Point of Sale Subsystem

UnifiedPOS

Retail Peripheral Architecture
Toshiba Defined Management Services

Version 1.14.2

Table of Contents

1 Installation and Configuration

The following sections provide details regarding the installation and configuration of Toshiba UnifiedPOS system management support on the following environments

- Windows

For Linux installation refer to Toshiba JavaPOS for Linux Installation Instructions.pdf

Notes:

1. The system management support is provided for all JavaPOS/OPOS supported devices.

1.1 Installation on Windows

During Toshiba UnifiedPOS installation, a check box option is provided for selecting system management support. When this option is selected, the installation automatically installs and configures necessary system management components. By default, this option is enabled. However, the system management has dependencies on the existence of Windows core components that support system management.

Windows WMI Component

The Microsoft WMI components are required to run the Toshiba UnifiedPOS Management Services on Windows. The WMI component is typically a part of Windows OS, and Toshiba UPOS drivers installation does not install Microsoft WMI components during installation. In case it is missing, it can be downloaded from:

http://www.microsoft.com/downloads/details.aspx?familyid=013BB284-3946-44A9-AC3C-BF2A569EAA72&displaylang=en

Optionally, the WMI components can be obtained by installing the Microsoft .NET component.

Validating System management:

To view the systems management properties for a device, the device must be opened and claimed.

- **open/claim/enable a device in JavaPOS:**

  Start POS Control Center utility:

  a. Start → Toshiba UnifiedPOS for Windows → JavaPOS → POS Control Center
  b. To configure devices, Click on AutoDetect and save jpos.xml to default location.
  c. Select a device that is online, and click on "System Management" tab, click on "Start Statistics Test". This will display the system management properties of the device, and it will also keep the device in open/claim/enable state.

- **open/claim/enable a device in OPOS:**

  Use the application or an OPOS utility to open/claim/enable a device(s).

- **Use wbemtest utility to view properties:**

  Start wbemtest tool

  a. Click on Start → Run, then enter wbemtest
  b. Connect to root\cimv2 name space
  c. Click on Connect
  d. Enter root\cimv2, click on Connect again

  Enumerate UPOS Device Instances

  a. Click on Enum Instances
  b. Enter UPOS_LogicalDevice
  c. Select Recursive
  d. Click on OK
  e. To view detailed properties, select a device in Query Results windows and double click on it.

1.2 Uninstallation on Windows

When the Toshiba UnifiedPOS software is uninstalled, it will automatically remove the system management components .

1.3 System Management Configuration File

To customize some of the system management functions, several properties are defined in sysmgmt.properties file. The details are described below.

File Name: systemgmt.properties

Location : <install directory>\sysmgmt directory (Windows).

/opt/tgcs/javapos/etc (Linux)

Property: provider.eventSocket.Port

Default value: 42114

Description: Port number used by Windows Event CIM Provider (JavaPOS).

Description: Port number used by Windows CIM Provider for events and requests (OPOS).

Property: provider.response.timeout

Default value: 30000

Description: Timeout value from CIM Provider to JavaPOS Management Services..

Property: upos.requestSocket.Port

Default value: 42115

Description: Port number used by JavaPOS Management Services.

Property: provider.eventSocket.IP

Default value: 127.0.0.1

Description: Destination IP address for event UDP datagrams. Typically this would be the local machine,

Property: provider.maxQueryThreads

Default value: 10

Description: Maximum number of threads that are allowed to connect to the Java drivers at the same time.

Property: provider.dataUpdate.frequency

Default value: 1

Description: Defines the value of the frequency in which the driver should try to update the data of the device, it will depend if the device is use or not, the driver will decide if it is possible to update the data at that moment, if not, it should ignore the request or enqueue it until it can be executed. This value is configured in minutes.

2 Problem Determination

The Toshiba UPOS provides facility to gather trace information for JavaPOS/OPOS and Provider components. You can selectively enable/disable traces as follows.

Enabling Trace in Linux

1. Edit the file `/opt/tgcs/javapos/etc/jutil.properties`
2. Turn on the "com.ibm.jutil.tracing.TurnOnAllNamedTracers=ON" trace.
3. The location for the trace file is “<HOME>/.tgcsjpos” where <HOME> is the absolute path to the user’s home directory.

To enable the provider tracing:

1. Edit/Create the file /var/pos/aiptrace.cfg.
2. Enable trace with the option " SystemManagement: ON". Also configure the following settings to a correct value:

   BackupCount: <Number from 0 to 4>

   FileSize: <Maximum Length of file>

   A value of 0 on FileSize will not have a limit on the file to be created.
3. Restart the CIM Server for changes to take effect:

