Point of Sale Subsystem

UnifiedPOS

Retail Peripheral Architecture
IBM Defined Management Services

Version 1.12
April 2009

Table of Contents

1.0 Installation and Configuration

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

- Windows
- SUSE Linux Enterprise Desktop V11

Notes:

1. The system management support is provided for all JavaPOS supported devices.
2. For OPOS, System Management support is provided for the following devices:
   - 4610 POSPrinter (Printer, MICR, CheckScanner, Tone Indicator, Cash Drawer)
   - AnyPlace Kiosk Scanners, MSRs, and the Motion Sensor devices

3.1 Installation on Windows

During IBM 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 IBM UnifiedPOS Management Services on Windows. The WMI component is typically a part of Windows OS, and IBM 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.

Installation on Windows 2000:

On Windows 2000, it is highly recommended that the WMI component is properly installed from one of the above methods. Otherwise you may see some failures when you try to enumerate POS devices in WMI due to missing dlls.

Validating System management:

The WMI Administrative Tools can be used to verify the system management properties of Retail Devices. The tool can be downloaded from.

http://www.microsoft.com/downloads/details.aspx?FamilyID=6430f853-1120-48db-8cc5-f2abdc3ed314

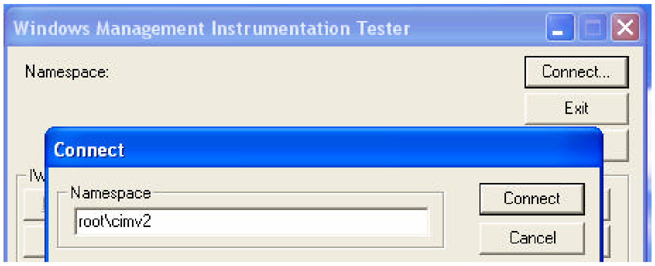
1. Install the WMI tool
2. Open the supported device(s)
3. Launch WMI Tool: Start -> WMI CIM Studio
4. In Connect to namespace, enter “root\cimv2”, and log into CIM Studio
5. To view the properties, select the device on left pane, under UPOS_LogicalDevices

3.2 Uninstallation on Windows

When the IBM UnifiedPOS software is uninstalled, it will automatically remove the system management components on Windows XP. However on Windows 2000, some manual clean is required as described below.

Uninstallation on Windows 2000

IBM UnifiedPOS installer uses the “wmic” utility to delete the mof classes from WMI. Since this utility does not exist on Windows 2000, the mof classes must be manually deleted.



Namespace: root\cimv2

1.- Press Start-> run menu item.

2.- Type “wbemtest” at open field and press the “OK” button.

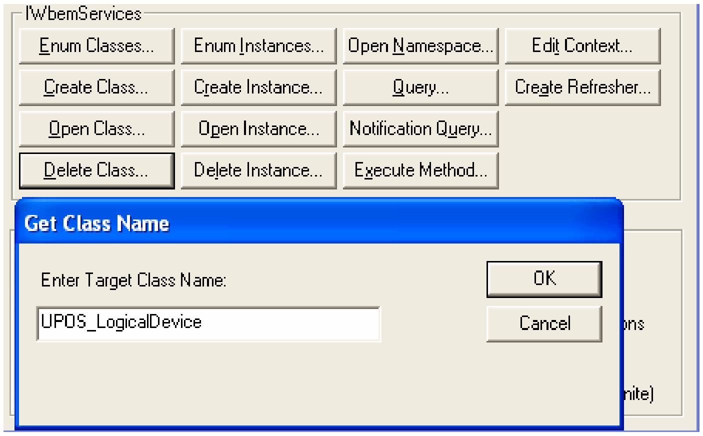
At “Windows Management Instrumentation Tester” window :

3.- Click on “connect…” button.

At Connect window:

4.- type “root\cimv2” at Namespace label.

5.- Click “connect” button.



Enter Target Class Name: UPOS_LogicalDevice

6. Click on “Delete Class…”

At “Get Class Name” window

7.- Type “UPOS_LogicalDevice” and click “OK” button.

