_MISSING；或者若另一个卡未在指定的超时值内刷卡或插入，则为”数据”提供DATA_MISSING。

航司应用若读卡器能够检测同一文档或项目上的多个条形码(例如，袋子上的多个袋标签或条形码)，则某些设备(如条形码扫描器)也可能返回多个”轨道”。若航司应用程序希望支持这种类型的设备，则可能需要其他逻辑来检测和使用平台提供的其他任何跟踪数据。

若柜机组件在启用时读取多个媒体，而航司应用程序不调用Receive()来响应每次读取，则平台必须放弃除了最新的媒体值外的所有媒体值,并且仅在航司应用程序最终调用Receive()时提供此最新接收的数据。

返回事件中的StatusCode表明功能调用状态，并取决于此指令航司应用的虚拟组件，如表35所示：

表 35　　Receive 指令 StatusCode 表

Receive	虚拟组件类型		
StatusCode	DataInput	UserInput	MediaInpu
OK	×	×	×
TIMEOUT	×	×	×
WRONG_STATE	×	×	×
CANCELLED	×	×	×
SOFTWARE_ERROR	×	×	×
OUT_OF_SEQUENCE	×	×	×
FORMAT_ERROR	×	×	×
LENGTH_ERROR	×	×	×

DATA_MISSING	×	×	×
HARDWARE_ERROR	×	×	×
CRITICAL_SOFTWARE_ERROR	×	×	×
NOT_REACHABLE	×	×	×
NOT_RESPONDING	×	×	×
THRESHOLD_ERROR	×	×	×
THRESHOLD_USAGE	×	×	×
CONFIGURATION_ERROR	×	×	×

6.2.9.3 发送（Send）指令

表36是Send指令表。

表 36　　Send 指令表

描述	航司应用程序发送数据到虚拟组件。
执行者	所有获取的“输入”类的虚拟组件(不包括“本地设备”组件)。若组件也属于“用户”类,则必须在之前启用它。
调用者	处于 ACTIVE 状态的航司应用程序 服务提供商系统管理器
访问权限	独占，本地/远程，同步/异步

输入参数	Structure of {
	Timeout: 调用的Timeout值；在第7.4节中定义
	Application Token: 在第 7.5 节中定义
	Data: 在第7.10节中定义结构
	}

执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：7.12 节中定义结构

返回事件(若有)中的数据流应为输入数据参数的相同类型。例如,若航司应用程序发送了AEA消息,则应在返回事件的数据字段中返回AEA消息。

若输入数据参数为 aeaDataType:CP、CI、TK、TI、TR，则仅接受AEA命令。任何其他命令一定会导致RC_UNAUTHORIZED。

袋标签打印机也应该支持BTP请求。

其他AEA命令(例如操作插入、弹出、托管等)由特定的CUSS指令(如SetIOMode、Offer、Retain等)实现。

VSR(Void Stacker Ribbon indicato)字段在ATB响应中是必需的。

返回事件中的StatusCode表明功能调用状态，并取决于此指令航司应用的虚拟组件，如表37所示:

当组件链接到MediaOutput 组件阻止Send()完成时,请参阅第4.5.3节了解预期行为。

表 37　　Send 指令 StatusCode 表

Send	虚拟组件		
StatusCode	DataOutput	UserOutput	MediaOutput
OK	×	×	×
TIMEOUT	×	×	×
WRONG_STATE	×	×	×
CANCELLED	×	×	×
SOFTWARE_ERROR	×	×	×
OUT_OF_SEQUENCE		×	×
FORMAT_ERROR	×	×	×
LENGTH_ERROR	×	×	×
DATA_PRESENT	×	×	×
CONSUMABLES			×
HARDWARE_ERROR	×	×	×
CRITICAL_SOFTWARE_ERROR	×	×	×
NOT_REACHABLE	×	×	×
NOT_RESPONDING	×	×	×
THRESHOLD_ERROR	×	×	×
THRESHOLD_USAGE	×	×	×
CONFIGURATION_ERROR	×	×	×

6.2.10 文档指令集

6.2.10.1 总则

以下列出的两个文档指令允许文档操作，即提供和保留。它们都通过航司应用程序可用的同步和异步接口调用实现。

6.2.10.2 提供（0ffer）指令

表38是0ffer指令表。

表 38　　0ffer 指令表

描述	将文档从虚拟组件提供给用户或其他组件。
执行者	所有获取的"feeder"类，"dispenser"类的虚拟组件。
调用者	处于 ACTIVE 状态的航司应用程序 服务提供商系统管理器
访问权限	独占，本地/远程，同步/异步
输入参数	**Timeout**：调用的 Timeout 值；在第 7.4 节中定义 **ApplicationToken**：在第 7.5 节中定义
执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：7.12 节中定义结构

仅当手动进纸口或实体出纸口时,才需要提供指令。一些示例包括：MediaOutput(例如卡读写器)需要显式表单源或暂存（ESCROW）设备。

返回事件中的StatusCode中表示设备状态,并依赖于此指令航司应用于的虚拟组件,如表39所示:

表 39　　Offer 指令 StatusCode 表

Offer	虚拟组件	
StatusCode	Dispenser	Feeder
OK	×	×
TIMEOUT	×	×
WRONG_STATE	×	×
CANCELLED	×	×
SOFTWARE_ERROR	×	×
OUT_OF_SEQUENCE	×	
MEDIA_JAMMED	×	×
MEDIA_MISPLACED	×	×
MEDIA_PRESENT	×	
MEDIA_ABSENT	×	
MEDIA_HIGH	×	
MEDIA_FULL	×	
MEDIA_LOW		×
MEDIA_EMPTY		×

HARDWARE_ERROR	×	×
CRITICAL_SOFTWARE_ERROR	×	×
NOT_REACHABLE	×	×
NOT_RESPONDING	×	×
THRESHOLD_ERROR	×	×
THRESHOLD_USAGE	×	×
CONFIGURATION_ERROR	×	×

若出纸口组件是实体的，则需要Offer()指令使文档可供用户使用(例如从电动读卡器弹出卡或打开暂存器的门（escrow door））。

若出纸口组件是虚拟的,即使没有调用Offer()，用户也能立即使用文档。

若出纸口组件(实体或虚拟)可以检测用户何时删除文档，则Offer()指令是一个阻塞调用，仅在获取文档(或请求超时)后，在正常条件下返回。例如使用文档传感器进行纸张输出拍摄的情况,用户可立即使用，虚拟出纸口可以阻塞，但可以检测文档何时存在。

若出纸口不是拥塞中(如上文定义)，则Offer()和Query()指令应返回StatusCode码OK，表明没有其他错误条件存在。

若出纸口是虚拟的，并且没有传感器，那么当航司应用程序异步调用Offer() 时，若不存在其他错误条件，CUSS平台必须异步响应StatusCode码OK，以便航司应用程序可以确定没有物理传感器存在于部件中。

6.2.10.3 保留（Retain）指令

表40是Retain指令表。

表 40　　Retain 指令表

描述	从与安全箱关联的组件中获取文档
执行者	所有获取的“Capture”类的虚拟组件。
调用者	处于 ACTIVE 状态的航司应用程序 服务提供商系统管理器
访问权限	独占，本地/远程，同步/异步
输入参数	**Timeout**：调用的 Timeout 值；在第 7.4 节中定义 **ApplicationToken**：在第 7.5 节中定义
执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：7.12 中定义结构

返回事件中的StatusCode中表示设备状态,并依赖于此指令航司应用于的虚拟组件,如表41所示:

表 41　　Retain 指令 StatusCode 表

Retain	虚拟组件

StatusCode	Capture
OK	×
TIMEOUT	×
WRONG_STATE	×
CANCELLED	×
SOFTWARE_ERROR	×
OUT_OF_SEQUENCE	×
MEDIA_JAMMED	×
MEDIA_ABSENT	×
MEDIA_HIGH	×
MEDIA_FULL	×
MEDIA_LOW	×
MEDIA_EMPTY	×
HARDWARE_ERROR	×
CRITICAL_SOFTWARE_ERROR	×
NOT_REACHABLE	×
NOT_RESPONDING	×
THRESHOLD_ERROR	×
THRESHOLD_USAGE	×

6.2.11 事件指令集——取消（Cancel）指令

表42是Cancel指令表。

表 42　　Cancel 指令表

描述	取消在此指令使用时与组件相关的所有挂起(以前在异步模式下调用)指令。
执行者	所有获取的“外围设备”类的虚拟组件。
调用者	处于 ACTIVE 状态的航司应用程序 服务提供商系统管理器
访问权限	独占，本地/远程，同步/异步

输入参数	**Timeout**：调用的 Timeout 值；在第 7.3 节中定义 **ApplicationToken**：在第 7.4 节中定义
执行结果	**FunctionReturnCode**：附录 A.1 节中定义
数据返回	**Event**：在 7.11 节中定义结构

返回事件中的StatusCode中表示功能调用状态,并依赖于此指令航司应用于的虚拟组件,如表43所示:

表 43　　Cancel 指令 StatusCode 表

Cancel	虚拟组件类型								
StatusCode	Capture	DataInput	DataOutput	UserInput	UserOutput	Dispenser	Feeder	MediaInput	MeidaOutput
OK	×	×	×	×	×	×	×	×	×
TIMEOUT	×	×	×	×	×	×	×	×	×
WRONG_STATE	×	×	×	×	×	×	×	×	×
SOFTWARE_ERROR	×	×	×	×	×	×	×	×	×

6.2.12 出纸口组件纸将满或纸满

它说明了出纸口组件的行为，这些组件对在柜机用户检索之前可以持有的文档数量有有限和实际的限制。

在CUSS中，这种情况通常适用于打印机设备,对于具有剪切/保持功能（cut/hold capability）的行李条打印机很常见。

当通过Send()指令发出打印命令时,若它能够检测到文档是否存在(通过文档传感器等),链接的出纸口组件将返回MEDIA_PRESENT、MEDIA_HIGH或MEDIA_FULL的StatusCode，并且若无法检测文档是否存在，则确定检测文档。

若平台知道无法打印更多文档(例如由于物理打印机或stacker/escrow限制)，则必须返回MEDIA_FULL。CUSS航司应用程序应监视出纸口的StatusCode MEDIA_HIGH或MEDIA_FULL，并在达到此状态时(以及当用户完成打印所有文档(若保留))时,向用户提供出纸口中的文档。

StatusCodeMEDIA_FULL 可能是软错误的还是硬错误，具体取决于虚拟组件。已满的出纸口组件的StatusCode为MEDIA_FULL 将作为软错误。该组件将保持可用，以便向用户提供媒体。这个事件是一个私人事件。

若航司应用程序在链接的出纸口已满时尝试打印更多文档,则该请求将：

——使用 HARDWARE_ERROR 失败。

——弹出/删除现有文档并打印新文档

具有简单打印机设备的CUSS柜机可能只能堆叠一个文档。在这种情况下，平台将在每次打印请求后返回MEDIA_FULL，然后航司应用程序必须等待从出纸口中删除，然后再打印下一个文档。

为了适应这种情况，CUSS航司应用程序必须使用正确的Offer()指令。若删除/弹出当前文档是可以接受的，则航司应用程序可以尝试另一个Send()指令,但是若收到硬件错误,航司应用程序知道并且必须处理打印机出纸口无法删除/弹出文档。

平台应通过Bin特性指示其出纸口组件的任何实际容量限制。然后航司应用程序可以使用此信息来确定可能的连续打印数。

此问题在行李条打印机中最为常见。对于这些设备，航司应用程序方案是:

——航司应用程序发送 Send() 请求为了 BTP 打印

——出纸口组件转到 MEDIA_PRESENT 或 MEDIA_FULL(即使航司应用程序未启用设备)

——航司应用程序调用 Enable()和 Offer()超时，并提示用户删除标记

——航司应用程序等待删除(若需要)

——航司应用程序发出另一个 Send()指令

——若柜机无法丢弃/弹出袋标记，则生成硬件错误

——否则,文档将被丢弃/弹出,并打印下一个文档

——对于多页打印请求，平台可以自动删除/弹出页面，以完成整个打印请求。

6.3 设备组件（Callback）事件

6.3.1 总则

硬件设备发生故障时，DC会生成事件，并且调用监听者提供的Callback方法，将事件传递给航司应用。设备事件的监听者在航司应用调用Acquire方法时当参数传入（Listener引用）。

6.3.2 Callback 指令

表44是Callback指令表。

表 44　　Callback 指令表

描述	向之前注册的监听器发送事件，
执行者	航司应用程序 服务提供商系统管理器 提供航司应用程序系统管理器
调用者	航司应用程序管理类 所有“外围设备”类的虚拟组件
访问权限	本地/远程，同步
输入参数	**Event**：7.11 节中定义结构
数据返回	**None**

6.3.3 CUSS 设备组件 Callback 事件

表45是设备组件生成的所有事件的列表，这些事件或者发送给其所有适用的侦听器，或者在请求事件时返回到其调用方。列表按EventCode排序,该代码表示状态转换或无状态转换情况下的当前状态，并包括所有可能的关联状态代码。

表 45　　CUSS 设备组件 Callback EventCode 表

EventCode		
EventCode	描述	
001	状态转换正确&软件错误= EVENTHANDLING_READY 仅仅用于软件正确	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	003	CANCELLED
	004	SOFTWARE_ERROR
	006	OUT_OF_SEQUENCE
	102	MEDIA_MISPLACED
	103	MEDIA_PRESENT
	104	MEDIA_ABSENT
	105	MEDIA_HIGH
	107	MEDIA_LOW
	109	MEDIA_DAMAGED
	110	MEDIA_INCOMPLETELY_INSERTED
	201	FORMAT_ERROR
	202	LENGTH_ERROR
	203	DATA_MISSING
	205	DATA_PRESENT
002	状态转换重启= UNAVAILABLE_RELEASED_PLATFORM 授权平台组件出于任何原因释放了处于 UNAVAILABLE 状态的组件	
	相关 StatusCode	
	801	CUSS_MANAGER_REQUEST
003	状态转换硬件错误= EVENTHANDLING_UNAVAILABLE 由于组件无法使用的恶劣条件引起的	
	相关 StatusCode	
	101	MEDIA_JAMMED

	102	MEDIA_MISPLACED
	106	MEDIA_FULL
	108	MEDIA_EMPTY
	301	CONSUMABLES
	302	HARDWARE_ERROR
	303	CRITICAL_SOFWARE_ERROR
	304	NOT_REACHABLE
	305	NOT_RESPONDING
	306	THRESHOLD_ERROR
	307	THRESHOLD_USAGE
	308	CONFIGURATION_ERROR
004	状态转换版本= UNAVAILABLE_RELEASED 航司应用程序发布了处于 UNAVAILABLE 状态的组件	
	相关 StatusCode	
	000	OK
	004	SOFTWARE_ERROR
005	状态转换版本=READY_RELEASED_APPLICATION 航司应用程序发布了处于 READY 状态的组件	
	相关 StatusCode	
	000	OK
	004	SOFTWARE_ERROR
006	状态转换重启= READY_RELEASED_PLATFORM 授权平台组件出于任何原因释放了处于 READY 状态的组件	
	相关 StatusCode	
	801	CUSS_MANAGER_REQUEST
007	状态转换接受=RELEASED_READY 航司应用程序已获取正常运行的组件	
	相关 StatusCode	
	000	OK
	004	SOFTWARE_ERROR
	102	MEDIA_MISPLACED

	103	MEDIA_PRESENT
	104	MEDIA_ABSENT
	105	MEDIA_HIGH
	107	MEDIA_LOW
	109	MEDIA_DAMAGED
	110	MEDIA_INCOMPLETELY_INSERTED
008	状态转换接受（硬件错误）= EVENTHANDLING_UNAVAILABLE 航司应用程序获取的组件无法正常工作	
	相关 StatusCode	
	101	MEDIA_JAMMED
	102	MEDIA_MISPLACED
	103	MEDIA_PRESENT
	106	MEDIA_FULL
	107	MEDIA_ EMPTY
	301	CONSUMABLES
	302	HARDWARE_ERROR
	303	CRITICAL_SOFWARE_ERROR
	304	NOT_REACHABLE
	305	NOT_RESPONDING
	306	THRESHOLD_ERROR
	307	THRESHOLD_USAGE
	308	CONFIGURATION_ERROR
201	状态：释放 无状态组件，组件处于释放状态	
	相关 StatusCode	
	000	OK
	001	TIMEOUT
	002	WRONG_STATE
	003	CANCELLED
	004	SOFTWARE_ERROR
202	状态：无法使用	