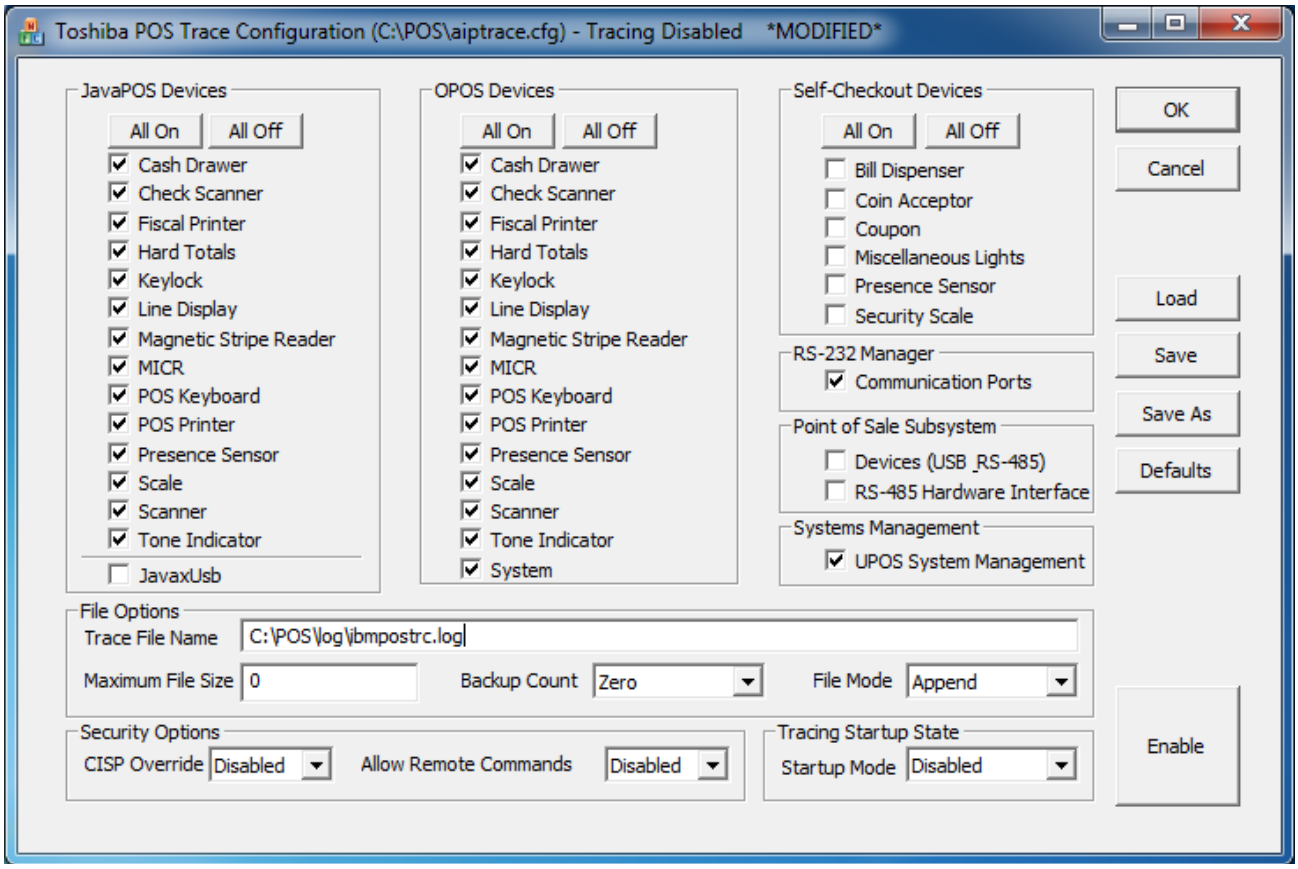
   ```
   /etc/init.d/sfcb restart
   ```
4. The location for the trace file is “/var/log/UPOS_SysMgmt.log”.

Note: If no configuration file is found, trace will be disabled.

Enabling Trace in Windows

For Toshiba UPOS Systems Management, the following steps are used to gather trace information.

1 Using a command line prompt, type in AIPTRCCFG. The application will appear



2 Choose item `UPOS Systems Management` for tracing.

3 When trace configuration is complete, click Save and then click Enable.

Notes:

2 different trace files are created:

- The first file is indicated in the option `Trace File Name`. The file will be created based on the settings selected in the Toshiba POS Trace Configuration window.
- The second one is the UPOS system management provider trace and is created in the directory <INSTALL_DIR>\log\UPOS_SysMgmt.log.

  This trace file will be created based on the settings for Maximum File Size and Backup Count only; the other options will not take effect for this trace.

3 Toshiba Defined Management Services

3.1 Introduction

This document explains the high level design of the UnifiedPOS Management Services Subsystem and related components. This strategy conforms to the Common Information Model (CIM) from the Distributed Management Task Force (DMTF). The CIM model for Retail devices has been included as part of UnifiedPOS Specification starting in version 1.12. The Toshiba management service for Retail devices is based off the CIM schema for Retail devices.

3.2 CIM ClassNames for UnifiedPOS Device Category Names

The correlations of UnifiedPOS programmatic names and CIM class names are defined in the following table

UnifiedPOS Device Programmatic Names	CIM Class Name	OPOS Supported since	JPOS Supported since
Belt	UPOS_Belt		
BillAcceptor	UPOS_BillAcceptor		
BillDispenser	UPOS_BillDispenser		
Biometrics	UPOS_Biometrics		
BumpBar	UPOS_BumpBar		
CashChanger	UPOS_CashChanger		
CashDrawer	UPOS_CashDrawer	1.13.0	1.9.1
CAT	UPOS_CAT		
CheckScanner	UPOS_CheckScanner	1.13.0	1.9.1
CoinAcceptor	UPOS_CoinAcceptor		
CoinDispenser	UPOS_CoinDispenser		
ElectronicJournal	UPOS_ElectronicJournal		
ElectronicValueRW	UPOS_ ElectronicValueRW		
FiscalPrinter	UPOS_FiscalPrinter	1.13.0	
Gate	UPOS_Gate		
HardTotals	UPOS_HardTotals	1.13.0	1.9.1
ImageScanner	UPOS_ImageScanner		
ItemDispenser	UPOS_ItemDispenser		
Keylock	UPOS_Keylock	1.13.0	1.9.1
Lights	UPOS_Lights		
LineDisplay	UPOS_LineDisplay	1.13.0	1.9.1
MICR	UPOS_MICR	1.13.0	1.9.1
MotionSensor	UPOS_MotionSensor	1.13.0	1.9.1
MSR	UPOS_MSR	1.13.0	1.9.1
PINPad	UPOS_PINPad		
PointCardRW	UPOS_PointCardRW		
POSKeyboard	UPOS_POSKeyboard	1.13.0	1.9.1
POSPower	UPOS_POSPower		
POSPrinter	UPOS_POSPrinter	1.13.0	1.9.1