8.- This will delete the parent “UPOS_LogicalDevice” class and all its children classes from WMI.

9.- Repeat 6 and 7 steps for “UPOS_DecoupledProvider” and “UPOS_SysMgmtEvent” classes.

3.3 Installation on Linux

The IBM UnifiedPOS support for System Management comprises of the following components:

rpms:

posIBM_sblim-cmpi-upos-server-<platform>-<version>.i586.rpm

posIBM_XML4C-5.4.6.i586.rpm

jar files:

jpos1122.jar:

jpos_sysmgmt.jar

These files will be located in /opt/ibm/javapos/lib directory after installing the ibm-javapos-<version>.i386.rpm. Having this file on the system will enable the Systems Management capabilities of the devices.

Running on SLED\SLES 11

In this release the supported version is SFCB 1.3.3 (Small Footprint CIM Broker). You can obtain this version from Novell's website.

Note that the system management support on SLED\SLES is provided on use as-is basis.

Pre-requisites:

- Small Footprint CIM Broker (SFCB) 1.3.3 must be installed.
- SFCB settings must be configured correctly to allow http communication and authentication. The configuration file is located in the following path:

  `/etc/sfcb/sfcb.cfg`
- A standardized DMTF CIM Schema, which is installed and configured when installing SFCB.
- A working IBM JavaPOS configuration. In other words, the application should be able to open UPOS devices.
- wbemcli can be used to verify the system management properties of Retail Devices. The tool can be downloaded from:

  `http://sblim.wiki.sourceforge.net/Wbemcli`

Note: For instructions about the different configuration for SFCB refer to the home page:

`http://sblim.wiki.sourceforge.net/Sfcb`

Installation and Configuration:

You must be root user to install and configure the system management components.

- Install posIBM_XML4C-5.4.6-1.i586.rpm.
    - `rpm -ivh posIBM_XML4C-5.7.1-1.i586.rpm`
- Install “posIBM_sblim-cmpi-upos-server-sled-<version>.i586.rpm”.
    - `rpm -ivh posIBM_sblim-cmpi-upos-server-sled-<version>.i586.rpm`
- Open one or more devices using a JavaPOS application.
- Get System Management properties for a given device by issuing the following command:
    - `wbemcli ei http://localhost:5988/root/cimv2:UPOS_POSPrinter`

3.4 Removing UPOS Systems Management

- Remove “posIBM_sblim-cmpi-upos-server-<platform>-<version>.i586.rpm” .
    - `rpm -e posIBM_sblim-cmpi-upos-server-<platform>-<version>`
- Optionally, remove posIBM_XML4C-5.7.1-1.i586.rpm.
    - `rpm -e posIBM_XML4C-5.7.1-1.i586.rpm`

3.5 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/ibm/javapos/etc (IRES)

Property: provider.eventSocket.Port

Default value: 42114

Description: Port number used by Windows Event CIM Provider.

Property: provider.response.timeout

Default value: 30000

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

Property: upos.requestSocket.Port

Default value: 42115

Description: Port number used by UPOS 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, however due to limitations in IRES implementation, it might be necessary to report events directly to the Pegasus Server.

Property: provider.maxQueryThreads

Default value: 10

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

2.0 Problem Determination

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

Enabling Java Trace:

1. Go to IBM JavaPOS properties directory and open the "jutil.properties", for windows is “c:\pos\IBMJPOS\CONFIG” and for IRES is “/opt/ibm/javapos/etc”
2. Turn on the "com.ibm.jutil.tracing.TurnOnAllNamedTracers=ON" trace.
3. The location for these files is “<HOME>/.ibmjpos” where <HOME> is the absolute path to the user’s home directory.