<table>
<tr><td></td><td colspan="2">无状态组件，组件处于无法使用状态</td></tr>
<tr><td></td><td colspan="2">相关 StatusCode</td></tr>
<tr><td></td><td>001</td><td>TIMEOUT</td></tr>
<tr><td></td><td>002</td><td>WRONG_STATE</td></tr>
<tr><td></td><td>003</td><td>CANCELLED</td></tr>
<tr><td></td><td>004</td><td>SOFTWARE_ERROR</td></tr>
<tr><td></td><td>101</td><td>MEDIA_ JAMMED</td></tr>
<tr><td></td><td>102</td><td>MEDIA_MISPLACED</td></tr>
<tr><td></td><td>103</td><td>MEDIA_PRESENT</td></tr>
<tr><td></td><td>106</td><td>MEDIA_FULL</td></tr>
<tr><td></td><td>107</td><td>MEDIA_ EMPTY</td></tr>
<tr><td></td><td>301</td><td>CONSUMABLES</td></tr>
<tr><td></td><td>302</td><td>HARDWARE_ERROR</td></tr>
<tr><td></td><td>303</td><td>CRITICAL_SOFWARE_ERROR</td></tr>
<tr><td></td><td>304</td><td>NOT_REACHABLE</td></tr>
<tr><td></td><td>305</td><td>NOT_RESPONDING</td></tr>
<tr><td></td><td>306</td><td>THRESHOLD_ERROR</td></tr>
<tr><td></td><td>307</td><td>THRESHOLD_USAGE</td></tr>
<tr><td></td><td>308</td><td>CONFIGURATION_ERROR</td></tr>
<tr><td>203</td><td colspan="2">状态：READY
无状态组件，组件处于 READY 状态</td></tr>
<tr><td></td><td colspan="2">相关 StatusCode</td></tr>
<tr><td></td><td>000</td><td>OK</td></tr>
<tr><td></td><td>001</td><td>TIMEOUT</td></tr>
<tr><td></td><td>002</td><td>WRONG_STATE</td></tr>
<tr><td></td><td>003</td><td>CANCELLED</td></tr>
<tr><td></td><td>004</td><td>SOFTWARE_ERROR</td></tr>
<tr><td></td><td>006</td><td>OUT_OF_SEQUENCE</td></tr>
<tr><td></td><td>102</td><td>MEDIA_MISPLACED</td></tr>
<tr><td></td><td>103</td><td>MEDIA_PRESENT</td></tr>
<tr><td></td><td>104</td><td>MEDIA_ABSENT</td></tr>
</table>

	107	MEDIA_ EMPTY
	109	MEDIA_DAMAGED
	110	MEDIA_INCOMPLETELY_INSERTED
	201	FORMAT_ERROR
	202	LENGTH_ERROR
	203	DATA_MISSING
	205	DATA_PRESENT
210	状态：BUSY 组件处于 BUSY 瞬态	
	相关 StatusCode	
	000	OK
	102	MEDIA_MISPLACED
	103	MEDIA_PRESENT
	104	MEDIA_ABSENT
	107	MEDIA_ EMPTY
	109	MEDIA_DAMAGED
	110	MEDIA_INCOMPLETELY_INSERTED
	201	FORMAT_ERROR
	202	LENGTH_ERROR

7 数据结构定义

7.1 总则

IATA官方提供以下IDL文件：

characteristics.IDL

codes.IDL

comps.IDL

types.IDL

这些IDL定义了CUSS平台和航司应用之间交互的接口规范。接口规范中数据结构的定义，在本章中描述。

7.2 引用（Reference）

它是用于航司应用程序和组件引用的字符串的别名类型（AliasType）。

7.3 名称（Name）

它是用于名称定义的字符串的别名类型（AliasType）。

7.4 超时（Timeout）

Timeout：数值<0：拒绝超时值，异步调用

=0：没有超时，同步(blocking forever)调用

>0：超时值；异步调用

(超时值以毫秒为单位表示。)

7.5 航司应用程序令牌（ApplicationToken）

ApplicationToken：Reference

(CUSS 航司应用程序管理器将此令牌分配给航司应用程序。它被用来作为航司应用程序可用的指令访问控制机制)

7.6 相关性（Correlation）

Correlation：any任意

(这是航司应用程序在发出指令(注册事件或获取)以注册其侦听器时设置的,它允许事件与特定指令颁发关联。它仅用于比较用户定义的专用标识。)

CUSS航司应用程序可以将他们在此参数中选择的任何数据插入到acquire() 和registerEvent()函数调用。但是,出于性能原因,不应使用大量数据对象。航司应用程序通常使用字符串或长整型对象作为相关参数。

平台不会对此值执行数据/类型检查,因为它是特定于航司应用程序的,并且平台将在每个事件广播中将相同的相关数据返回到响应这些调用而创建的事件侦听器。然后,航司应用程序可以根据需要分析其相关值,作为航司应用程序事件处理的一部分。

7.7 虚拟组件参数（VcompReference）

Vcomp Reference：Reference

(这是对虚拟组件的 CORBA 引用(IOR).)

7.8 柜机位置（KioskLocation）

表46是KioskLocation数据结构表。

表 46　　KioskLocation 数据结构表

字段名	类型	说明
airportCode	String	机场三字码
terminal	String	航站楼
area	String	区域
address	String	地址

7.9 柜机的 GPS 坐标（KioskGPSCoordinates）

表47是KioskGPSCoordinates数据结构表。

表 47　　KioskGPSCoordinates 数据结构表

经度		
字段名	类型	说明
Degree	int	度，取值{0-179}
Minute	int	分，取值{0-59}
Second	int	秒，取值{0-59}
Hundreds	int	取值{0-99}
纬度		
字段名	类型	说明
Degree	int	度，取值{0-89}
Minute	int	分，取值{0-59}
Second	int	秒，取值{0-59}
Hundreds	int	取值{0-99}
海拔		
字段名	类型	说明
number	int	从海平面以米为单位

7.10 数据（Data）

表48是Data数据结构表。

表 48　　Data 数据结构表

字段名	类型	说明
AEA	int	字符链（AEA 格式的消息）
CLOCK		字符链（格式为“yyyymmddhhmmss”）
SVG	int	字符链（格式为“yyyymmddhhmmss”）
SWITCH		设置{关、开、开、关、是、否、未知}（传感器设备使用）
umber of data records		编号
Record 3		数据状态：{确定，损坏，不完整，零长度}的集合”的结构的数据记录数
Message	byte	字符链

7.11 柜机航司应用身份证明（KioskApplicationID）

表49是KioskApplicationID数据结构表。

表 49　　KioskApplicationID 数据结构表

字段名	类型	说明
companyCode	String	航司应用的航司两字代码
applicationName	String	航司应用软件名
vendorCode	String	平台提供商公司名称
kioskName	String	平台软件名

当用作调用平台的输入参数时(例如,对于级别(0 调用),航司应用程序 ID 值"公司代码"和航司应用程序名称用于唯一标识发出请求的 CUSS 航司应用程序。忽略供应商代码和展台名称参数(即航司应用程序无法请求特定的展台 ID)。

该平台将用柜机提供商选择的任何值填充输出环境级别 akID 和位置字段,前提是这些值符合 CUSS 标准。

航司应用程序供应商不能假定特定的柜机名称在其整个网络中是唯一的。只有供应商代码、柜机位置和柜机名称的组合才能保证是唯一的。

航司应用程序供应商不能要求 CUSS 柜机提供特定的柜机名称。若航司应用程序需要特定的语法或值的 kioskName/位置值来操作,则航司应用程序必须包括内部映射/查找表或其他功能,用于将平台提供的 akID/位置值转换为航司应用程序需要。

7.12 事件（Event）

表50是Event数据结构表。

表 50　　Event 数据结构表

字段名	类型	说明
Timestamp	long	事件发起人时间戳
Kiosk-Application-ID		柜机航司应用程序的 ID，若事件源是 CAM，则为适用柜机航司应用程序的 ID）
Kiosk Location	location	信息亭位置
Kiosk GPS Location	gps	信息亭 GPS 坐标
VCompReference		虚拟组件引用来源
Function	string	调用的指令的名称
Event code	int	EventCode：附录 A 中定义的数字（航司应用程序/组件状态转换，若不航司应用转换，则为当前航司应用程序/组件状态）
Status code	int	组件或函数调用状态

Correlation		仅用于比较，由航司应用程序设置
Data		与事件一起传递的数据

无论事件源或原因如何,都使用相同的事件结构。因此,某些事件字段可能并不总是适用的。在这种情况下,字段值可能留空,或者其值被视为不适用。以下是关于不适用值的一些示例：柜机-航司应用程序-身份验证(若事件源是设备虚拟组件)、虚拟组件参数(若事件源是柜机航司应用程序)、函数名称(若事件是未经请求的)、状态代码(若事件源是一个柜机航司应用程序)。

7.13 事件列表选择（EventListSelection）

表51是EventListSelection数据结构表。

表 51　　EventListSelection 数据结构表

字段名	类型	说明
ANY	int	等待任何组件的任何代码，仅用于 waitEvent
ALL	int	适用于所有组件的所有代码
CODE	int	事件代码选择：第 7.14 节定义的结构
TYPE	int	事件类型选择：第 7.15 节中定义的结构
COMPONENT	int	组件选择：结构定义第 7.16 节

7.14 事件代码选择（EventCodeSelection）

表52是EventCodeSelection数据结构表。

表 52　　EventCodeSelection 数据结构表

字段名	类型	说明
ALL	int	将此代码航司应用于所有组件
ANY	int	航司应用于此代码到任何组件,仅用于等待事件指令
COMPONET	int	结构内容
Number of Components		组件数量

7.15 事件类型选择（EventTypeSelection）

表53是EventCodeSelection数据结构表。

表 53　　EventTypeSelection 数据结构表

字段名	类型	说明
COMPONENT	int	结构内容{此列表类型不能与获取指令一起使用}

ALL	int	将此代码航司应用于所有组件
ANY	int	航司应用于此代码到任何组件,仅用于等待事件指令
Number of Components		组件数量

7.16 组件选择（ComponentSelection）

表54是ComponentSelection数据结构表。

表 54　　ComponentSelection 数据结构表

字段名	类型	说明
ALL	int	将此代码航司应用于所有组件
ANY	int	航司应用于此代码到任何组件,仅用于等待事件指令
CODE	int	附录 A 中定义编号
TYPE	int	集合(PRIVATE, PUBLIC, PLATFORM)
CATEGORY		集合(NORMAL, ALERT, ALARM)

7.17 事件分类选择（EventcategorySelection）

表55是EventcategorySelection数据结构表。

表 55　　EventcategorySelection 数据结构表

字段名	类型	说明
ALL	int	将此代码航司应用于所有组件
ANY	int	航司应用于此代码到任何组件,仅用于等待事件指令
Number of Components		数值

8 虚拟组件属性

8.1 公共属性

8.1.1 总则

8.2节到8.14节中描述的各种类型的属性都继承下面这些属性。

8.1.2 存纸盒设置（BinSetting）

表56是BinSetting属性表。

表 56　　BinSetting 属性表

属性	描述
BinSize	描述一个打印机存纸盒可以容纳的打印介质最大数量。 于 MEDIA_FULL 状态。
AlmostFullLevel	显示打印机存纸盒容量的高阀值:MEDIA_HIGH 状态相一致的打印介质数量（例如，出纸口组件），若安装了相应的传感器。
ALmostEmptyLevel	显示打印机存纸盒容量的低阀值：MEDIA_LOW 状态相一致的打印介质数量（例如，进纸口组件），若安装了相应的计数传感器
currentNoOfDocuments	若打印机装有计数传感器，显示打印机存纸盒中当前打印介质的数量。若 AlmostEmptyLevel 还没有达到，这个数值没必要特别准确。另一方面当接近 AlmostEmptyLevel 时，数值应该尽量准确。

平台必须为存在实际容量限制的任何出纸口组件准确设置这些特性。这允许航空公司航司应用程序准确地管理多打印文档作业。

8.1.3 组件字体（ComponenetFont）

表57是ComponenetFont属性表。

表 57　　ComponenetFont 属性表

属性	描述
useStandard	表明使用的条形码标准（code39、code128 或者是 cdoe2of5）
Fonts	列举组件提供的所用字体（和字体属性）。字体属性有： fontName,fontSizes,vectorFont,bold,italic,underlined,strikethrough, reverse,superscript,subscript,colorDepth,spacing 以及 CharacterLength。

8.1.4 输入输出模式（IOMOode）

表58是IOMOode属性表。

表 58　　IOMOode 属性表

属性	描述
mode	组件当前使用的读写模式

表59是setIOMode指令表。

表 59　　setIOMode 指令表

描述	为 ATB 打印机设置输入/输出模式。
执行者	所有需要“媒介输入”或者“媒介输出”属性的虚拟组件
调用者	航空公司航司应用程序(在初始化(INITIALIZE)状态或者激活(ACTIVE)状态)
访问控制	独占，本地/远程，同步的
输入参数	航司应用程序标识：一个有效的活动航司应用程序引用，在 6.1.4 节中定义 输入输出模式：正在使用的输入/输出模式
数据返回	ReturnCode，在附录 A 中定义

setIOMode指令是上下文特定的，当航司应用程序变为激活(ACTIVE)状态，平台确保为航司应用程序选择正确的上下文。

8.1.5 位置（Location）

表60是Location属性表。

表 60　　Location 属性表

属性	描述
Map	柜机组件的位置图片文件的 URL
mapType	图片文件的类型(BMP, GIF, JPEG, PNG, Flash 等)
howTo	使用柜机组件的图像/动画文件的 URL。这是一个完整的 XML 结构，指示组件的设备帮助使用说明
howToType	图片/动画的类型(GIF, JPEG, Flash 等)
componentLocation	组件位置(柜机内部或者柜机外部)

8.1.6 制造商（Manufacturer）

表61是Manufacturer属性表。

表 61　　Manufacturer 属性表

属性	描述
realComponentIdentification	系统管理使用的组件识别标识(例如ATB打印机-BIN1)
downloadableFirmware	描述固件是否可以升级

firmwareVersion	固件/软件版本
manufacturerName	制造商名称
modelNumber	硬件组件的型号
serialNumber	硬件组件的序列号

8.1.7 媒介类型（MediaType）

表62是MediaType属性表。

表 62　　MediaType 属性表

属性	描述
type	包含以下媒介类型之一的属性： MagneticStripe，Chip，Printed（OCR/Barcode/Plain paper），JIS

8.1.8 媒介类型列表（MediaTypeList）

表63是MediaTypeList属性表。

表 63　　MediaTypeList 属性表

属性	描述
mtList	媒介类型列表。（请参阅第 8.1.6 节：媒介类型）

8.2 航司应用程序（Application）属性

继承Manufacturer（详见8.1.6）属性。其他属性详见表64。

表 64　　Application 属性表

属性	描述
identification	航空公司航司应用标识
allContacts	包含所用可用公司联系方式的列表，表项包括：公司名称，联系人姓名，地址（邮编，电话号码，传真，传呼机，电子邮件，远程支持航司应用地址，若适用）以及关于个人或公司的简要说明
firstIPPort	指定此航司应用程序可以使用的第一个 IP 端口范围。
lastIPPort	指定此航司应用程序可以使用的最后一个 IP 端口范围。

8.3 废纸槽（Capture）属性

继承BinSettings（详见8.1.2)和Manufacturer（详见8.1.6）属性。

8.4 数据输入（DataInput）属性

继承Manufacturer（详见8.1.6）属性。其他属性详见表65。

表 65　　DataInput 属性表

属性	描述
timeZone	相对于格林尼治标准时间的时差（适用于时钟组件）
supportedDataTypes	组件支持的数据类型列表（CLOCK，MSG，NIL，WITCH）