UnifiedPOS Device Programmatic Names	CIM Class Name	OPOS Supported since	JPOS Supported since
RemoteOrderDisplay	UPOS_RemoteOrderDisplay		
RFIDScanner	UPOS_RFIDScanner		
Scale	UPOS_Scale	1.13.0	1.9.1
Scanner	UPOS_Scanner	1.13.0	1.9.1
SignatureCapture	UPOS_SignatureCapture		
SmartCardRW	UPOS_SmartCardRW		
ToneIndicator	UPOS_ToneIndicator	1.13.0	1.9.1

Note: There are some differences between the UPOS Specification 1.12 definition and the class names listed in this table. For a complete class definitions refer to:

In Windows: File located in "<installdir>\sysmgmt\UPOSMgmtSrv.mof"

In Linux using Pegasus: File located in /usr/share/Pegasus/mof/Pegasus

In SLED using SFCB: File located in /usr/share/sfcb

3.3 JavaPOS Changes

To seamlessly support the integration of UnifiedPOS management services, some changes are required to the UnifiedPOS specification, as well as the device controls provided by members of the committee.

Each component, the control and the service will have the capability to expose the device to UnifiedPOS Management Services. A read/write Boolean property, **AllowManagement** at control, will allow the application to determine if the device should participate in systems management. The default value is true for **AllowManagement** property. Also, each Service should implement the **UPOSManagementService** interface. The interface, from UPOS Management Services, it is what the component passes to Management Services when it registers with it. This interface serves as the connection point to Management Services and eventually the CIMOM. Registration occurs when the device is opened, and un-register when closed.

When the device is open, it should check with its corresponding control and service to determine if it will handle systems management, checking the **CapServiceAllowManagement** capability and **AllowManagement** property. If **CapServiceAllowManagement** is true, the Service will accept the responsibility to interact with systems management, notify StatusUpdateEvents and register its own **UPOSManagementService** with UPOS Management Services and handle the systems management interface on behalf of the named device.

During open time when **CapStatisticsReporting** is true the **UPOSManagementService** will try to claim and enable the device to call the retrieveStatistics method and gather the device information statistics like SerialNumber. Once the retrieveStatistics method is complete the device will be released and the control returned to the application.

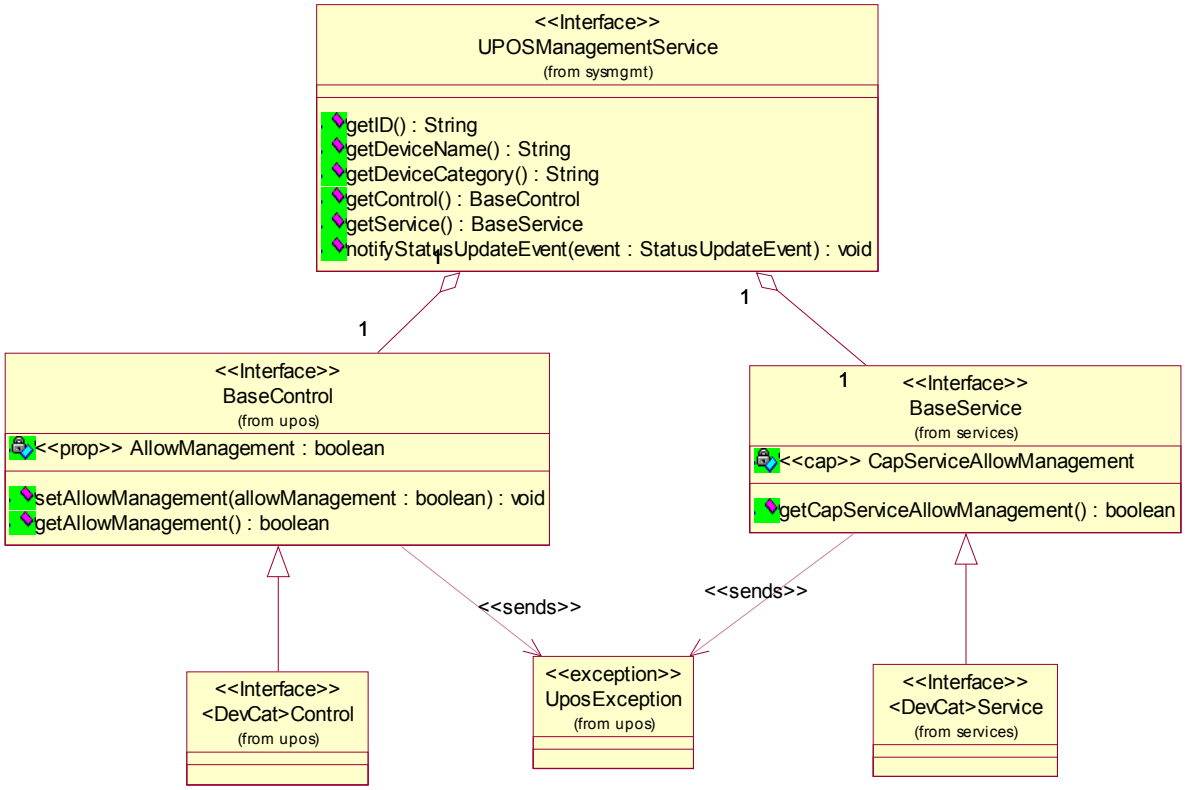
When **CapServiceAllowManagement** is false and the **AllowManagement** is true the control of the device will register the **UPOSManagementService** with UPOS Management Services. Finally when both are false the device will not participate in systems management.