Enabling Provider logging:

1. Create a system environment variable called = "UPOS_SYSMGMT_LOG" with value="1"
    a. As an example for Windows XP:
        - Right click on “My Computer” and select properties
        - Select “advanced” tab.
        - Click on “Environment variables”.
        - In “System variables” select “new”.
        - At “variable name:” write “UPOS_SYSMGMT_LOG”.
        - At “variable value:” write “1”;
        - Click on “OK” at “New System variables” window.
        - Click on “OK” at “environment variables” window.
        - Click on “OK” at “System Properties” window.

    b. An example for Linux:
        - Edit the “/etc/profile” file
        - Add the following line “export UPOS_SYSMGMT_LOG=1”
        - Restart the system
2. The log file “UPOS_SysMgmt.log” will be created in "c:\pos\log" directory on Windows and “/var/log” directory on IRES.

3.0 IBM Defined Management Services Design

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 IBM management service for Retail devices is based off the CIM schema for Retail devices.

UnifiedPOS 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. The property is initialized at open time.

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 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.

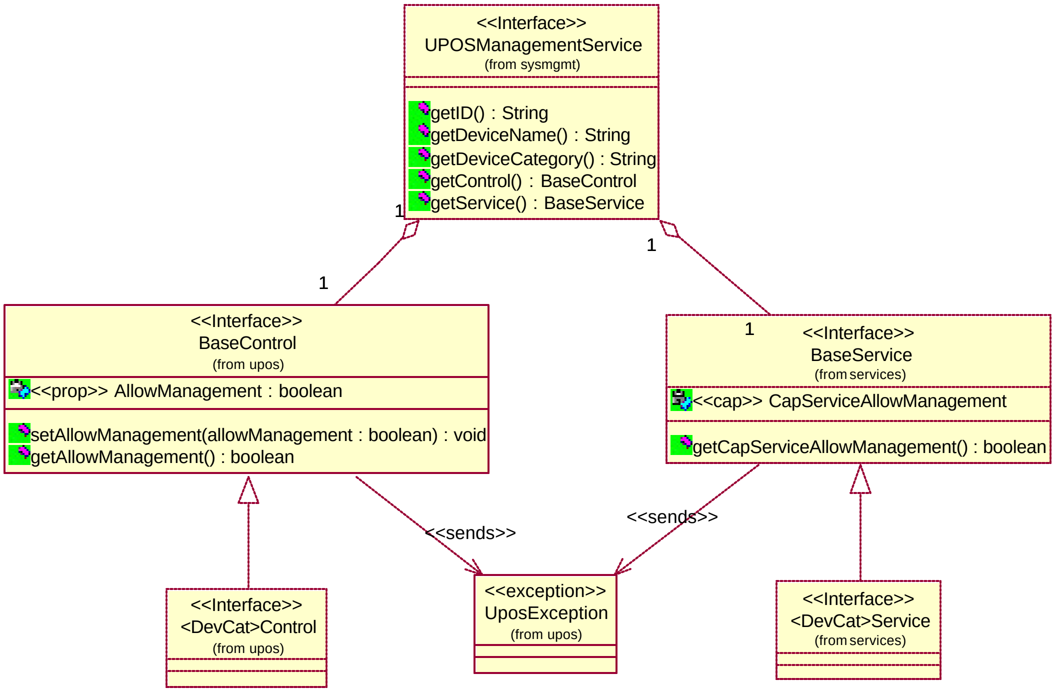
When **CapServiceAllowManagement** is false and the **AllowManagement** is true the control of the device will register the **UPOSManagementService** with UPOS Management Services. Finally when both are false the device will not participate in systems management.