8.5 数据输出（DataOutput）属性

继承Manufacturer（详见8.1.6）属性。其他属性详见表66。

表 66　　DataOutput 属性表

属性	描述
supportedDataTypes	组件支持的数据类型列表（MSG，NIL，WITCH）

8.6 出纸口（Dispenser）属性

继承BinSettings（详见8.1.2）、Location（详见8.1.5）和Manufacturer（详见8.1.6）属性。其他属性详见表。67

表 67　　Dispenser 属性表

属性	描述
kind	标明出纸口是实体的还是虚拟的。 若必须使用 offer()指令将打印结果交给用户（例如从自动读卡器中弹出卡或打开暂存装置的门），则出纸口组件是实体的。 若用户可以直接获得打印结果，而且不用使用 offer()，例如使用纸张输出滑槽，则出纸口组件是虚拟的。 两种类型的出纸口都可以决定是否使用传感器来检测出纸口中是否有打印文件。请参阅 6.6.9.1 节 offer()获取更多使用这个属性和出纸口状态指示器的信息，以便正确地向用户提供文档并监控（若可能）文档是否已被取走。

8.7 显示（Display）属性

继承Location（详见8.1.5）和Manufacturer（详见8.1.6）属性。其他属性详见表68。

表 68　　Display 属性表

属性	描述
displayResolution	列出支持的屏幕分辨率： 1024 标识分辨率 1024×768 1280 标识分辨率 1280×1024 1600 标识分辨率 1600×1200. CUSS 平台可以提供至少 1024 像素宽和 768 像素高的任何屏幕分辨率，包括纵向模式显示和宽高比不是 4：3 的显示模式。
currentResolution	当前使用的分辨率。
screenDiagonal	以毫米度量的物理屏幕尺寸。

表69是SetScreenResolution指令表。

表 69　　SetScreenResolution 指令表

描述	设置新的显示分辨率。
执行者	所有需要“显示”类的虚拟组件
调用者	航空公司航司应用程序在激活(ACTIVE)状态
访问权限	独占，本地/远程，同步的
输入参数	Application Token：在 XXX 章节定义 Resolution: 航司应用程序使用的屏幕分辨率
Structure returned	ReturnCode，在附录 A 中定义

若平台没有实现这个功能，则设置屏幕分辨率应返回RC_NOT_SUPPORTED。

8.8 进纸口（Feeder）属性

继承Location（详见8.1.5）和Manufacturer（详见8.1.6）属性。

8.9 媒介输入（MediaInput）属性

继承ComponenetFonts(详见8.1.3)、IOMode（详见8.1.4）、Location（详见8.1.5）、Manufacturer（详见8.1.6）、MediaTypeList（详见8.1.8）属性。其他属性详见表70。

表 70　　MediaInput 属性表

属性	描述

typeOfReader	由此组件处理的阅读器类型（ Motorized、 DIP、 Swipe、 Contactless、 FlatbedScan、 PenScan）
supportedDataTypes	这个组件支持的数据类型列表（AEA，MSG，NIL）
setupDataType	描述此组件支持的数据流类型。
numberOfTracks	读设备支持的最大数据条数（磁卡指磁道，护照指行数等）

8.10 媒介输出（MediaOutput）属性

继承ComponenetFonts(详见8.1.3)、IOMode（详见8.1.4）、Location（详见8.1.5）、Manufacturer（详见8.1.6）、MediaTypeList（详见8.1.8）属性。其他属性详见表71。

表 71　　MediaOutput 属性表

属性	描述
type	包含所用媒介类型的属性 (Ticket,BoardingPass,GeneralPurposeDoc,Baggage Tag,InsertedDoc,Card)
supportedDataTypes	组件支持的数据类型（AEA，MSG，NIL，SVG）
bufferSize	内部数据缓冲区大小
numberOfTracks	写设备支持的最大数据条数（磁卡指磁道，护照指行数）
minDocumentLength	以毫米为单位的最小文件长度
maxDocumentLength	以毫米为单位的最大文件长度
maxPrintSizeX	以毫米为单位的 X 方向最大打印尺寸
maxPrintSizeY	以毫米为单位的 Y 方向最大打印尺寸
mediaTransferType	所用的打印技术规格：DirectThermal、ThermalTransfer、nonApplicable

printOrientation	当前打印方向（纵向或横向）

表72是SetPrintOrientation指令表。

表 72　　SetPrintOrientation 指令表

描述	设置组件的打印方向。
执行者	所有需要“媒介输出”类的虚拟组件
调用者	航空公司航司应用程序在激活(ACTIVE)状态
访问控制	独占，本地/远程，同步的
输入参数	Application Token：有效的活动应用程序引用，在 7.15 节定义 Orientation:打印方向（纵向或横向）
执行结果	ReturnCode，在附录 A 中定义

每次变为ACTIVE状态时，航司应用程序必须检查打印方向（并在必要时进行设置）以确保正确的方向（先前活动的航司应用程序可能已更改方向）。

8.11 网络（Network）属性

继承Manufacturer（详见8.1.6）属性。

8.12 存储（Storage）属性

继承Manufacturer（详见8.1.6）属性。其他属性详见表73。

表 73　　Storage 属性表

属性	描述
Size	指定磁盘上航司应用程序可用的总大小（以MB为单位）。
Path	指定可写/可读位置的完整路径（所有路径规范以分隔符结尾，例如斜杠或反斜杠）。例如，在Windows下，Path将类似于C:\CUSS\APPS\AC\CKC\。

在CUSS中，Path属性将给出包含航司应用程序本地文件的位置的完整路径。若航司应用程序没有本地文件，它将指向平台在上创建的为航司应用程序保留的目录。

CUSS平台必须为每个航司应用程序创建和提供存储组件，并允许航司应用程序使用自己独特的路径属性。存储位置可以是的本地位置，也可以是网络，但该目录对于航司应用程序和特定必须是唯一的（不与其他上的同一航司应用程序共享）。

8.13 用户输入（UserInput）属性

继承Location（详见8.1.5）和Manufacturer（详见8.1.6）属性。

8.14 用户输出（UserOutput）属性

继承Location（详见8.1.5）和Manufacturer（详见8.1.6）属性。

9 实体设备编程规范

9.1 总则

本部分阐明在CUSS实现中使用设备时的实体设备行为。虚拟组件链接和属性。

为了简化序列图，假设所航司应用程序通过调用Acquire()已经获取了所有组件，并且完成使用或监听虚拟组件，则将调用Release()释放。

序列图并不是为了显示每种可能的情况。它们是典型用例的例证，以及一些可能难以概念化的情况。理论上，错误场景（例如设备无响应）可以在任何时间发生，与正在执行的操作无关。

9.2 ATB 打印机（AEA 打印设备）

9.2.1 设备描述

普通的ATB打印机是一种简单的AEA打印设备，没有磁编码功能，也没有用于读取/重新验证登机牌的插槽。它没有临时保存的装置，登机牌打印后会自动将登机牌递给打印者。

9.2.2 虚拟组件链接图

图5是ATB打印机链接图。

图 5 ATB 打印机链接图

9.2.3 虚拟组件链接的描述

ATB打印机组件链接进纸口和虚拟出纸口。

9.2.4 属性

表74是ATB打印机**MediaOutput**属性表。表75是ATB打印机**Dispenser**属性表。

表 74 ATB 打印机 MediaOutput 属性表

属性	Dispenser 值
MediaType.MediaTypeListDef	MediaType.MediaTypeDef.Printed
MediaOutput.MediaType.type	MediaOutput.MediaType.BoardingPass

	MediaOutput.MediaType.Ticket MediaOutput.MediaType.GeneralPurposeDoc
MediaOutput.supportedDataTypes	DataType.AEA

表 75　　ATB 打印机 Dispenser 属性表

属性	值
Dispenser.kind	Dispenser.DispenserType.virtual_ 因为不需要 offer()，所以出纸口是虚拟的。这是因为打印完成后，最终用户就可以使用该媒介。 若调用 offer()，平台将生成 StatusCode 为 OK 的响应，以便航司应用程序知道平台在出纸口组件中没有物理传感器。 若对虚拟出纸口调用了 offer()，则若平台支持检测 tickets 删除的功能，则 offer()必须阻塞，直到登机牌被删除。

9.2.5 AEA 打印机的典型序列图

图6是ATB打印机的典型序列图。

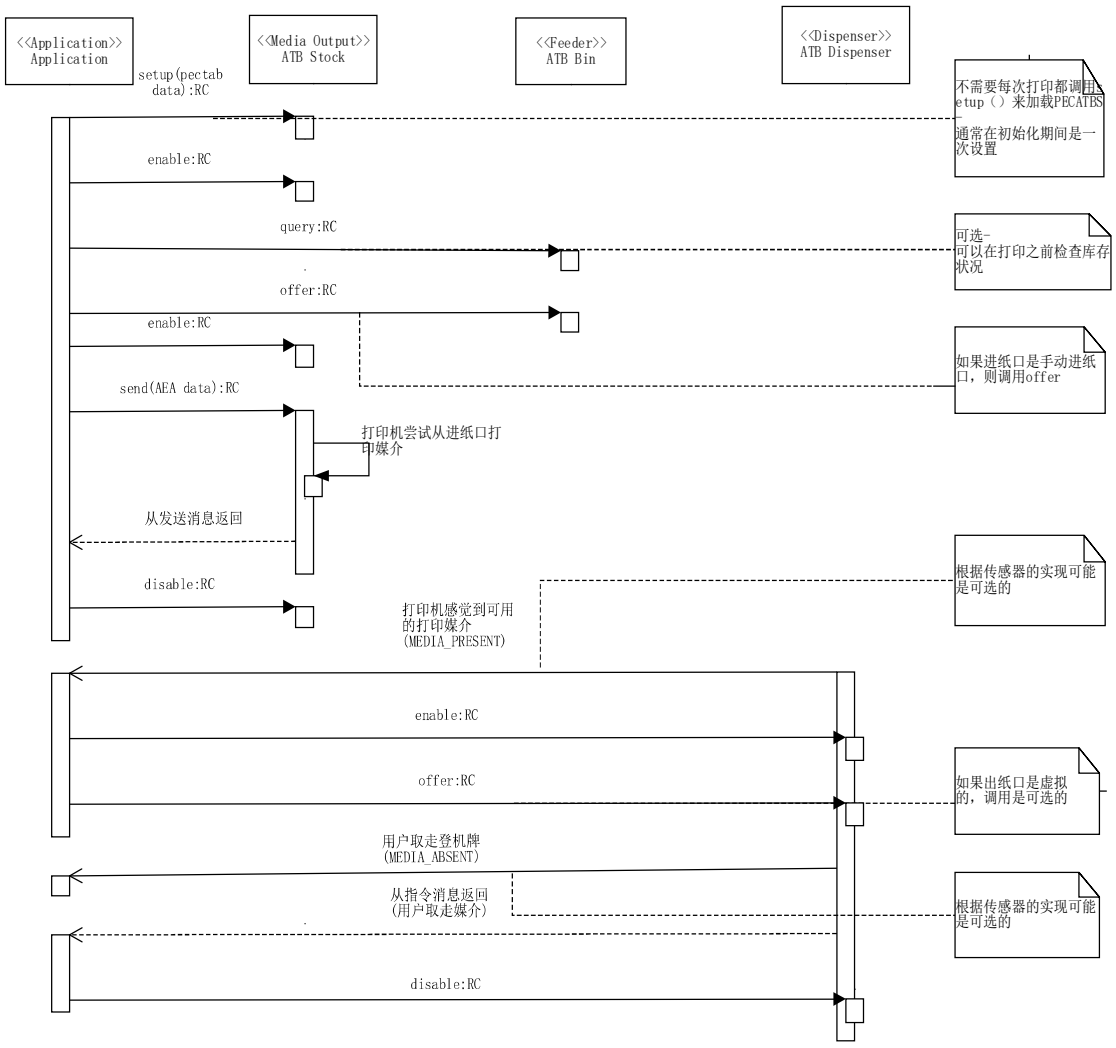

图 6 ATB 打印机的典型序列图

9.3 行李条打印机

9.3.1 设备描述

行李条打印机根据BTP AEA规范打印行李条。在这个简单的行李条打印机示例中，行李条在打印后可以提供给用户使用。

CUSS柜机的行李条打印机可能具备检测用户何时取出打印完成的行李条的功能。若确实存在此功能，则CUSS平台提供商可以（但非强制）通过实现“实体”出纸口组件向CUSS航司应用程序提供此功能。

9.3.2 虚拟组件链接图

图7是行李条打印机链接图。

行李条打印机

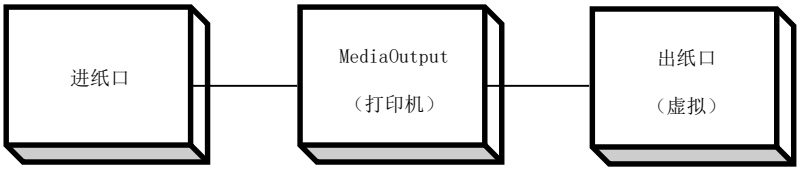

图 7 行李条打印机链接图

9.3.3 虚拟组件链接的描述

行李条打印机组件链接进纸口和出纸口。若平台检测用户是否使用行李条，以及何时使用行李条的，或者平台有一个物理的行李条输出箱，那么出纸口就是实体的，否则是虚拟的。

9.3.4 属性

表76是行李条打印机**MediaOutput**属性表，表77是行李条打印机**Dispenser**属性表。

表 76 行李条打印机 **MediaOutput** 属性表

属性	值
MediaType.MediaTypeListDef	MediaType.MediaTypeDef.Printed
MediaOutput.MediaType.type	MediaOutput.MediaType.BaggageTag
MediaOutput.supportedDataTypes	DataType.AEA

表 77 行李条打印机 **Dispenser** 属性表

属性	值
Dispenser.kind	Dispenser.DispenserType.virtual_ 若出纸口是虚拟的，则平台不会检测到打印机所打印的行李条何时或是否被取走。在这种情况下，不需要 offer()。这是因为打印后，最终用户就可以使用该设备。 若对虚拟出纸口调用 offer()，则 offer()可能会阻塞，直至移除行李条（若支持检测行李条移除功能），即使出纸口类型为 virtual。 Dispenser.DispenserType.real_ 若出纸口是实体的，则平台可以检测或控制何时从行李条输出位置取出打印的行李条。在这种情况下，需要使用 offer()才能使文档对最终用户可用。 通常，行李条打印机的出纸口可能只能容纳最多一个行李条。因此，对出纸口组件的查询或异步事件可能显示 MEDIA_FULL 而不是 MEDIA_PRESENT 航司应用程序可以使用 Dispenser.BinSize 属性以编程方式确定出纸口箱子的最大容量。此外，Dispenser.currentNoOfDocuments 可以显示出纸口中的行李条数量。缺少传感器功能的平台可能不支持 Dispenser.currentNoOfDocuments。