The **UPOSManagementService** is unregistered when the device is closed.

CapServiceAllowManagement capability has as default value false

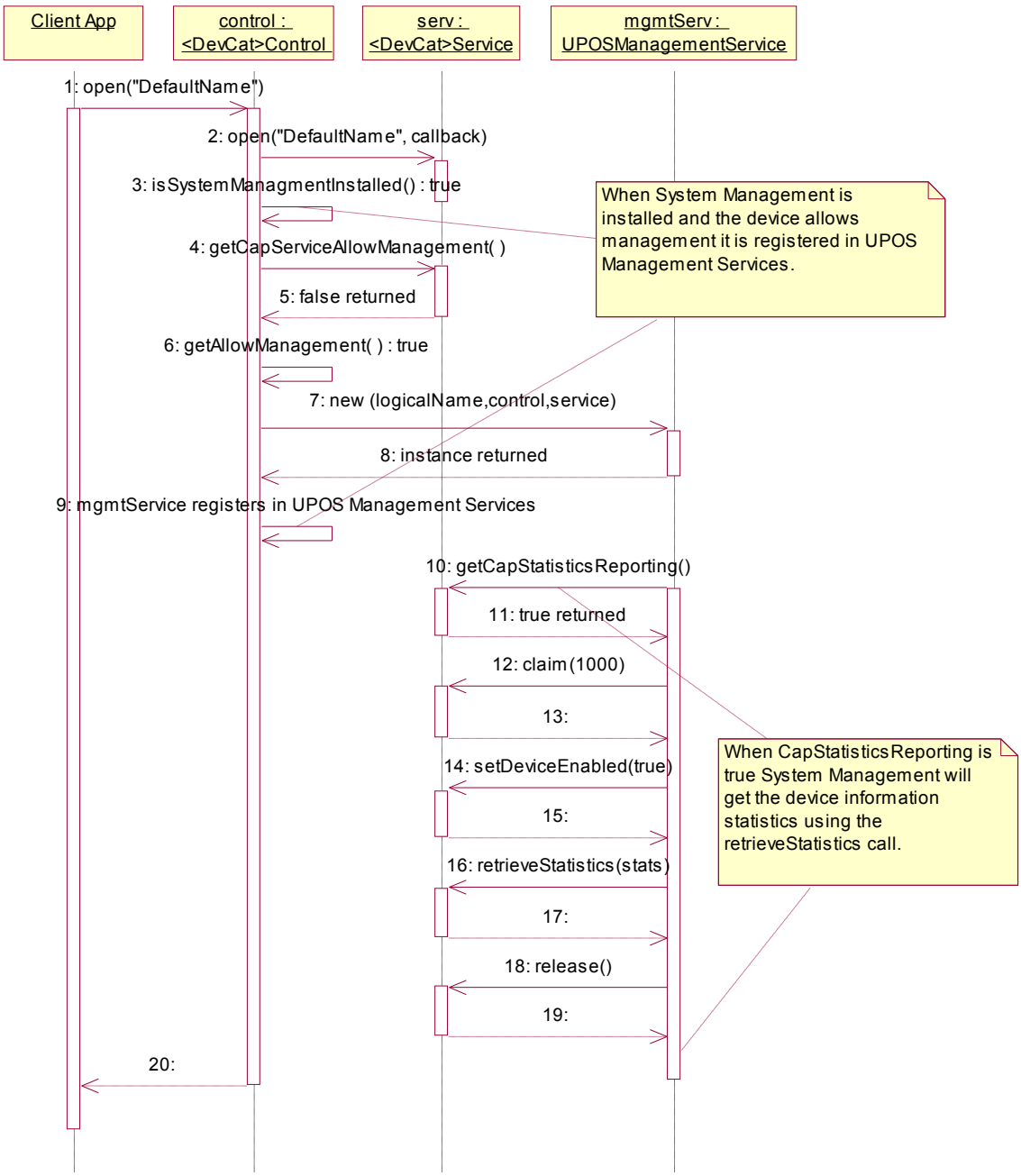
The following sections describe the class diagrams and sequence diagrams created in Toshiba UnifiedPOS 1.9.1 release.

