

# Microsoft Point of Service for .NET Service Object Verification Test

---

## Test Specification for POS Device Peripherals

A testing methodology to ensure that the software that controls POS Device Peripherals implements standard API calls.

**Microsoft**

Version 1.06  
August, 2007

---

Portions of this document specify software that is still in development. Some of the information in this documentation may be inaccurate or may not be an accurate representation of the functionality of final documentation or software. Microsoft assumes no responsibility for any damages that might occur directly or indirectly from these inaccuracies.

Microsoft does not make any representation or warranty regarding specifications in this document or any product or item developed based on these specifications. Microsoft disclaims all express and implied warranties, including but not limited to the implied warranties of merchantability, fitness for a particular purpose and freedom from infringement. Without limiting the generality of the foregoing, Microsoft does not make any warranty of any kind that any item developed based on these specifications, or any portion of a specification, will not infringe any copyright, patent, trade secret or other intellectual property right of any person or entity in any country. It is your responsibility to seek licenses for such intellectual property rights where appropriate. Microsoft shall not be liable for any damages arising out of or in connection with the use of these specifications, including liability for lost profit, business interruption, or any other damages whatsoever. Some states do not allow the exclusion or limitation of liability or consequential or incidental damages; the above limitation may not apply to you.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Microsoft, Active Directory, MSDN, Windows, the Windows Logo, Win32, and Windows NT are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

© 2005-2007 Microsoft Corporation. All rights reserved.

---

## Contents

Welcome .....	4
1. Technical Requirements .....	5
1.1 Common Functionality .....	5
1.2 Cash Drawer .....	6
1.3 Check Scanner .....	6
1.4 Line Display .....	6
1.5 MSR (Magnetic Stripe Reader) .....	6
1.6 Pin Pad .....	6
1.7 POS Keyboard .....	6
1.8 POS Printer .....	6
1.9 Scanner .....	7
1.10 Biometrics .....	7
1.11 MICR (Magnetic Ink Character Recognition) .....	7
2. Testing POS Device Peripherals with POSVerify .....	8
2.1 Connect and Configure the POS Device Peripheral .....	8
2.2 Confirm POSVerify identifies the POS Device Peripheral .....	8
2.3 Run the Verification Test .....	9
2.4 View Results and Log File .....	12
2.5 Troubleshooting POSVerify .....	13

## Welcome

The “Microsoft Point of Service for .NET Service Object Verification Test” describes the technical requirements for POS Device Peripherals and the software that controls them to pass verification testing for Microsoft Point of Service for .NET (POS for .NET). This test is intended to ensure that the software that controls the device implements various standard API calls and that a minimal level of functionality has been defined when exercising the device from an application leveraging POS for .NET.

Even though there are currently 26 different device classes defined, this test is restricted to the following device classes:

- Cash Drawer
- Check Scanner
- Line Display
- MSR
- Pin Pad
- POS Keyboard
- POS Printer
- Scanner
- Biometrics
- MICR

# 1. Technical Requirements

The Microsoft Point of Service for .NET Service Object Verification Test for POS Device Peripherals is broken into two separate areas. The first is to ensure that the POS Device Peripheral implements a common set of methods, properties and events that must be implemented by all POS Device Peripherals. The second is to exercise the device for a minimal functionality test that is specific to the Device Class of the POS Device Peripheral.

## 1.1 Common Functionality

All POS Device Peripherals must support select Properties, Methods, and Events, and support Plug and Play.

The **Properties** that must be implemented are:

- AutoDisable (for devices with event driven input only)
- CapPowerReporting
- CapStatisticsReporting
- CapUpdateStatistics
- CheckHealthText
- Claimed
- DataCount (for devices with event driven input only)
- DataEventEnabled (for devices with event driven input only)
- DeviceDescription
- DeviceEnabled
- DeviceName
- FreezeEvents
- OpenResult
- OutputID (for Asynchronous Output devices only)
- PowerNotify
- PowerState
- ServiceObjectDescription
- ServiceObjectVersion
- State

The **Methods** that must be implemented are:

- Open(...)
- Close(...)
- Claim (...)
- Release (...)
- CheckHealth(...)
- ClearInput(...)
- ClearOutput(...)
- DirectIO(...)
- ResetStatistics(...)
- RetrieveStatistics(...)
- UpdateStatistics(...)