9.3.5 行李条打印机的典型序列图

图8是行李条打印机的典型序列图，与ATB打印机序列图相同。在某些情况下，出纸口虚拟组件的容量为1，这是为确保在打印下一个行李条时，需要取走上一个打印好的行李条。

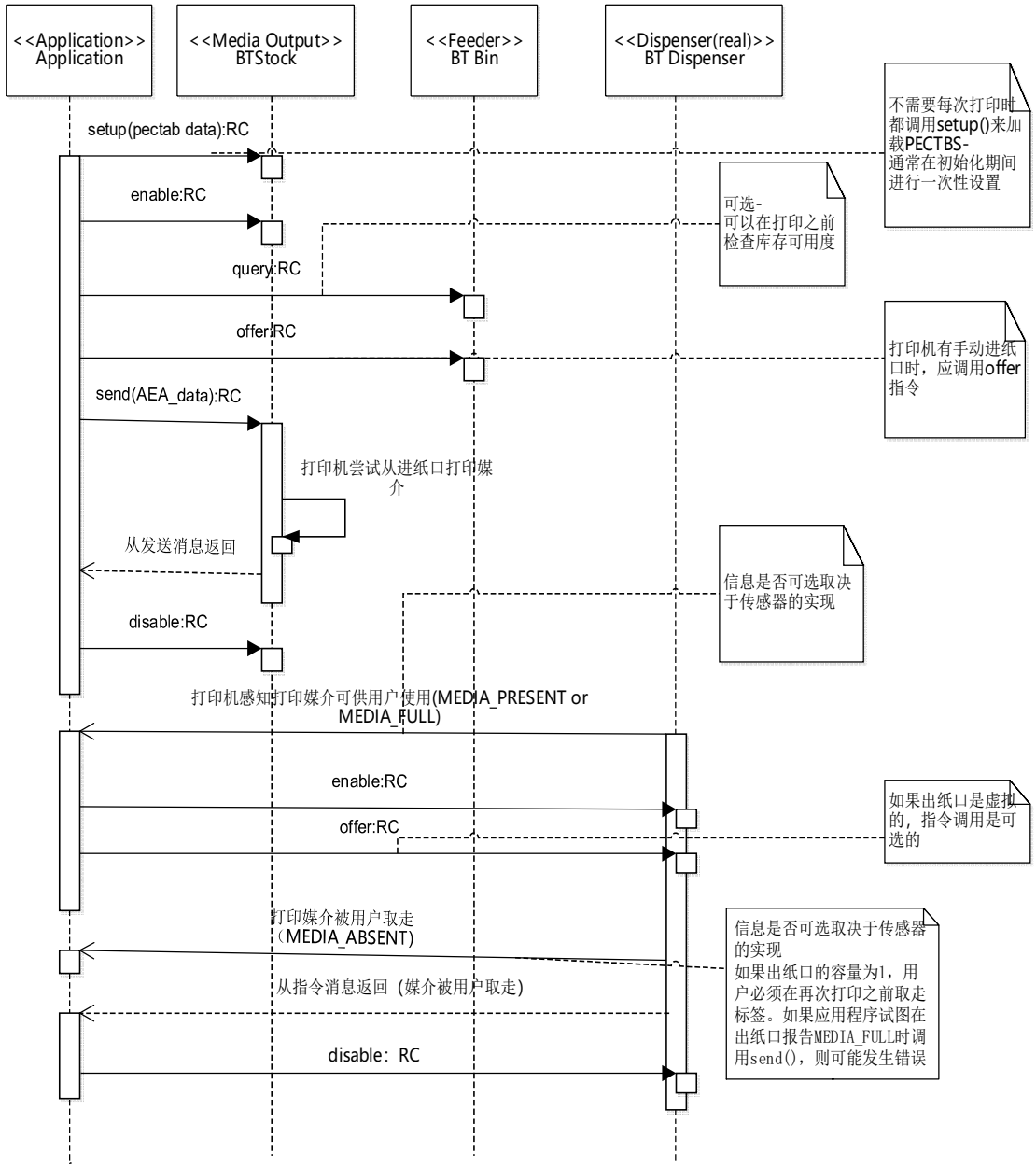

图 8　行李条打印机的典型序列图

9.4 插卡式/刷卡式磁卡读卡器

9.4.1 设备描述

插卡式/刷卡式磁卡读卡器接收并读取ISO磁编码卡。有一个MediaInput组件表示插卡式/刷卡式磁卡读卡器。

9.4.2 虚拟组件链接图

图9是插卡式/刷卡式磁卡读卡器链接图。

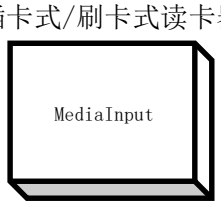

图 9 插卡式/刷卡式磁卡读卡器链接图

9.4.3 虚拟组件链接的描述

一个简单的MediaInput虚拟组件，配置为类型插卡式/刷卡式读卡器。

9.4.4 属性

表78是插卡式/刷卡式磁卡读卡器MediaInput属性表。

表 78 插卡式/刷卡式磁卡读卡器 MediaInput 属性表

属性	值
MediaType.MediaTypeListDef	MediaType.MediaTypeDef.MagneticStripe
MediaInput.typeOfReader	MediaInput.ReaderType.DIP MediaInput.ReaderType.Swipe
MediaOutput.supportedDataTypes	DataType.MSG
MediaInput.numberOfTracks	<int>，取决于硬件能够读取的实际轨道数。通常对于 ISO 阅读器的此数字为 2 或 3
Manufacturer.FirmwareVersion	<string>-显示支持的数据类型，例如： DS_TYPES_FOID_ISO,DS_TYPES_PAYMENT_ISO, DS_TYPES_DISCRETIONARY_ISO.

9.4.4 支持的扩展数据类型

默认情况下，当客户插入支付卡时，CUSS读卡器接口将仅向航司应用程序提供被删节的追踪数据。若航司应用程序需要完全合法地访问支付卡信息，必须调用Setup()来配置访问权限。

有关扩展数据类型和支付卡数据缩减的更多信息，请参阅图10和图11。

9.4.5 插卡式/刷卡式磁卡读卡器的典型序列图

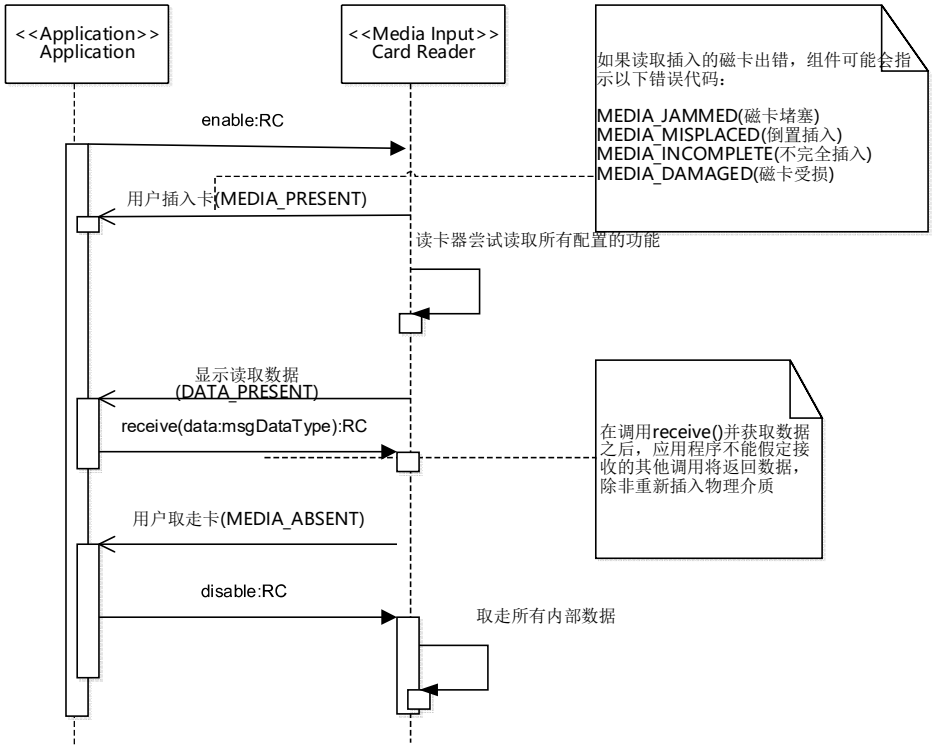

图 10　　插卡式磁卡读卡器的典型序列图

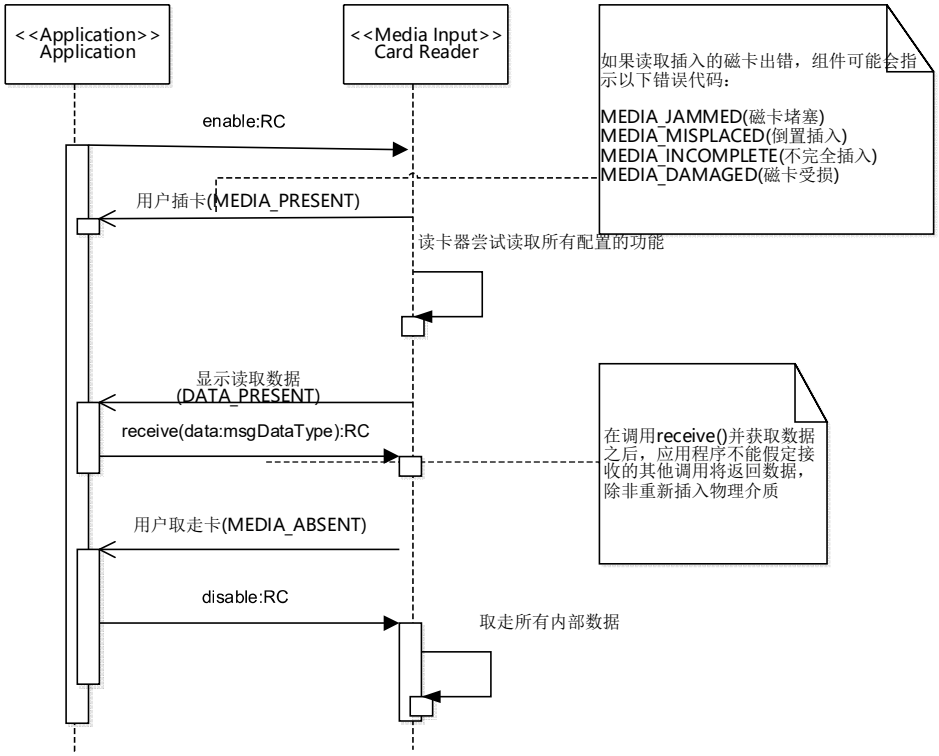

图 11　　刷卡式磁卡读卡器的典型序列图

9.5 GPP 打印机

9.5.1 设备描述

GPP打印机可以打印SVG格式的软纸登机牌。它没有暂存装置，打印完成后，打印用户直接拿到软纸登机牌。平台提供程序还可以选择使用同一组虚拟组件，来打印虚拟组件属性中支持的AEA和SVG类型数据。另一方面，即使两组虚拟组件代表相同的物理打印机，另一个平台提供商也可以选择使用完全不同的虚拟组件集来表示SVG和AEA数据。

注：根据其功能，CUSS可能没有GPP打印机，也可能有一个GPP接口，还可能有两个或更多用于其他特殊功能的GPP打印机，如重型行李条（自助行李托运设备使用）。

9.5.2 虚拟组件链接图

图12是GPP打印机链接图。

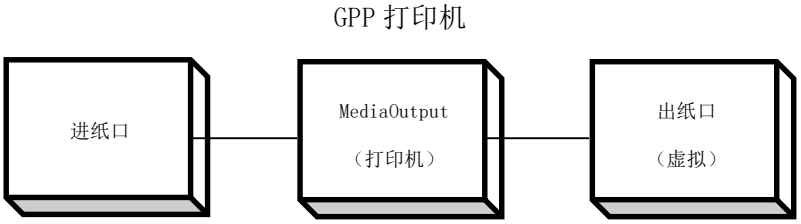

图 12　　GPP 打印机链接图

9.5.3 虚拟组件链接的描述

GPP打印机组件链接进纸口和虚拟出纸口

9.5.4 属性

表79是GPP打印机MediaOutput属性表，表80是GPP打印机的Dispenser属性表。

表 79　　GPP 打印机 MediaOutput 属性表

属性	值
MediaType.MediaTypeListDef	MediaType.MediaTypeDef.Printed
MediaOutput.MediaType.type	MediaOutput.MediaType.GeneralPurposeDoc
MediaOutput.supportedDataTypes	DataType.SVG DataType.AEA（若虚拟组件同时支持 AEA 和 SVG）

表 80　　GPP 打印机的 Dispenser 属性表

属性	值
Dispenser.kind	Dispenser.DispenserType.virtual_

	出纸口是虚拟的，因为不需要 offer（）。这是因为一旦打印出来，用户就将拿走软纸登机牌。 若调用 Offer（），平台将生成 StatusCode 为 OK 的响应，以便航司应用程序知道平台在出纸口组件中没有物理传感器。 若对虚拟出纸口调用 offer()，offer()必须阻塞直至移除软纸登机牌（若平台支持移除软纸登机牌的功能）
MediaOutput.maxPrintSizeX MediaOutput.maxPrintSizeY	文档的宽度和高度，以毫米为单位，用于区分支持不同尺寸纸张的多台打印机。
MediaOutput.PrintOrientation	表示文档是纵向（X 比高度 Y 窄）或横向（X 比高度 Y 宽）
MediaOutput.Manufacturer.firmwareVersion	可能包含有关此 GPP 打印的文档的特殊性质的其他说明（例如，自助行李托运系统的重型标签）

9.5.5 GPP 打印机的典型序列图

图13是刷卡阅读器上读取磁卡的典型序列图。

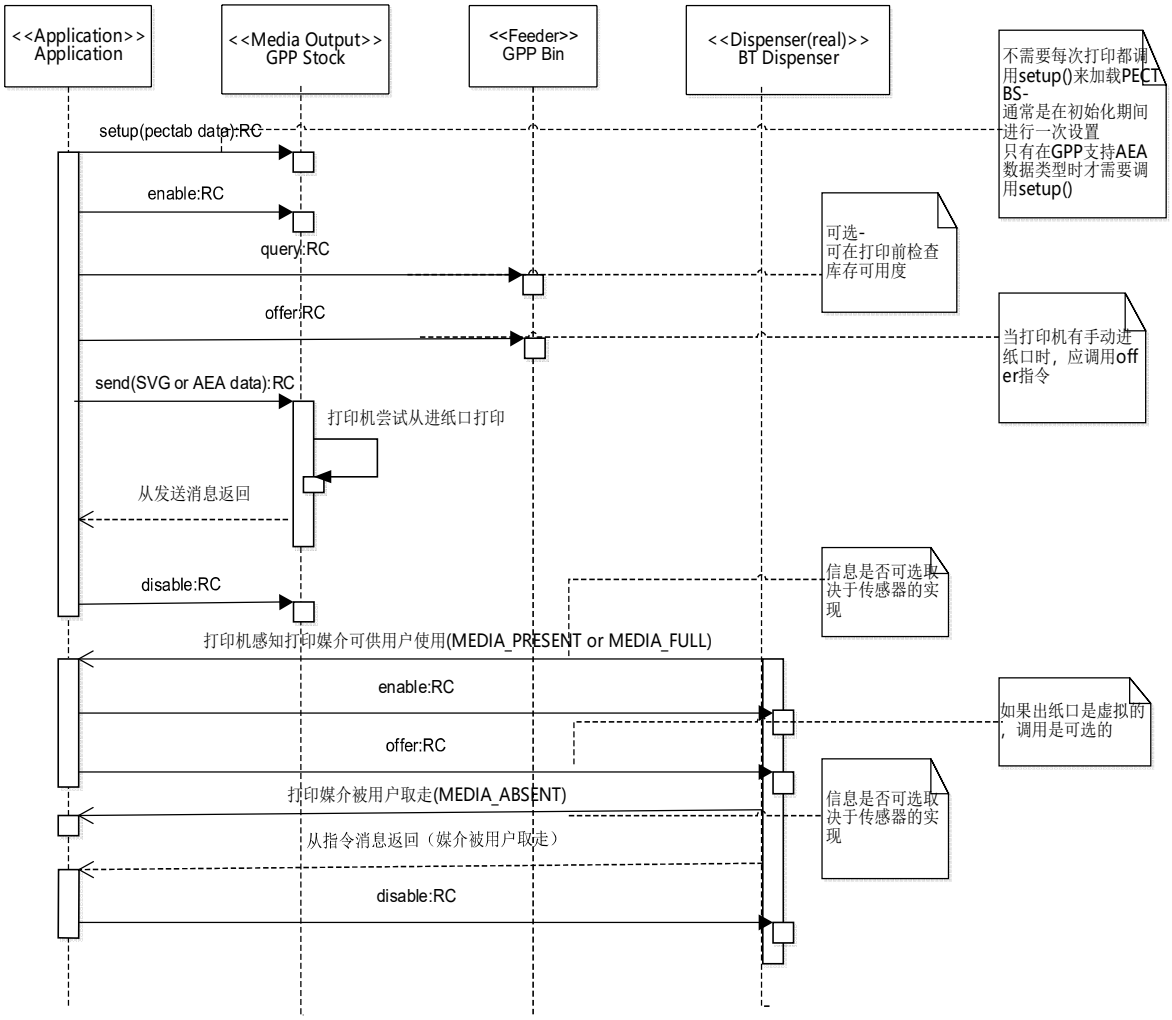

图 13　　刷卡阅读器上读取磁卡的典型序列图

9.6 刷卡护照阅读器

9.6.1 设备描述

刷卡护照阅读器接受并读取带有OCR格式数据的护照。使用一个MediaInput组件代表刷卡阅读器。

9.6.2 虚拟组件链接图

图14是刷卡护照阅读器的链接。

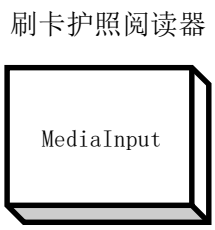

图 14　　刷卡护照阅读器的链接

9.6.3 虚拟组件链接的描述

一个单独的MediaInput虚拟组件，配置为刷卡阅读器类型。

9.6.4 属性

表81是刷卡护照阅读器的**MediaInput**属性表。

表 81　　刷卡护照阅读器的 **MediaInput** 属性表

属性	**值**
MediaType.MediaTypeListDef	MediaType.MediaTypeDef.Printed
MediaInput.typeOfReader	MediaInput.ReaderType.DIP MediaInput.ReaderType.Swipe
MediaOutput.supportedDataTypes	DataType.MSG
MediaInput.numberOfTracks	<int>，取决于 OCR 的实际轨道数。
ComponentFonts.usedStandard	ComponentFonts.BarcodeStandard.nonApplicable BarcodeTypes