3.2.1 JavaPOS Controls Class Diagram for Management Service

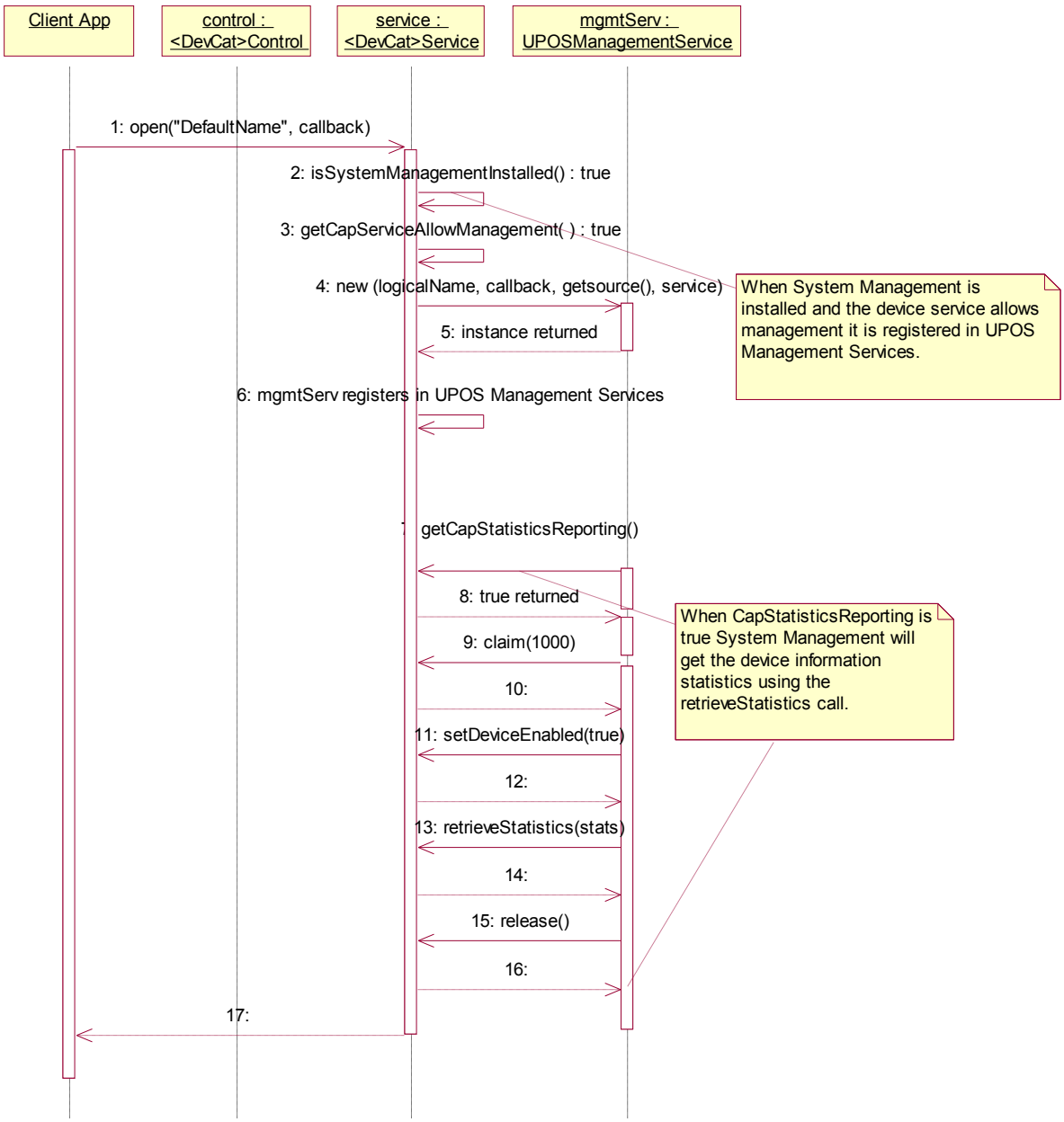


3.2.2 JavaPOS Device Registration Sequence Diagram (CapServiceAllowManagement = false)

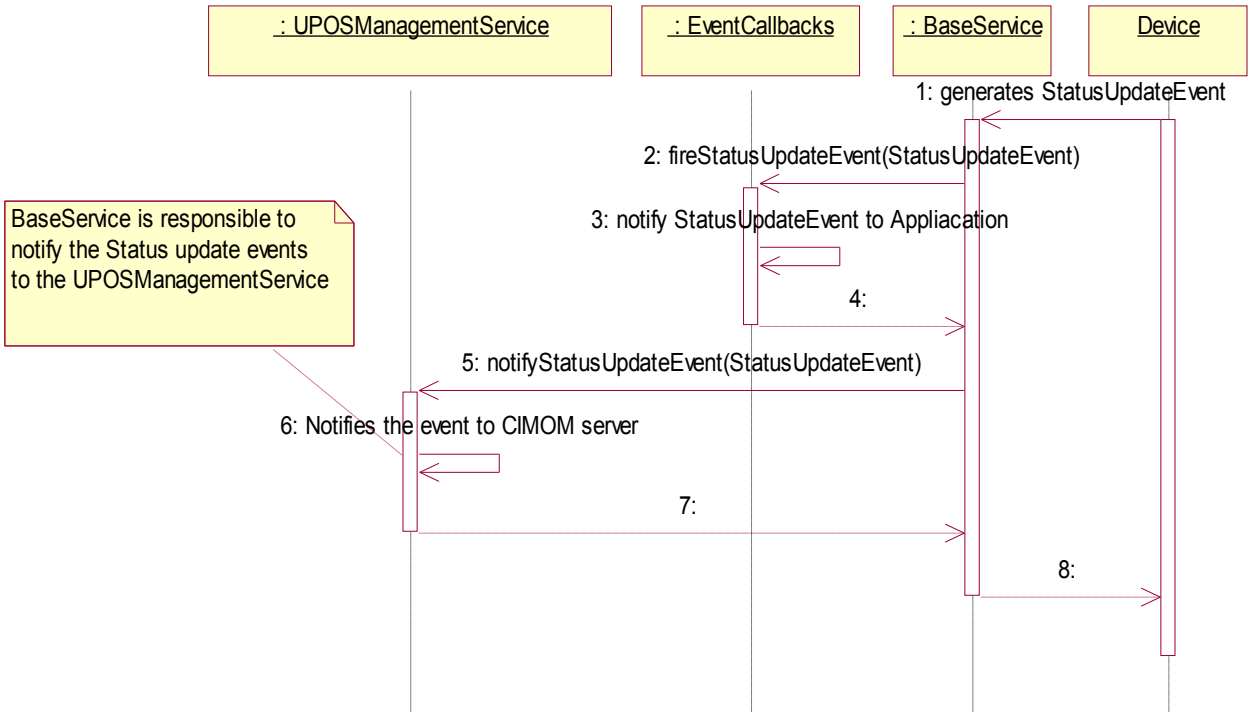
The following sequence diagram show the new sequences added to the Device Control to register the device with Toshiba Management Services at open() time.