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

CapServiceAllowManagement capability has as default value false

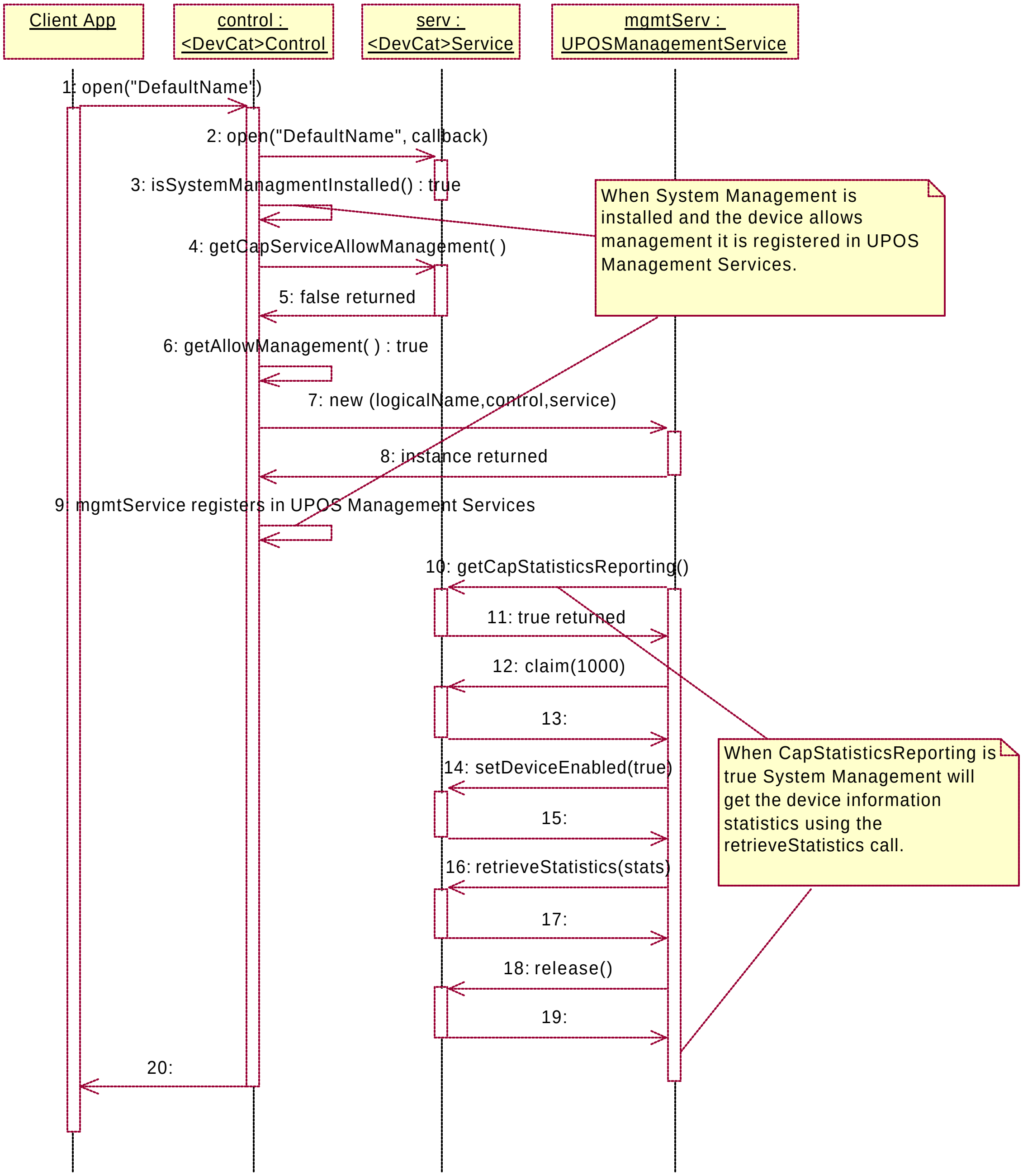
The following sections describe the class diagrams and sequence diagrams created in IBM UnifiedPOS 1.9.1 release.

3.2 Controls Class Diagram for Management Service



3.3 Device Registration Sequence Diagram

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


Client App
control : <DevCat>Control
serv : <DevCat>Service
mgmtServ : UPOSManagementService
1: open("DefaultName")
2: open("DefaultName", callback)
3: isSystemManagmentInstalled() : true
4: getCapServiceAllowManagement( )
5: false returned
6: getAllowManagement( ) : true
7: new (logicalName,control,service)
8: instance returned
9: mgmtService registers in UPOS Management Services
10: getCapStatisticsReporting()
11: true returned
12: claim(1000)
13:
14: setDeviceEnabled(true)
15:
16: retrieveStatistics(stats)
17:
18: release()
19:
20:

When System Management is installed and the device allows management it is registered in UPOS Management Services.