9.6.5 刷卡护照阅读器的典型序列图

图15是刷卡阅读器上读取护照的典型序列图。

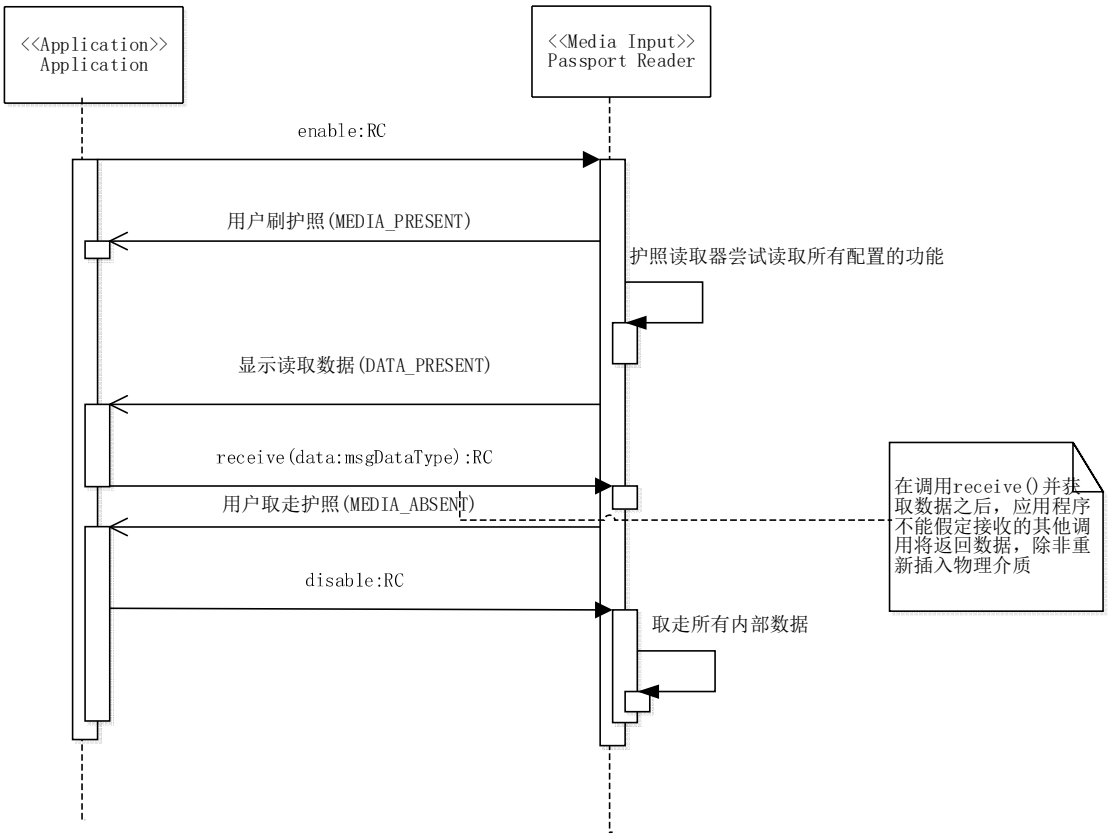

图 15　　刷卡阅读器上读取护照的典型序列图

9.7 条形码扫描仪

9.7.1 设备描述

条形码扫描仪读取它所支持的编码技术的条形码。使用一个MediaInput组件代表条形码扫描仪。OCR数据存储在MSG数据类型中的数据记录中。若条形码扫描器支持多个条形码，则单个OCR数据存储在MSG数据类型中的不同数据记录中。

9.7.2 虚拟组件链接图

图16是条形码扫描仪的链接图。

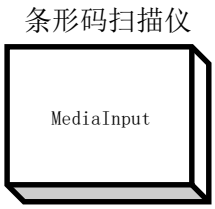

图 16　　条形码扫描仪的链接图

9.7.3 虚拟组件链接的描述

配置为有效读取器类型之一的MediaInput虚拟组件。

9.7.4 属性

表82是条形码扫描仪的MediaInput属性表。

表 82　　条形码扫描仪的 MediaInput 属性表

属性	值
MediaType.MediaTypeListDef	MediaType.MediaTypeDef.Printed
MediaInput.typeOfReader	MediaInput.ReaderType.PenScan MediaInput.ReaderType.Contactless MediaInput.ReaderType.Swipe MediaInput.ReaderType.FlatbedScan
MediaOutput.supportedDataTypes	DataType.MSG
ComponentFonts.usedStandard	ComponentFonts.BarcodeStandard.Code39 ComponentFonts.BarcodeStandard.Code128 ComponentFonts.BarcodeStandard.Code2of5 为了将它与护照扫描仪区分开来，条形码标准必须是以上之一。
Manufacturer.FirmwareVersion	<string>应包含支持的条形码列表,例如 PDF417 等。

9.7.5 条形码扫描仪的典型序列图

图17是读取条形码的典型序列图。

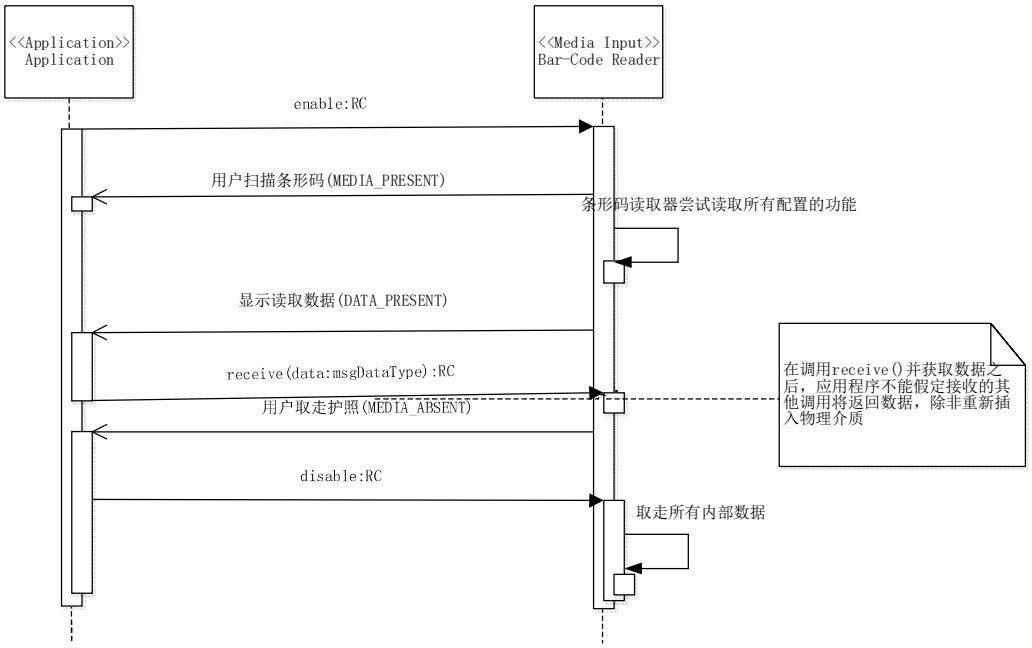

图 17　　读取条形码的典型序列图

9.8 综合行李系统（自助行李托运 AEA-SBD）：

CUSS定义使用自助行李托运系统的输送设备的方法——AEA-SBD允许使用AEA2012-2规范对行李托运设备进行完全控制。

9.8.1 设备描述

综合行李系统是一种复杂的设备，允许乘客自己办理行李托运。通常这些装置由一组单独的装置组成，用于称重和检查行李的尺寸的装置，以及在将行李送入机场行李分拣系统之前检验行李上的标签。扫描和验证行李标签通常基于IATA第740号决议中定义的LicensePplate，并且也可以由RFID天线提供支持，但规范不限制使用的条形码格式。

综合行李输送机系统不适用于需要X光检测爆炸物或其他重要安全检测的行李,因为此任务通常在行李登机过程之前或之后完成。

综合行李系统始终包括传送带并集成到行李分拣系统中。柜机可以同时连接到行李秤和综合行李传送带。

综合行李系统界面不能取代CUSS中的读卡器，护照阅读器和文档扫描仪接口。若这些设备配置在综合行李系统处，CUSS平台必须为设备提供普通的CUSS组件接口。

除了CUSS接口之外，平台还可以提供可选AEA-SBD接口，但这不是必需的。

该组件定义扩展了现有的Conveyor组件定义。运行CUSS平台且带有自助行李托运设备的，必须同时提供此AEA-SBD接口组件以及完整的CUSS传送器组件组。

9.8.2 虚拟组件链接图

选择使用AEA-SBD接口的CUSS航司应用程序只应获取前面列出的UserOutput和DataOutput组件(若存在且需要），并且不得获取与CUSS SBD接口相关的任何其他组件（传送带等）。

9.8.3 虚拟组件链接的描述

为阅读器可以读取的合法数据类型设置了同一个MediaInput虚拟组件。图18是自助行李托运AEA-SBD的链接图。

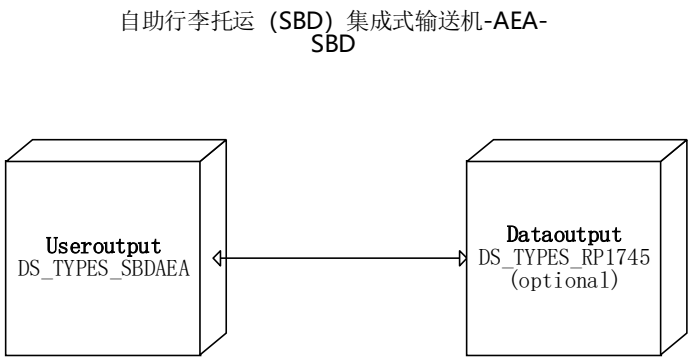

图 18　　自助行李托运 AEA-SBD 的链接图

9.8.4 属性

表83是自助行李托运AEA-SBD的UseOutput表，表84是自助行李托运AEA-SBD的DataOutput表。

表 83　　自助行李托运 AEA-SBD 的 UseOutput 表

UseOutput	
属性	**值**
Manufacturer.FirmwareVersion	DS_TYPES_SBDAEA,表示该设备是支持 AEA2012-2 SBD 扩展的综合行李输送机。

表 84　　自助行李托运 AEA-SBD 的 DataOutput 表

DataOutput airport BHS - 可选	
属性	**值**
Manufacturer.FirmwareVersion	DS_TYPES_RP1745，表示数据接口支持符合 RP1745 的 BSM。

重要状态条件：

当航司应用程序获取CUSS-SBD组件后，若尝试在AEA-SBD组件上调用Acquire()，平台应返回RC_DENIED。

9.8.5 自助行李托运的典型序列图

图19是自助行李托运时序图。

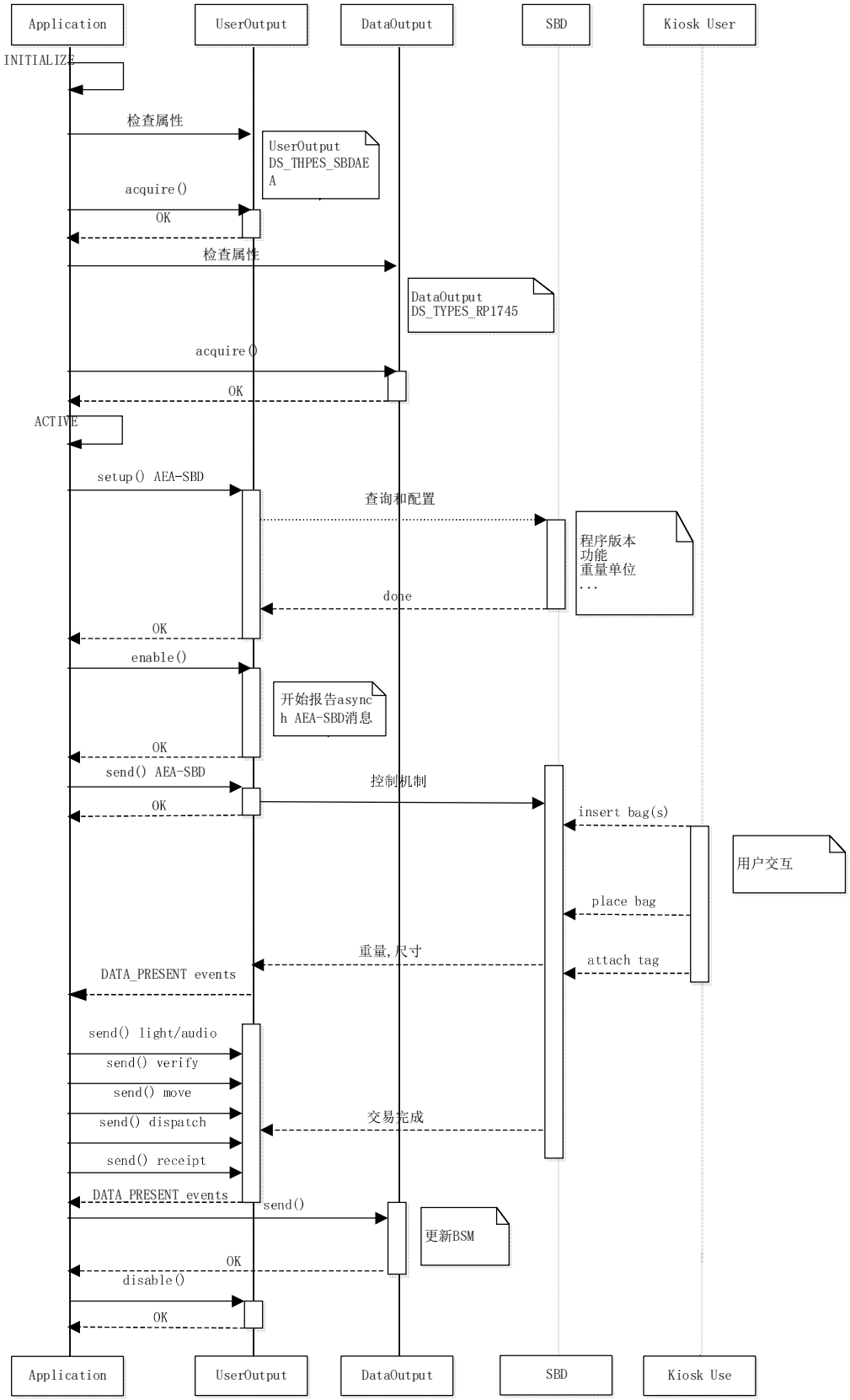

图 19　　自助行李托运时序图

9.9 身份证阅读器（ID Reader）

9.9.1 设备描述

身份证阅读器组件用于读取标准身份证信息，目前支持两个格式数据读取（1.全数据读取；2.身份证号码读取），MediaInput组件用于支持IDR功能。身份证阅读器必须能正确读取居民身份证和外国人永久居留身份证。