3.2.3 JavaPOS Device Registration sequence diagram (CapServiceAllowManagement = true)



3.2.4 JavaPOS Events notification sequence diagram (CapServiceAllowManagement = true)



Toshiba UnifiedPOS Provider for System Management

The Toshiba Provider acts as driver and interface between the abstract world of the Common Information Model (CIM) and the UnifiedPOS device characteristics of Retail Hardware.

Following describes the providers supported by Toshiba:

3.4.1 Instance Provider

An instance provider supplies instances of one or more given classes. For example, an instance provider can supply information regarding a POSPrinter device.

The information provided at CIM getInstance call will depend on the current state of the Device, when the **DeviceEnabled** property is false the Provider will generate an instance with all the properties available at open time plus all properties defined by UPOS statistics (device information and device specific properties).

Note: When device is not enabled the initial statistic properties detected during open time are used.

When the **DeviceEnabled** property is true the Provider will generate an instance with all the properties defined at the UPOS CIM Class.

CIM Method	WMI Equivalent	Supported by		
		Windows	IRES 2	SLED 11
GetInstance	GetObjectAsync	Yes	Yes	Yes
ModifyInstance	PutInstanceAsync	No	No	No
DeleteInstance	DeleteInstanceAsync	No	No	No
EnumerateInstances	CreateInstanceEnumAsync	Yes	Yes	Yes
EnumerateInstanceNames		No	Yes	Yes
ExecQuery	ExecQueryAsync	No	No	No

3.4.2 Method Provider

A method provider allows CIMOM access to the methods of a class.

CIM Method	WMI Equivalent	Supported by		
		Windows	IRES 2	SLED 11
InvokeMethod	ExecMethodAsync	No	No	No

3.4.3 UPOS System Management Events in Windows

```
class UPOS_SysMgmtEvent : __ExtrinsicEvent
{
   String eventtype;
   String classname;
   String deviceid;
   String codename;
} ;
```

3.4.4 UPOS System Management Events in Linux

```
class UPOS_SysMgmtEvent: CIM_ProcessIndication
{
   string eventtype;
   string classname;
   string deviceid;
   string codename;
};
```

3.4.5 Description of Event fields

eventtype property

The type of event generated

Values	Meaning
addInstance	An UPOS device has registered with UPOS System Management
modifyInstance	The UPOS device has generated an Status Update Event
delInstance	An UPOS device has unregistered with UPOS System Management

classname property

The CIM class name. Refer to section 1.5 in “CIM Class name” column for values.

deviceid property

The ID whom generates the event, currently this ID is created in base of the logicalName and CIM class name.

For example:

“POSPrinter1UPOS_POSPrinter”

codename property

Property used when the event type is “modifyInstance”. The value describes the type of device category-specific status change. Below the details for supported categories.

UPOS Device Category	Status Update Events (SUE) reported to System Management
Common (all devices)	JPOS_SUE_POWER_ONLINE = 2001
	OPOS_SUE_POWER_ONLINE = 2002
	JPOS_SUE_POWER_OFF = 2002
	OPOS_SUE_POWER_OFF = 2002
	JPOS_SUE_POWER_OFFLINE = 2003
	OPOS_SUE_POWER_OFFLINE = 2003
	JPOS_SUE_POWER_OFF_OFFLINE = 2004
	OPOS_SUE_POWER_OFF_OFFLINE = 2004
	JPOS_SUE_UF_PROGRESS = 2100
	OPOS_SUE_UF_PROGRESS = 2100
	JPOS_SUE_UF_COMPLETE = 2200
	OPOS_SUE_UF_COMPLETE = 2200
	JPOS_SUE_UF_FAILED_DEV_OK = 2201
	OPOS_SUE_UF_FAILED_DEV_OK = 2201
	JPOS_SUE_UF_FAILED_DEV_UNRECOVERABLE = 2202
	OPOS_SUE_UF_FAILED_DEV_UNRECOVERABLE = 2202
	JPOS_SUE_UF_FAILED_DEV_NEEDS_FIRMWARE = 2203
	OPOS_SUE_UF_FAILED_DEV_NEEDS_FIRMWARE = 2203
	JPOS_SUE_UF_FAILED_DEV_UNKNOWN = 2204
	OPOS_SUE_UF_FAILED_DEV_UNKNOWN = 2204
	JPOS_SUE_UF_COMPLETE_DEV_NOT_RESTORED = 2205
	OPOS_SUE_UF_COMPLETE_DEV_NOT_RESTORED = 2205
CashDrawer	CASH_SUE_DRAWERCLOSED = 0
	CASH_SUE_DRAWEROPEN = 1
CheckScanner	CHK_SUE_SCANCOMPLETE = 11