When CapStatisticsReporting is true System Management will get the device information statistics using the retrieveStatistics call.

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

RFIDScanner	**UPOS_RFIDScanner**	
Scale	**UPOS_Scale**	**1.9.1**
Scanner	**UPOS_Scanner**	**1.9.1**
SignatureCapture	**UPOS_SignatureCapture**	
SmartCardRW	**UPOS_SmartCardRW**	
ToneIndicator	**UPOS_ToneIndicator**	**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 definition 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

3.5 IBM UnifiedPOS Provider for System Management

The IBM 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 IBM:

3.5.1 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**

3.5.2 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**

3.5.3 UPOS System Management Events in Windows

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

3.5.4 UPOS System Management Events in Linux

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

3.5.5 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`
	`JPOS_SUE_POWER_OFF = 2002`
	`JPOS_SUE_POWER_OFFLINE = 2003`
	`JPOS_SUE_POWER_OFF_OFFLINE = 2004`
	`JPOS_SUE_UF_PROGRESS = 2100`
	`JPOS_SUE_UF_COMPLETE = 2200`
	`JPOS_SUE_UF_FAILED_DEV_OK = 2201`
	`JPOS_SUE_UF_FAILED_DEV_UNRECOVERABLE = 2202`
	`JPOS_SUE_UF_FAILED_DEV_NEEDS_FIRMWARE = 2203`
	`JPOS_SUE_UF_FAILED_DEV_UNKNOWN = 2204`
	`JPOS_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_ANY = 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
	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;

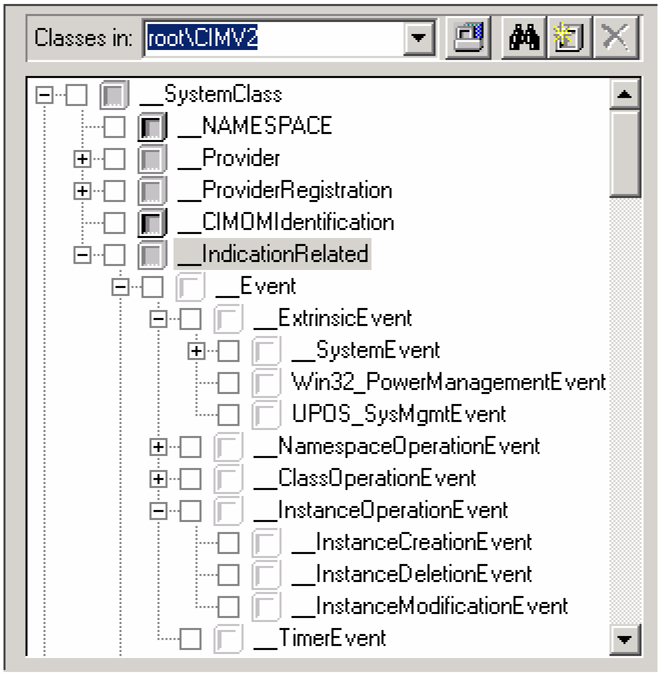
UPOS Device Category	Status Update Events (SUE) reported to System Management
Scale	`SCL_SUE_STABLE_WEIGHT     = 11`
	`SCL_SUE_WEIGHT_UNSTABLE   = 12`
	`SCL_SUE_WEIGHT_ZERO       = 13`
	`SCL_SUE_WEIGHT_OVERWEIGHT = 14`
	`SCL_SUE_NOT_READY         = 15`
	`SCL_SUE_WEIGHT_UNDER_ZERO = 16C`

3.5.6 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.

IBM 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)

3.5.7 Linux Indication Provider(supported only in SLED 11)

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

4.0 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
- CMPI Specification v1.3 http://www.wbemsource.org/doc.tpl?CALLER=documents.tpl&dcat=&gdid=3712
- 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