9.9.2 虚拟组件连接图

图20是身份证阅读器链接图。

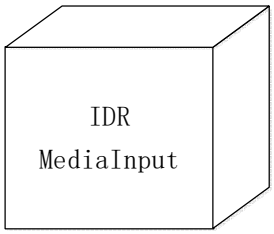

图 20　　身份证阅读器链接图

9.8.3 虚拟组件链接的描述

配置为有效读取器类型之一的MediaInput虚拟组件。

9.8.4 属性

表85是身份证阅读器MediaInput表。

表 85　　身份证阅读器 MediaInput 表

属性	值
MediaInput.supportedDataTypes	DataType.AEA
Manufacturer.FirmwareVersion	IDR
realName	IDCardReader

9.8.5 身份证阅读器的典型序列图

图21是身份证阅读器的典型序列图。

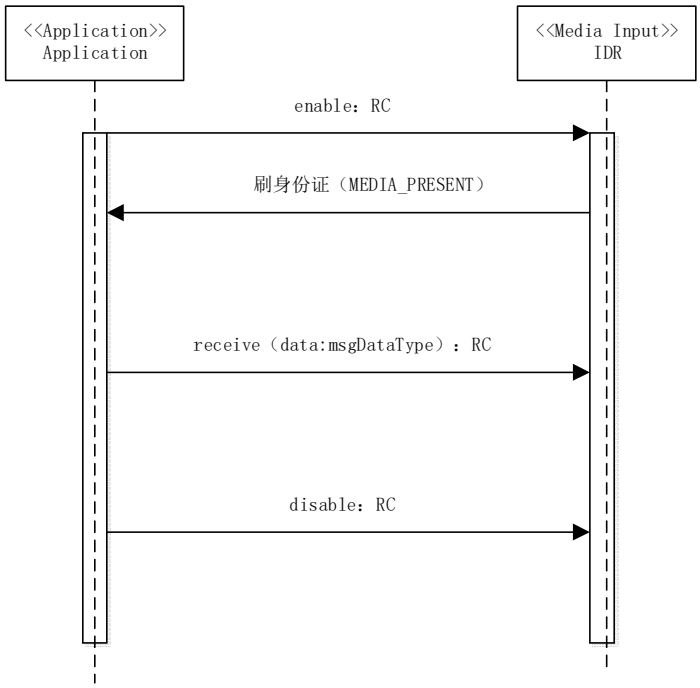

图 21　　身份证阅读器的典型序列图

附 录 A

（资料性附录）

FunctionReturnCode、EventCode、StatusCode和DataStatuCode

A1 FunctionReturnCode

ReturnCode描述了接口调用（指令）的字典式句法分析结果。若ReturnCode是0(RC_OK)，那么就接受该指令。NegativeReturnCode意味着拒绝该指令。表A1是FunctionReturnCode表。

表A1　　FunctionReturnCode表

FunctionReturnCode		
代码	**名称**	**描述**
0	RC_OK	接受指令。函数未检测到错误。
-1	RC_REFERENCE	无效的航司应用参考。调用航司应用程序正使用还没有由航司应用程序管理器分配的无效航司应用程序令牌。
-2	RC_STATE	无效的状态。航司应用程序在不正确的状态调用被调函数。
-3	RC_DENIED	拒绝访问。不允许航司应用程序使用此函数。
-4	RC_PARAMETER	参数错误。传递参数给被调函数时检测到错误。
-5	RC_ANY_PARAMETER	使用 CORBA::any type 时出错。包含在 CORBA::any type 中的数据类型可能不可用。即数据流参数，CORBA::any 的别名，在发送/设置函数中传递的数据流参数不是他接受的数据类型之一：aeaDataType，svgDataType，msgDataType 或 nilDataType。
-6	RC_LISTENER	没有设置事件监听。没有给异步事件设定监听参考。
-7	RC_SHARE	无效的共享模式。当 SP　SM 尝试在没有得到允许的情况下对设备进行独占访问时使用此选项。
-8	RC_UNAUTHORIZED	试图发出未经授权的命令。在发送/设置函数中传递的数据流参数可以包含未接受的 ATB 或 SVG 命令和/或数据。
-9	RC_ERROR	包含以上未定义的其他错误。
-10	RC_NOT_SUPPORTED	不支持该函数（未实现）。

A2 EventCode

EventCode或者是反应一个航司应用程序，或者是一个组件的状态转换（或者在没有状态转换的情况下本身的状态）。表A2是EventCode表。

表A2　　EventCode表

EventCode			
代码	**名称**	**用法**	**描述**
000	EC_OK	System Manager	用于返回事件调用 SuspendAll, ResumeAll 或 StopAll 指令。
组件状态转换			
001	EVENTHANDLING_ READY	外设	航司应用程序已成功执行设备组件的指令（只有软条件和 OK）。当设备组件从硬件状态恢复时，也会使用此选项。
002	UNAVAILABLE_ RELEASED_ PLATFORM	外设	授权平台组件已发布脱机或无法正常运行的设备组件(例如，在 CAM 将航司应用程序改变为禁用(DISABLED)状态之前）。
003	EVENTHANDLING_ UNAVAILABLE	外设	由于硬件条件（诸如变为脱机或不可用），所获取的设备组件已经变成不可用状态。
004	UNAVAILABLE_ RELEASED_ APPLICATION	外设	航司应用已经发布了在线或者正常运行的设备组件。
005	READY_RELEASED_ APPLICATION	外设	航司应用已经发布了在线或者正常运行的设备组件。
006	READY_RELEASED_ PLATFORM	外设	授权平台组件已发布脱机或无法正常运行的设备组件(例如，在 CAM 将航司应用程序改变为禁用(DISABLED)状态之前）。
007	RELEASED_READY	外设	航司应用已经成功接受在线或者正常运行的设备组件。
008	RELEASED_ UNAVAILABLE	外设	航司应用已经成功接受在线或者未正常运行（不可用）的设备组件。
航司应用程序状态转换			
EventCode			
代码	**名称**	**用法**	**描述**
101	INITIALIZE_DISABLED	CAM -> App	状态转换 **Disable**：由于在初始化（INITIALIZE）状态下的错误行为，CAM 将航司应用程序置于禁用(DISABLED)状态。
102	AVAILABLE_ DISABLED	CAM -> App	状态转换 **Disable**：由于在 AVAILABLE 状态下的错误行为，CAM 将航司应用程序置于禁用(DISABLED)状态。
103	ACTIVE_DISABLED	CAM -> App	状态转换 **Disable**：由于在激活(ACTIVE)状态下的错误行为（KILL_TIMEOUT 到期），CAM 将航司应用程序置于禁用(DISABLED)状态。
104	UNAVAILABLE_ AVAILABLE	CAM -> App	状态转换 **Wait**：在确定 CUSS 环境足以正确执行之后，航司应用程序已请求转换为 AVAILABLE 状态。
105	AVAILABLE_ACTIVE	CAM -> App	状态转换 **Activate**：当用户在 CLA 选择图标后，CAM 将航司应用程序置于激活(ACTIVE)状态。

106	ACTIVE_AVAILABLE	CAM -> App	状态转换 **Wait**：完成会话之后，航司应用程序已经请求返回 AVAILABLE 状态。
107	INITIALIZE_STOPPED_ST OP	CAM -> App CAM -> App	状态转换 **Stop**：当航司应用在初始化（INITIALIZE）状态时，航司应用程序或 SM 请求停止该航司应用程序。
108	AVAILABLE_STOPPED_S TOP	CAM -> App CAM -> App	状态转换 **Stop**：当航司应用在 AVAILABLE 状态时，航司应用程序或 SM 请求停止该航司应用程序。
109	ACTIVE_STOPPED_ STOP	CAM -> App	状态转换 **Stop**：当航司应用在激活(ACTIVE)状态时，航司应用程序或 SM 请求停止该航司应用程序。
110	SUSPENDED_ STOPPED_STOP	CAM -> App	状态转换 **Stop**:CAM 根据已暂停航司应用程序的 SM 请求将航司应用程序置于未启用(STOPPED)状态。
111	DISABLED_STOPPED_ST OP	CAM -> App	状态转换 **Stop**：在人工交互后，CAM 或 SP SM 已请求停止在禁用(DISABLED)状态的航司应用程序。
112	SUSPENDED_ AVAILABLE	CAM -> App	状态转换 **Resume**：根据已经停止它的同一个 SM 的请求，CAM 将航司应用程序返转换回 AVAILAVLE 状态。
113	AVAILABLE_ SUSPENDED	CAM -> App	状态转换 **Suspend**：根据 SM 要求，CAM 将航司应用程序状态由 AVAILABLE 状态转换为 SUSPENDED。
114	INITIALIZE_STOPPED_RE START	CAM -> App	状态转换 **Restart**：由于系统重启，将停止或重新加载航司应用程序。
115	AVAILABLE_ STOPPED_RESTART	CAM -> App	状态转换 **Restart**：由于系统重启，将停止或重新加载航司应用程序。
116	ACTIVE_STOPPED_ RESTART	CAM -> App	状态转换 **Restart**：由于系统重启，将停止或重新加载航司应用程序。
118	SUSPENDED_ STOPPED_RESTART	CAM -> App	状态转换 **Restart**：由于系统重启，将停止或重新加载航司应用程序。
119	STOPPED_INITIALIZE	CAM -> App	状态转换 **Load**： CAM 根据 SM 或它自己的要求，加载一个 STOPPED 的航司应用程序。
120	DISABLED_INITIALIZE	CAM -> App	状态转换 **Load**:在人工交互后,CAM 根据 SM 或它自己的要求，加载一个 DISABLED 的航司应用程序。
121	UNAVAILABLE_ STOPPED_RESTART	CAM -> App	状态转换 **Load**：由于系统重启，将停止或重新加载航司应用程序。
122	UNAVAILABLE_ DISABLED	CAM -> App	状态转换 **Disable**：由于在 UNAVAILABLE 状态下的错误行为，CAM 将航司应用程序置于禁用(DISABLED)状态。
123	UNAVAILABLE_ SUSPENDED	CAM -> App	状态转换 **Suspend**：根据 SM 要求，CAM 将航司应用程序状态由 UNAVAILABLE 状态转换为 SUSPENDED。
127	SUSPENDED_ UNAVAILABLE	CAM -> App	状态转换 **Resume**：根据已经停止它的同一个 SM 的请求，CAM

			将航司应用程序返转换回 UNAVAILABLE 状态。
128	UNAVAILABLE_ STOPPED_STOP	CAM -> App	状态转换 **Stop**:航司应用程序或 SM 要求停止在 UNAVAILABLE 状态的航司应用程序。
129	INITIALIZE_ UNAVAILABLE	CAM -> App	状态转换 **Check**：完成初始化之后，航司应用程序申请转换为 UNAVAILABLE 状态。
130	AVAILABLE_ UNAVAILABLE	CAM -> App	状态转换 **Check**：确定 CUSS 环境不足以完成执行之后，航司应用程序请求返回到 UNAVAILABLE 状态。
132	ACTIVE_ACTIVE	CAM -> App	状态转换 **Wait**：航司应用程序指示在持久单航司应用程序模式中已经启动客户事务。
133	ACTIVE_UNAVAILABLE	CAM -> App	状态转换 **Check**：确定 CUSS 环境不足以完成执行之后，航司应用程序请求返回到 UNAVAILABLE 状态。
航司应用程序或设备状态			
201	RELEASED	外设	航司应用程序已释放（或尚未获取）设备组件。它可能不可用。
202	UNAVAILABLE	外设	之前获取的设备组件处于脱机状态或无法正常运行（不可用）。
		航司应用程序	航司应用程序在 UNAVAILABLE 状态。
203	READY	外设	之前获取的设备组件处于脱机状态或无法正常运行（备用）。
204	STOPPED	航司应用程序	航司应用程序在未启用(STOPPED)状态。
205	SUSPENDED	航司应用程序	航司应用程序在挂起(SUSPENDED)状态。
206	DISABLED	航司应用程序	航司应用程序在禁用(DISABLED)状态。
207	INITIALIZE	航司应用程序	航司应用程序在初始化（INITIALIZE）状态。
208	AVAILABLE	航司应用程序	航司应用程序在 AVAILABLE 状态。
209	ACTIVE	航司应用程序	航司应用程序在激活(ACTIVE)状态。若 SESSION_TIMEOUT 已经过去，CAM 可以使用此 EventCode 要求航司应用程序完成其会话。
210	BUSY	外设	器件组件处于瞬态 BUSY（例如正在进行读/写）。

A3 StatusCode

StatusCode描述的是组件的当前状态，或者是组件接口调用的语义分析结果，或者是组件接口调用的执行结果。表A3是**StatusCode表**。

表A3　　StatusCode表

StatusCode 描述

代码	名称	描述	事件类型
000	OK	在线设备和就绪或接口调用不会产生错误。	public[2]
001	TIMEOUT	同步或异步函数调用超时	private
002	WRONG_STATE	组件处于错误状态以接收此调用	private platform
003	CANCELLED	异步函数调用已取消。	private
004	SOFTWARE_ERROR	在执行功能期间检测到可恢复的软件错误。	private platform
005	ALMOST_OUT_OF_ TIME		private platform
006	OUT_OF_SEQUENCE	函数被不按顺序调用。（例如，在启用或调用启用/禁用两次之前调用 Send/Receive）	private
相关媒介(100-199)			
101	MEDIA_JAMMED	文件或磁条卡卡在设备内。	public
102	MEDIA_MISPLACED	文档或磁条卡插入不正确。例如 ATB 文档上插入。	private platform[3]
103	MEDIA_PRESENT	文档已插入设备。即使媒介没有保存在外围设备中，也必须发送事件（例如，刷卡读卡器或插卡读卡器）	private
104	MEDIA_ABSENT	没有要提供的文件（对于出纸口/进纸口）或文件在启用时从设备中删除（MediaInput，MediaOutput）。在后一种情况中，即使文件没有保存在外围设备中，也必须发送事件（例如，刷卡或 DIP 读卡器）	private
105	MEDIA_HIGH	组件（如：废纸槽）达到 AlmostFullLevel 阈值	public
106	MEDIA_FULL	进纸口/出纸口/废纸槽设备都是文件。	public
107	MEDIA_LOW	进纸口达到 AlmostEmptyLevel 阙值。	public
108	MEDIA_EMPTY	进纸口没有文件（没有纸了）。	public
109	MEDIA_DAMAGED	例如：卡、登机牌物理损坏	public
110	MEDIA_ INCOMPLETELY _ INSERTED	文档未完全插入设备并被删除	private
相关数据(200-299)			

[2] 对于已请求的事件是 private

[3] 仅对无用类

StatusCode 描述			
代码	**名称**	**描述**	**事件类型**
201	FORMAT_ERROR	用于发送或接收的数据格式检测到错误。 例如 ATB 卡、登机牌编码错误或错误 PECTAB 或无效数据流（AEA99 或 SVG）。	private platform[4]
202	LENGTH_ERROR	为发送或接收提供的数据流是不完整的。	private platform[6]
203	DATA_MISSING	当使用发送或接收函数时没有数据。	private platform[6]
204	PHYSICAL_ERROR		private platform[6]
205	DATA_PRESENT	从插入的文档读取数据，航司应用程序可以通过调用 receive 获取此数据。	private
相关硬件错误 (300-399)			
301	CONSUMABLES	例如：打印机色带，头	public
302	HARDWARE_ERROR	一个错误，是由于由于硬件故障导致组件 UNAVAILABLE。	public
303	CRITICAL_ SOFWARE_ERROR	在执行函数（能使组件转变为 UNAVAILABLE 的）期间检测到软件错误	public
304	NOT_REACHABLE	设备不可连接（未知状态）。	public
305	NOT_RESPONDING	设备可连接但是无应答或未就绪。	public
306	THRESHOLD_ERROR	出现太多错误。	public
307	THRESHOLD_USAGE	插入或取出卡、登机牌的次数过多。	public
308	CONFIGURATION_ ERROR		public
309	SESSION_TIMEOUT	ACTIVE 状态的航司应用会话超时。当 SessionTimeout 结束时发送。	private platform
310	KILL_TIMEOUT	在将航司应用程序转换至禁用(DISABLED)状态之前发送.. 若航司应用程序仍处于激活(ACTIVE)状态，则在 KillTimeout 结束时发送。	private platform
相关航司应用(400-599)			
4xx	Application dependent (technical)	这些事件为航司应用程序提供了向 SM 发布事件的可用性，并就 EventCode 的确切含义达成双边或多边协议。	private