UPOS Device Category	Status Update Events (SUE) reported to System Management	
FiscalPrinter	FPTR_SUE_COVER_OPEN	= 11
	FPTR_SUE_COVER_OK	= 12
	FPTR_SUE_JRN_EMPTY	= 21
	FPTR_SUE_JRN_NEAREMPTY	= 22
	FPTR_SUE_JRN_PAPEROK	= 23
	FPTR_SUE_REC_EMPTY	= 24
	FPTR_SUE_REC_NEAREMPTY	= 25
	FPTR_SUE_REC_PAPEROK	= 26
	FPTR_SUE_SLP_EMPTY	= 27
	FPTR_SUE_SLP_NEAREMPTY	= 28
	FPTR_SUE_SLP_PAPEROK	= 29
	FPTR_SUE_IDLE	=1001
	FPTR_SUE_JRN_COVER_OPEN	= 60
	FPTR_SUE_JRN_COVER_OK	= 61
	FPTR_SUE_REC_COVER_OPEN	= 62
	FPTR_SUE_REC_COVER_OK	= 63
	FPTR_SUE_SLP_COVER_OPEN	= 64
	FPTR_SUE_SLP_COVER_OK	= 65
Keylock	LOCK_KP_ELECTRONIC	= 0
	LOCK_KP_LOCK	= 1
	LOCK_KP_NORM	= 2
	LOCK_KP_SUPR	= 3
MotionSensor	MOTION_M_PRESENT	= 1
	MOTION_M_ABSENT	= 2

UPOS Device Category	Status Update Events (SUE) reported to System Management	
POSPrinter	PTR_SUE_COVER_OPEN	= 11
	PTR_SUE_COVER_OK	= 12
	PTR_SUE_JRN_EMPTY	= 21
	PTR_SUE_JRN_NEAREMPTY	= 22
	PTR_SUE_JRN_PAPEROK	= 23
	PTR_SUE_REC_EMPTY	= 24
	PTR_SUE_REC_NEAREMPTY	= 25
	PTR_SUE_REC_PAPEROK	= 26
	PTR_SUE_SLP_EMPTY	= 27
	PTR_SUE_SLP_NEAREMPTY	= 28
	PTR_SUE_SLP_PAPEROK	= 29
	PTR_SUE_JRN_CARTRIDGE_EMPTY	= 41
	PTR_SUE_JRN_CARTRIDGE_NEAREMPTY	= 42
	PTR_SUE_JRN_HEAD_CLEANING	= 43
	PTR_SUE_JRN_CARTDRIGE_OK	= 44
	PTR_SUE_JRN_CARTRIDGE_OK	= 44
	PTR_SUE_REC_CARTRIDGE_EMPTY	= 45
	PTR_SUE_REC_CARTRIDGE_NEAREMPTY	= 46
	PTR_SUE_REC_HEAD_CLEANING	= 47
	PTR_SUE_REC_CARTDRIGE_OK	= 48
	PTR_SUE_REC_CARTRIDGE_OK	= 48
	PTR_SUE_SLP_CARTRIDGE_EMPTY	= 49
	PTR_SUE_SLP_CARTRIDGE_NEAREMPTY	= 50
	PTR_SUE_SLP_HEAD_CLEANING	= 51
	PTR_SUE_SLP_CARTRIDGE_OK	= 52
	PTR_SUE_IDLE	= 1001
	PTR_SUE_JRN_COVER_OPEN	= 60
	PTR_SUE_JRN_COVER_OK	= 61
	PTR_SUE_REC_COVER_OPEN	= 62
	PTR_SUE_REC_COVER_OK	= 63
	PTR_SUE_SLP_COVER_OPEN	= 64
	PTR_SUE_SLP_COVER_OK	= 65
	JPOS specific:	
	IBM_JPOS_SUE_PTR_REC_UNEXPECTED_COVER_OPEN	= 10000;
	IBM_JPOS_SUE_PTR_SLP_UNEXPECTED_COVER_OPEN	= 10001;
	IBM_JPOS_SUE_PTR_MAIN_LOGIC_CARD_FAILURE	= 10002;
	IBM_JPOS_SUE_PTR_INTERFACE_LOGIC_CARD_FAILURE	= 10003;
	IBM_JPOS_SUE_PTR_REC_PRINT_HEAD_FAILURE	= 10004;
	IBM_JPOS_SUE_PTR_SLP_PRINT_HEAD_FAILURE	= 10005;
	IBM_JPOS_SUE_PTR_PAPER_MOTION_SENSOR_FAILURE	= 10006;
	IBM_JPOS_SUE_PTR_REC_CRITICALLY_LOW_PAPER	= 10007;
	IBM_JPOS_SUE_PTR_REC_PRINT_HEAD_OVERHEAT	= 10008;
	IBM_JPOS_SUE_PTR_REC_PRINT_HEAD_OK	= 10009;