[4] 仅对输出类

StatusCode 描述			
代码	**名称**	**描述**	**事件类型**
5xx	Application dependent (security)	这些事件为航司应用程序提供了向 SM 发布事件的可用性，并就 EventCode 的确切含义达成双边或多边协议。	private
航司应用程序状态改变请求(800-899)			
801	CUSS_MANAGER_ REQUEST	当 CAM 向航司应用程序发送事件并根据 CAM 本身的请求更改其状态时使用。	private platform
802	SP_SYSTEM_ MANAGER_ REQUEST	当 CAM 向航司应用程序发送事件并根据 SP SM 的请求更改其状态时使用。	private platform
803	AL_SYSTEM_ MANAGER_ REQUEST	当 CAM 向航司应用程序发送事件并根据 AL SM 的请求更改其状态时使用。	private platform
804	CL_APPLICATION_REQU EST	当 CAM 向航司应用程序发送事件并根据 CLA 的请求更改其状态时使用。	private platform
805	AL_APPLICATION_REQU EST	当 CAM 向航司应用程序发送事件并因为航司应用程序的要求更改其状态时使用。	private platform
相关航司应用(900-999)			
9xx	Application dependent (business, functional)	这些事件为航司应用程序提供了向 SM 发布事件的可用性，并就 EventCode 的确切含义达成双边或多边协议。	private

A4 DataStatuCode

DataStatuCode描述MSG数据类型中每个数据记录的状态（有关MSG数据流的定义，请参阅第7.9节：数据Data）。表A4是**DataStatuCode表。**

表A4　　DataStatuCode表

DataStatuCode 描述		
代码	**名称**	**描述**
0	DS_OK	数据记录是 OK。
1	DS_CORRUPTED	数据记录被损坏（没有数据包括在内）。
2	DS_INCOMPLETE	数据记录是不完整的。
3	DS_ZEROLENGTH	数据记录长度为 0。

有关如何将新的和扩展的媒介类型和信息用于其他DS_TYPES和其他数据状态代码的信息，请查看第9章。

需要注意的是，若平台检测到并返回DS_CORRUPTED作为数据状态，则它不应在数据记录中包含任何数据。

附 录 B

（资料性附录）

组件映射

B1 介绍

物理组件数量应尽量最少。使用可以读写相同库存的设备，而不是使用两个设备。使用可以管理多种类型库存的设备，例如：磁卡，ATB2和芯片卡，而不是每种库存类型的一个设备。

实体组件代表安装在CUSS平台上的实体物理外围设备。许多外围设备被映射到虚拟组件类型，为了表明它们具有相同的一般特征：媒介，数据类型等。

可以将实体组件映射到一组或多组不相交的虚拟组件（例如，真正的ATB2打印机可以映射到虚拟BoardingPass打印机和虚拟ReceiptPrinter，假设此ATB2打印机配置是同时具有登机牌和收据库存）。有关详细信息，请参阅第6.1.1节：虚拟组件概念。

航司应用程序开发人员还应该查看第9章，它提供了与此处相同的信息，但更实际地组织起来以协助编写代码，以在柜机航司应用程序中查找和使用CUSS设备。

B2 实体组件映射

表B1列出了典型的CUSS实体组件，以及要求（强制，推荐，可选），和每个实体组件所包含的相关虚拟组件。

根据设备配置（支持的设备功能或介质类型），有些实体组件映射需要多个相同类型的虚拟组件，但该列表仅包含该类型的一个代表。实体组件未按公司和型号列出，而是按通用组件列出的，允许减少列表的长度并仍然提供全局视图。

这不是实体设备类型的完整列表，也不是可以在柜机上找到的实体设备类型，虚拟组件类型或虚拟组件链接的完整列表。具体地说，它并不意味着可以作为一个参考来区别柜机上的设备：它也不能反映出柜机上可能存在的所有CUSS设备。

最重要的是，下面列出的“实体组件名称”，并不能保证能够与柜机中该设备类型的虚拟组件特性值相匹配。CUSS航司应用程序不应该仅依赖于realComponentName来确定柜机上存在的功能和设备。它必须分析组件链接和特性，来选择和使用它所需的虚拟组件。第9章详细介绍了如何执行此操作。

表 B1　　典型的 CUSS 实体组件

实体组件名称	要求	虚拟组件类型/名称
ATB2Device	推荐	MediaInput
		MediaOutput
		Dispenser
		Feeder
ATB2DeviceWithEscrow	可选	MediaInput
		MediaOutput

实体组件名称	要求	虚拟组件类型/名称
		Dispenser
		Dispenser (托管)
		Capture (托管)
		Feeder
ATB2Printer	推荐	MediaOutput
		Dispenser
		Feeder
ATB2Reader	推荐	MediaInput
		Dispenser
		Capture (托管)
Audio	可选	DataOutput
BagTagPrinter	推荐	MediaOutput
		Dispenser
		Feeder
BarCodeScanner	可选	MediaInput
		Capture
BoardingPassCaptureBin	可选	Capture
BoardingPassDispenserBin	可选	Dispenser or Feeder
BoardingPassPrinter	强制	MediaOutput
		Dispenser
		Feeder
CardReader (Mag+Chip)	推荐	MediaInput（磁卡）
		MediaInput（芯片卡）
		Dispenser
ChipCardCaptureBin	可选	Capture
ChipCardDevice	可选	MediaInput
		MediaOutput
		Dispenser
ChipCardDispenserBin	可选	Dispenser or Feeder

实体组件名称	要求	虚拟组件类型/名称
ChipCardReader	可选	MediaInput
		Dispenser
ChipCardWriter	可选	MediaOutput
		Dispenser
		Feeder
Clock	强制	DataInput
		Dispenser
Display	强制	Display
DoorSensor	推荐	DataInput
Escrow	可选	Dispenser
		Capture
FingerprintReader	可选	UserInput
GPPrinter	可选	MediaOutput
		Dispenser
		Feeder
HardDisk	强制	Storage
Keypad	可选	UserInput
LEDIndicator	可选	UserOutput
MagneticCardDevice	可选	MediaInput
		MediaOutput
		Dispenser
		Feeder
MagneticCardEncoder	可选	MediaOutput
		Dispenser
		Feeder
MagneticCardReader	强制	MediaInput
		Dispenser（仅适用于电动读卡器）
Monitor	可选	Display
Network	强制	Network

实体组件名称	要求	虚拟组件类型/名称
OCRReader	可选	MediaInput
PassportReader	推荐	MediaInput
PinPadEncrypting	可选	UserInput
		DataOutput
ProximitySensor	可选	UserInput
RadioRFID	可选	MediaInput
ReceiptPrinter	推荐	MediaOutput
		Dispenser
		Feeder
TicketPrinter	可选	MediaOutput
		Dispenser
		Feeder
		Capture
TouchScreenOverlay	强制	UserInput
UPS	推荐	DataInput
VideoCamera	可选	UserInput
VisualCustomerAssistanceLight	可选	DataInput
Hardware WatchDog	推荐	DataInput

表B1并非全部可能的虚拟设备组合的完整列表，而是最常见类型设备的指南。例如，具有物理夹紧或整页扫描等功能的，新型护照阅读器可能确实具有实体出纸口组件。

航司应用程序开发人员必须确保他们检查链接的组件。例如，从MediaInput设备读取时，可能存在链接出纸口组件，在这种情况下，航司应用程序必须调用Offer()将文档返回给用户。

若航司应用程序需要播放声音，它应使用原生方法和API，只播放声音而不修改柜机的行为，如调整音量。

要确定柜机是否能够播放音频，航司应用程序应查找虚拟音频组件。然后，当且仅当存在虚拟音频组件时，航司应用程序才能播放本机声音。若柜机不能播放音频，CUSS平台可能不包含该音频组件。

若音频组件存在且航司应用程序调用Send()指令，平台应将RC_NOT_SUPPORTED返回给航司应用程序：这不存在向平台发送音频的CUSS标准方式。

注：不允许航司应用使用平台中未封装的外围设备，避免对其他共享设备的航司应用产生影响。

附 录 C

（资料性附录）

技术和标准

C1 介绍

CUSS平台必须支持以下所有典型技术和标准。

选择这些技术是为了允许单个航司应用程序在多个标准和商用浏览器技术（如Flash和Silverlight）上运行，并且在标准Java环境中运行。这允许航司应用程序提供者们可预测和一致的行为。

以下是CUSS柜机必须提供的概要列表，CUSS航司应用程序供应商必须在开发航司应用程序时预测和测试。

——标准CUSS浏览器是信息亭提供商选择的Microsoft Internet Explorer 8（IE8）或具有兼容模式的Microsoft Internet Explorer 11，如下所述

——标准CUSS java是用于命令行和浏览器插件环境的Oracle JRE7（7u21），在中等安全模式下运行，定义如下

——Adobe Flash 16（或更高版本）可用

——Adobe Shockwave 12（版本12.1或更高版本）可用

——Adobe AIR 16（或更高版本）可用

——Apple Quicktime 7（版本7.7.6或更高版本）可用

——Windows Media Player 11（版本11.0或更高版本）可用

——Microsoft Silverlight 5（发布5.1或更高版本）可用

——Adobe PDF Reader XI（版本11.0或更高版本）可用

——AEA2009用于文档打印（ATB和BTP）

——AEA2012-2适用于自助行李托运设备

——除标准CUSS浏览器外，还提供Google Chrome 40（或更高版本）

根据以下信息，上述概要中的“或更高版本”是指柜机或平台提供商的选择。航司应用程序提供商不得强制要求比上面列出的最低版本更新的版本。

尽管Windows XP不再使用，许多柜机在发布时会继续在XP上运行。一般来说，上面的列表包括保留对Windows XP支持的工具发布者提供的最新版本。

C2 软件许可和分发

CUSS标准需要多种工具和技术，作为CUSS兼容环境的一部分，例如Microsoft、Oracle、Adobe和Google。

每一项外部技术均须获得其提供商有关商业许可，再分发，最终用户许可协议，出口限制，安全数据收集和数据隐私政策，更新和修补，责任免除和免责声明，以及保证其他法律和商业要求的使用约束。

尽管这些工具是CUSS 柜机的组成部分，但IATA和CUSS技术标准并未废除或取代任何这些第三方使用要求。

每个平台提供商，平台运营商和航司应用程序提供商都应该审查、理解并遵守，适用于在特定站点安装、部署和使用第三方技术的任何此类要求，如有需要，也可拓展应用范围，如航司应用有需要，也可拓展此类工具应用范围，使用可靠的开源应用。

C3 标准 CUSS Java 环境

运行在CUSS 柜机上的航司应用程序几乎总是需要继承Java技术为了充分利用CUSS标准的CORBA接口。

CUSS技术标准规定了特定版本，以便CUSS航司应用程序提供商对其开发和测试具有已知的一致的Java技术要求，这与机场和柜机提供商之间也是一致的。

CUSS 柜机要求的Java版本不是Java的当前版本，事实上它被认为已过时且不受Oracle支持。但是，选择此版本的Java是为了确保与大量现有CUSS航司应用程序的兼容性和最小变化。

Oracle在Java的更新版本中引入了许多实现Oracle Java愿景的更改。然而，许多近期的更改可能会对某些CUSS航司应用程序提供商提出额外要求，这可能是实施的重大挑战。

在旧版Java中使用CUSS可以减轻这种变化速度，并确保航司应用程序提供商具有更高的可预测性。因此，**CUSS定义一种标准的CUSS Java环境，Oracle JRE7更新到21（中等安全），它必须作为标准CUSS浏览器中Java插件的默认java环境以及系统命令行环境运行。**

为了符合CUSS，柜机**必须**符合以下要求：

Java 运行时环境（命令行）：

——柜机必须提供命令行 Java 运行时环境。

——柜机命令行的默认 Java 必须时 JRE7。“默认版本”被认为是在没有特定路径的情况下在柜机上调用“java.exe”时运行的版本。

——命令行上的 JRE7 版本必须是 JRE 7u21。不得安装较新的主要版本（如 JRE8）或较新的次要版本（如 JRE 7u45）。

——若提供了命令行 java 运行时环境的早期版本，则它们必须是此列表中的版本：

a) Java 运行时 1.3.1_06 (CUSS 1.0)

b) Java 运行时 1.3.1_19 (CUSS 1.1)

c) Java 运行时 1.3.1_20 (CUSS 1.2)

d) Java 运行时 1.5.0_04 (CUSS 1.2)

e) Java 运行时 JRE6u12 (CUSS 1.2)

——若提供了早期版本的命令行 java 运行时，则必须通过指定该版本的 java.exe 文件的完整路径名来调用它们。

Java 运行时环境（浏览器插件）：

——柜机必须提供 Java 运行时环境浏览器插件，并且必须将默认系统浏览器（在本规范中其他地方定义）配置为使用该浏览器插件。

——浏览器插件中提供的默认 java 必须是 JRE7。“默认版本”被认为是在使用类 ID“clsid:8AD9C840-044E-11D1-B3E9-00805F499D93”调用浏览器插件对象时运行的版本。

——浏览器中的 JRE7 版本必须为 JRE 7u21。不得安装较新的主要版本（例如 JRE8），并且

不得安装较新的次要版本（例如 JRE 7u45）。

——JRE7 浏览器插件必须是系统上安装的唯一插件。

——航司应用必须配置 JRE7 浏览器插件，为了避免在浏览器页面调用任何上述类 ID 时，出现弹出消息或确认。

——必须将 Java 插件安全模型配置为“中等安全性”，为了确保航司应用程序提供者的环境一致性。

——为了从与 HTML 内容相同的源加载 applet jar 文件，可能需要修改航司应用程序，由于 JRE7 中有跨域小程序加载的新限制。此技术调查是航司应用程序提供商的责任。

——尤其是，从本地柜机磁盘加载 jar 文件，从远程 Web 服务器上加载的 Web 小程序容器。

Java 运行时环境（GeneralComments）：

CUSS 平台提供商不得满足来自 CUSS 航司应用程序提供商的任何请求，为了安装或使用除此处所述的 JRE7 命令行和浏览器插件之外的 Java 任何版本。

不能与上述 Java 环境一起正常运行的 CUSS 航司应用程序不是一个符合 CUSS 标准的航司应用程序。

若 Java 仍然可用，将会重新审视 CUSS 的未来版本，且版本策略将生效。

C4 标准 CUSS 浏览器环境

在 CUSS 中，为了辅助提供一个更一致和和预测的环境，CUS 技术标准要求一个特定浏览器技术和环境作为默认环境。

现在 CUSS 定义了标准 CUSS 浏览器。这个标准的浏览器是 Microsoft Internet Explorer 8，它必须作为基于浏览器的航司应用程序运行的默认浏览器环境运行。

航司应用CUSS还定义了一个可备选择的CUSS浏览器，必须与标准CUSS浏览器一起部署，并作为CUSS航司应用程序提供商的选项提供。这个备用浏览器必须是Google Chrome 40或更高。

此备用浏览器必须是浏览器环境，仅适用于航司应用程序提供商在CUSS 柜机上运行时，特意请求在备用CUSS浏览器中运行其航司应用程序。另外，如果航司应用运行时需使用此两种之外的浏览器，建议航司软件中内置浏览器。