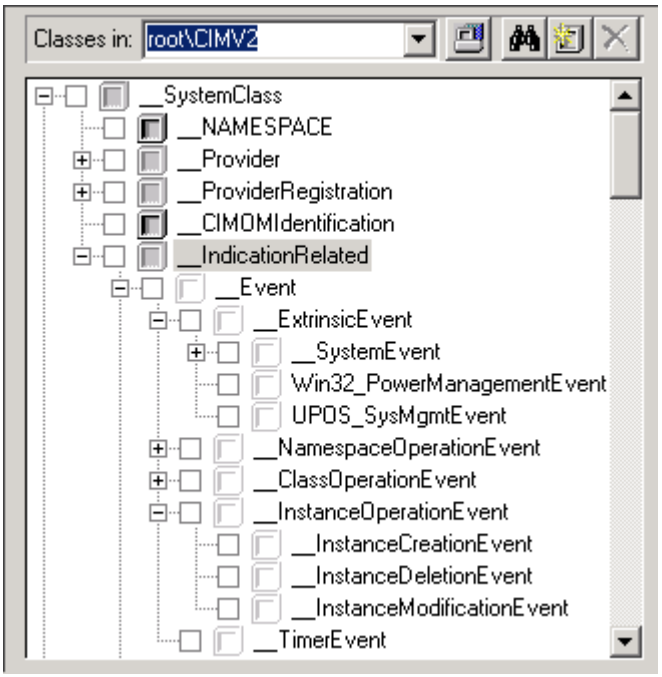
UPOS Device Category	Status Update Events (SUE) reported to System Management
Scale	SCL_SUE_STABLE_WEIGHT = 11
	SCAL_SUE_STABLE_WEIGHT = 11
	SCL_SUE_WEIGHT_UNSTABLE = 12
	SCAL_SUE_WEIGHT_UNSTABLE = 12
	SCL_SUE_WEIGHT_ZERO = 13
	SCAL_SUE_WEIGHT_ZERO = 13
	SCL_SUE_WEIGHT_OVERWEIGHT = 14
	SCAL_SUE_WEIGHT_OVERWEIGHT = 14
	SCL_SUE_NOT_READY = 15
	SCAL_SUE_NOT_READY = 15
	SCL_SUE_WEIGHT_UNDER_ZERO = 16
	SCAL_SUE_WEIGHT_UNDER_ZERO = 16

3.4.6 Windows Event Provider (WMI)

An event provider is a COM object that supplies WMI notifications of intrinsic and extrinsic events. An intrinsic event reports an internal data change to WMI, while an extrinsic event reports a user-defined event not described by an intrinsic event.

For example, an event in response to changes, creation, or deletion of the UnifiedPOS_POSPrinter class would classify as an intrinsic event. An event that is generated on the basis of something other than the modification, creation or deletion of an existing WMI object is an extrinsic event.

Toshiba UnifiedPOS provider provides events for creation, deletion and modification of instances (all the supported UnifiedPOS Status Update Events(SUE) are reported as instance modification events). Even though these events could be classified as intrinsic events, they are implemented as extrinsic events, since they are instances of UPOS_SysMgmtEvent, which is subclass of __ExtrinsicEvent (See figure below). Intrinsic events are generated by the WMI, rather than by the provider. Intrinsic events that concern to UPOS System Management are __InstanceOperationEvent and its subclasses. In order to receive intrinsic events, as a client application, it is necessary to subscribe a consumer with WMI for the intrinsic events specific to the CIM classes that are intended to monitor.



Detail of class hierarchy around __Event system class.

(This class hierarchy is shown using CIM Studio from the WMI Tools, see reference ahead)

3.4.7 Linux Indication Provider

An indication is the representation of the occurrence of an event. The indication provider identifies when a specific type of event happens in the system. Then it converts the event into a CIM_ProcessIndication and sends it to the CIM Server.

In order to be ready to receive those events the provider should be registered (this step is performed at installation time) and a client application should be developed to subscribe for specific indications. The following steps are required in the client application:

- Define an indication filter condition, to describe the event that should be monitored, for example, when an instance is created.
- Define an indication listener to specify how to handle the indication.
- Activate the subscription by associating a filter and a listener.
- Consume the indication when it arrives based on the configured filter and it will be handled in the registered listener.

The API JSR48 Java™ WBEM Services is a set of APIs for Web-Based Enterprise Management. It has implemented several functions that are useful to code a client application.

http://sblim.wiki.sourceforge.net/CimClient

4 References

Documents referenced and utilized for the implementation of UnifiedPOS Management Services:

- UnifiedPOS Retail Peripheral Architecture Version 1.9 http://www.nrf-arts.org/
- Common Information Model Version 2.2 http://www.dmtf.org/standards/cim
- Common Information Model Schema 2.9 http://www.dmtf.org/standards/cim/cim_schema_v29
- Java WBEM Services 1.0 API http://wbemservices.sourceforge.net/javadoc/api/index.html
- JSR 48: WBEM Services Specification http://jcp.org/en/jsr/detail?id=48
- SBLIM Project http://sblim.sourceforge.net/index.html
- Pegasus site http://www.openpegasus.org
- CIM Schema for Retail Devices: CIM-UPOS(6).pdf http://www.nrf-arts.org/
- SFCB site http://sblim.wiki.sourceforge.net/Sfcb
- wbemcli site http://sblim.wiki.sourceforge.net/Wbemcli