C5 演示工具和库

在技术表中使用“-”表示与先前版本的CUSS标准没有任何变化。若未定义早期版本，则该版本的CUSS不支持该技术。

CUSS1.4平台必须提供至少是在“1.4必需版本”这一列中列出的版本，工具的主要版本也必须如所示。

工具的新版本只有由CUSS技术解决方案组决定部署，方案组通过对本技术规范文档的更改或更新来决定。应即使可以从标准实体或组织获得更新，航司应用程序供应商也不能假设该工具的新版本可用于柜机。

允许存在一个例外，若需要更新到新的主要版本来纠正漏洞，或其他严重的下面指定工具的主要版本中未经供应商纠正的技术问题。

另一个例外，仅使用“主要”版本号架构，例如Adobe Flash和Google Chrome。在这些情况下，允许任何更新版本。

例如，这里给出了Adobe Shockwave，但适用于主要或次要版本发布的所有CUSS技术：

——所有 CUSS1.4 柜机平台提供商必须包括至少是 Adobe Shockwave 12.1。平台提供商可以选择部署 Adobe Flash 12 更高版本

- 例如，Adobe Shockwave 12.1.2 +，12.2,12.2 和其他版本的 Adobe Shockwave 12 在 CUSS 1.4 环境中是允许的并且是兼容的。

——平台提供商不得部署任何大于 12 的 Adobe Shockwave 主要版本：

- 例如，在 CUSS 环境中不允许使用 Adobe Shockwave 13.0，部署 Adobe Shockwave 13 的平台不符合 CUSS 网站。

——不得使用依赖大于 12 的 Shockwave 版本的 Shockwave 技术编写航司应用程序。

- 航司应用程序提供商必须意识到，它可能运行在比 Adobe Shockwave 12.1 更新的版本上。
- 然而，需要大于 12.1 的 Adobe Shockwave 版本的航司应用程序不 CUSS 兼容的航司应用程序。
- 航司应用提供商不能要求平台提供商安装比 12.1 更新的 Adobe Shockwave 12 的版本。
- 航司应用提供商不能要求平台提供商安装比 12 更新的 Adobe Shockwave 的主要版本（例如，Shockwave 13 or 13.2）。

另一个例外是仅使用“主要”版本号架构的工具，例如Adobe Flash和Google Chrome。 在这些情况下，允许任何更新版本。

——例如，这里给出了 Google Chrome，但适用于所有主要版本的 CUSS 技术：

——所有 CUSS 柜机平台提供商必须包括至少是 Google Chrome 40。

——平台提供商可以部署比 40 更高版本的 Google Chrome：

- 例如，在 CUSS 环境中允许使用 Google Chrome 42，并且部署 Google Chrome 的平台是 CUSS 标准兼容的网站。

——必须依赖比 40 更高版本的任意版本的 Google Chrome 来编写航司应用程序：

- 航司应用程序提供商必须意识到,它可能运行在比 Google Chrome 40 更新的版本上。
- 然而，要求 Google Chrome 版本超过 40 的航司应用程序不是 CUSS 标准兼容的航司应用程序
- 航司应用提供商不能要求平台提供商安装比 Google Chrome 40 更新的 Google Chrome 的主要版本。

绝对需要这一严格要求，以确保对 CUSS 航司应用程序可用的特定环境有一个共同的一致理解。因此，必须特别注意：

——柜机提供商必须确保 CUSS 柜机环境满足 CUSS 版本要求。

——航司应用程序提供商必须确保他们的开发团队设计和使用列出的版本，而不是工具的最新版本。

CUSS标准认为，有对所有航司应用程序提供者一致且可预测的环境和工具版本是需要优先考虑的事项。

允许各个航司应用程序提供商请求更新版本的工具，这将会需要强行更改部署在该网站的其他所有提供商的程序，这是不可接受的，因为CUSS将不再是可预测的或一致的环境。

与任何业务关系的情况一样，航司应用程序和平台提供商可以讨论在某个特定网站部署多种新版本工具的一对一选项，以满足航司应用程序需求，并在网站授予对其他航司应用程序施加操作风险的所有权。

C6 柜机 PC 系统要求

最低要求：

PC 对 CUSS 柜机的最低要求支持上面列出的所有强制性 CUSS 技术，是：

——具有 SSE2 指令集的 Pentium 4 2.33 GHz 或更高处理器（或等效）

——512MB RAM（基础平台，操作系统和单个 CUSS 航司应用程序）

——Windows 7 或 Windows 8.1

——对于现有的柜机，也允许使用旧版 Windows XP SP3

——每个航司应用程序至少有 1GB 的可用磁盘空间

——屏幕分辨率为 1024x768 或以上（或更高）

——这些是现有柜机机箱必须提供的绝对最低要求，以符合 CUSS 并提供所需的所有合规工具。

——为了满足这些技术和标准要求，可以部署 Windows（32 位或 x64）的任何其他版本或将来更高版本，都是可以接受的。

该平台还应为在柜机上运行的每个CUSS航司应用程序提供额外的128MB RAM。例如，运行五个CUSS航司应用程序的柜机所需最小总内存为1024 MB RAM。

单个航司应用程序可能会结合它们的所有进程使用最多192MB的内存。然而，假设平均航司应用程序不超过128MB，柜机的大小可以调整。

虽然这些是所需技术的最低系统要求，它们不一定是这些技术推荐的系统配置。尤其是，高清视频播放或复杂的动画可能会需要强加更高的硬件要求，甚至是CUSS不要求的特殊图形处理的硬件要求。

C7 航司应用程序可以使用的其他软件

CUSS是一种为了能在多个操作系统上运行而设计的标准，并且可以通过明确定义但固定的技术集保持互操作性。 因此，运行在CUSS 柜机的 CUSS航司应用程序必须基于Java或基于浏览器，并且只能使用上面列出的常用工具，以确保它们能够在所有平台上运行。

该方法用于确保跨平台、提供商和柜机网站的兼容性，确保柜机提供商的柜机图像和配置的完整性，并允许柜机提供者根据CUSS技术列表控制柜机图像。

因此，CUSS航司应用程序提供商**不能**为了允许部署其航司应用程序，要求在柜机上安装特殊的操作系统工具、库或框架。受限制的项目是，诸如本机.EXE程序、.DLL库，诸如.NET或者需要安装到或复制到柜机图像的其他非航司应用程序工具等OS框架。

航司应用程序要求的任何其他组件必须经过CUSS技术组的批准，并列入CUSS技术列表，也可能在CUSS标准的新版本中列出。平台提供商必须能安装和独立测试组件，以确保它们不会干扰平台其他部分。

例如，这些是不被允许的活动：

——在航司应用程序目录下安装 DLL 或 EXE（例如 SWT，一个需要 JNI 库的工具包）；在不属于航司应用程序提供商（共享或机场柜机 s）的系统或组件目录中安装 DLL 或 EXE。
——安装自定义浏览器插件。
——在航司应用程序目录中包含并运行客户 Java Runtime。
——安装特定的本机工具包或 API（例如.NET 框架）。
——这些是被允许的活动：
——安装字体资源（.TTF），以便在 java 或浏览器中可用。
——为了扩展语言支持，可以在柜机上请求多语言支持“包”。
——java 库和工具包作为航司应用程序的一部分来使用（提供或它们不需要 JNI 组件进程），如 xerces，log4j 等。
——安装安全通信所需的安全证书。

在柜机上运行时，航司应用程序提供商应遵守航司应用程序提供或包含的任何字体、库、工具包或其他资源的许可、条款和条件。

C8 航司应用程序的柜机或特定网站配置

因为CUSS标准旨在允许航空公司航司应用程序在所有柜机s和平台供应商中联合一致工作,航空公司应该最少需要在航司应用程序中拥有柜机、工作站或供应商特定的逻辑和配置。

何时以及是否需要特定网站的设置配置或航司应用程序，航司应用程序的集成和持续维护更加困难：它需要比其他航司应用程序更多的文档、协调和跟踪，并且可以使初始设置更加困难。总的来说，它还会影响CUSS独立于供应商且可互操作的这一观点。

因为很多原因，航司应用程序提供商的建议做法是，尽量最小化并尽可能消除柜机上航司应用所需的特定网站的设置或配置。（当然，在航司应用程序服务器或主机上，使用了许多特定网站的规则和设置。）

——航司应用程序不应该使用本地配置去设置和寻找哪个设备用在特定柜机。这应该在代码中完成，使用第 9 章和第 8 章的指南。
——若使用的话，将本地配置保持在最低限度。这类配置仅用于已知的柜机或设备互操作性问题。
——不应从本地配置业务逻辑。相反，航司应用程序或服务器逻辑应使用来自 CUSS 环境的柜机名称和站点代码，以从航空公司维护的中央配置中查找规则或配置。
——由于供应商互操作性或 CUSS 技术标准问题，需要的本地配置都应引起 CUSS 方案工作组的注意，以进行讨论和改革。

C9 服务器端航司应用技术

对CUSS航司应用程序架构的任何方面都没有任何限制，这些方面不在CUSS柜机上运行。只要是柜机上运行的航司应用程序组件且遵守此处列出的限制，航司应用程序提供商可以自由地实施他们选择的方法以及他们需要的工具和技术。

C10 平台技术

CUSS的目标是在不同的CUSS供应商和柜机之间维护CUSS航司应用程序的互操作性。CUSS平台本身不需要互操作-除了提供CUSS标准本身的完整和正确的实现。

因此，CUSS平台和柜机s能使用实现CUSS标准所需的任何工具、操作系统、框架或体系结构，并无限制地运行其平台服务、工具和其他产品，只要平台环境中没有任何内容与CUSS航司应用程序所需的标准、可互操作环境冲突（如上所列）。

C11 航司应用程序资源限制

确保单个CUSS航司应用程序不会影响共享通用柜机上的资源，以下是CUSS 1.3对CUSS航司应用程序施加的资源限制。考虑到一些较新工具和技术的需求增加，这些限制从以前的版本增加。‘

——在本地存储目录中，航司应用程序不得使用超过 1024MB。

——航司应用程序进程和所有子进程不能使用超过 192MB 的 RAM

——为允许航司应用程序在 java 进程中使用最多 192MB，平台必须将航司应用程序的 Java 堆大小限制设置为 256MB，或者在启动航司应用程序的命令行参数配置中，或者在全局 Web 浏览器中使用 Java 插件来设置所有航司应用程序。

——允许航司应用程序创建其他进程，但必须在完成后清理这些进程。

在事务处理或空闲时，航司应用程序不得过度使用CPU（或“挂起”）CPU。若在合理的评估下，平台提供商可以认为CPU使用率“过高”，则航司应用程序的CPU使用率会影响柜机上的其他服务、工具或航司应用程序正常和响应运行的能力。

附 录 D

（资料性附录）

扩展数据类型列表（DS_TYPES）

其中许多值也列在 datastatus 类的 codes.idl 中。D1 是扩展数据类型表。

表 D1 扩展数据类型表

标识符（DS_TYPES）	描述	数据格式	设置参数
DS_TYPES_ISO - 0	默认编码(护照/卡跟踪数据)	请参阅本文档中的相关章节	无
DS_TYPES_FOID_ISO - 100	ISO跟踪数据与FOID数据截获	请参阅本文档中的相关章节	
DS_TYPE_PAYMENT_ISO - 200	没有截获的ISO跟踪数据	请参阅本文档中的相关章节	要接受的IINs列表

DS_TYPES_DISCRETIONARY_ISO - 300	ISO跟踪数据与任意数据截获	请参阅本文档中的相关章节	要接受的IINs列表
DS_TYPES_FOID_JIS2 - 14100	JIS-2跟踪数据与FOID数据截获	请参阅本文档中的相关章节	
DS_TYPES_PAYMENT_JIS2 - 14200	没有截获的JIS-2跟踪数据	请参阅本文档中的相关章节	要接受的IINs列表
DS_TYPES_DISCRETIONARY_JIS2 -14300	JIS-2跟踪数据与任意数据截获	请参阅本文档中的相关章节	要接受的IINs列表
DS_TYPES_IMAGE_IR - 7000	红外生物特征识别或文件图像	行业标准JPG或BMP	“JPG”（默认）或“ BMP”
DS_TYPES_IMAGE_VIS - 8000	可见的生物特征识别或文件图像	行业标准JPG或BMP	“JPG”（默认）或“ BMP”
DS_TYPES_IMAGE_UV - 9000	紫外线生物测定或文件图像	行业标准JPG或BMP	“JPG”（默认）或“ BMP”
DS_TYPES_IMAGE_PHOTO - 10000	摄影生物测定或文件图像	行业标准JPG或BMP	“JPG”（默认）或“ BMP”
DS_TYPES_IMAGE_COAX - 11000	同轴生物测定或文件图像	行业标准JPG或BMP	“JPG”（默认）或“ BMP”
DS_TYPES_CODELINE - 12000	在文档中检测到OCR数据并进行解码	ASCII文本	无
DS_TYPES_BARCODE - 13000	在文档中检测到条形码数据并进行解码	ASCII文本或二进制流	无
DS_TYPES_MIWA - 14000[49]	MIWA磁卡编码的密码锁数据格式	查看MIWA规范文档	无
DS_TYPES_JIS2 - 14000	日本工业标准卡编码类型JIS2	ASCII文本，单磁道	无
DS_TYPES_SCAN_PDF417 - 15000	扫描仪/CCD支持的PDF417编码数据	二进制数据，多磁道	无
DS_TYPES_SCAN_AZTEC - 15100	扫描仪/CCD支持的Aztec编码数据	二进制数据，多磁道	无
DS_TYPES_SCAN_DMATRIX - 15200	扫描仪/CCD支持的数据矩阵编码数据	二进制数据，多磁道	无
DS_TYPES_SCAN_QR - 15300	扫描仪/CCD支持的QR编码数据	二进制数据，多磁道	无
DS_TYPES_SCAN_CODE39 - 15400	扫描仪/CCD支持code 3 of 9	文本数据，多磁道	无

	的一维条形码		
DS_TYPES_SCAN_CODE128 - 15500	扫描仪/CCD支持code 128的一维条形码	文本数据，多磁道	无
DS_TYPES_SCAN_CODE2OF5 - 15600	扫描仪/CCD支持Interleaved 2 of 5的一维条形码	文本数据，多磁道	无
DS_TYPES_PRINT_2S_PAGE - 16100	GPP支持背面打印单个页面	请参阅本文档中的相关章节	参见6.4.3节
DS_TYPES_PRINT_2S_MULTI - 16200	GPP支持多页文档的前后打印	请参阅本文档中的相关章节	参见6.4.3节
DS_TYPES_PRINT_PDF - 16300	PDF版本7.0 / ISO32000兼容的打印数据	请参阅本文档中的相关章节	参见6.4.2节
DS_TYPES_MIFARE - 17000	PICC/NFC/RFID设备的通信协议	请参阅本文档中的相关章节	参见7.15节
DS_TYPES_ISO15961 - 18000	IATA RFID行李标签装置	参考CUSS.SBD.XSD	
DS_TYPES_RP1745 - 18010	IATA行李服务信息格式	参考CUSS.SBD.XSD	
DS_TYPES_WEIGHT - 18020	来自秤或SBD设备的行李重量	参考CUSS.SBD.XSD	
DS_TYPES_HEAVYTAG - 18030	表明指定为行李重标签打印机的打印机	无	
DS_TYPES_SBDAEA - 18040	AEA-SBD控制语言	参考AEA2012-2	
DS_TYPES_SBDCUSS - 18050	CUSS-SBD控制语言		
DS_TYPES_EPASSPORT_DG1 - 20100	ICAO电子护照RFID数据DG1	二进制数据-参见ICAO规范	无
DS_TYPES_EPASSPORT_DG2 - 20200	ICAO电子护照RFID数据DG2[…]	二进制数据-参见ICAO规范	无
...DG3-DG20 - 20300-2200	ICAO电子护照RFID数据DG20	二进制数据-参见ICAO规范	无
DS_TYPES_EPAYMENT - 2300	通用支付和EMV交易CUSS 1.3格式	参考CUSS.PAYMENT.XSD	