The **Events** that must be supported are:

- DataEvent (for devices with event driven input only)
- DirectIOEvent
- DeviceErrorEvent (for devices that support DeviceErrorEvents only)
- OutputCompleteEvent (for Asynchronous Output devices only)
- StatusUpdateEvent

All POS Device Peripherals must support Plug and Play. Plug and Play is defined as the ability to attach an I/O device to a host, and enable the host to determine that the device has been added, identify it, and either automatically install the necessary device drivers or prompt the user for a diskette containing them. At a high level, the hardware must respond to the Windows Plug and Play sub-system by providing a Device ID and optionally provide an XML file that contains any appropriate configuration parameters.

## **1.2 Cash Drawer**

The minimal functionality required by a cash drawer is to respond to an "Open the drawer" command from an application.

## **1.3 Check Scanner**

A Check Scanner is used to scan a document into the system for processing. Typically, the document to be scanned is a Check or voucher of some type.

The minimal functionality of a Check Scanner is to read image data from a document.

To test a POS Peripheral Device for this device class, please provide a sample document to be scanned.

## **1.4 Line Display**

The minimal functionality of a Line Display device is to display text characters.

## **1.5 MSR (Magnetic Stripe Reader)**

An MSR is used to read the magnetic encoded data on the magnetic strip on credit cards or similar instruments (debit cards, stored value cards, identity cards, etc).

The minimum functionality of an MSR is to read encoded data from a magnetic stripe. Data is available from any combination of ISO or JIS-I tracks 1, 2, 3, and JIS-II.

To test a POS Peripheral Device for this device class, please provide a sample card with a magnetic strip to be read by the MSR.

## **1.6 Pin Pad**

The minimum functionality of a Pin pad is to accept a PIN entry at the Pin Pad keypad and provide it to the application encrypted.

## **1.7 POS Keyboard**

The minimum functionality of a POS Keyboard is to read keys from the POS Keyboard. The Keyboard may be an auxiliary keyboard, or it may be a virtual Keyboard consisting of some or all of the keys of the system keyboard.

To test a POS Peripheral Device for this device class, please provide a list identifying all POS keys, and the 32-bit number associated with each of them.

## **1.8 POS Printer**

The POS Printer does not attempt to encapsulate the behavior of a generic graphics printer. Rather, for performance and ease of use considerations, the interfaces are defined to directly control a POS printer. Usually, an application will print one line to one station per method, for ease of use and accuracy in recovering from errors.

The printer model defines three stations with the following general uses:

- Journal: Used for simple text to log transaction and activity information. Kept by the store for audit and other purposes.
- Receipt: Used to print transaction information. Usually given to the customer. Also often used for store reports. Contains either a knife to cut the paper between transactions, or a tear bar to manually cut the paper.
- Slip: Used to print information on a form. Usually given to the customer. Also used to print "validation" information on a form. The form type is typically a check or credit card slip. Sometimes, limited forms-handling capability is integrated with the receipt or journal station to permit validation printing. Often this limits the number of print lines, due to the station's forms-handling throat depth. The POS Printer nevertheless addresses this printer functionality as a slip station.

The minimum functionality of the POS Printer is to support printing in the default character set of ASCII characters (0x20 through 0x7F), which includes space, digits, uppercase, lowercase, and some special characters. It is acceptable for the POS Printer to translate them to close approximations – such as lowercase to uppercase.

## **1.9 Scanner**

The minimum functionality for a scanner is to read encoded bar code data.

To test a POS Peripheral Device for this device class, please provide a sample document with bar codes to be scanned.

## **1.10 Biometrics**

The minimum functionality for a Biometrics device is the ability to gather biometrics data from a sensor in the form of a BiometricsInformationRecord (BIR) and to be able to perform operations on the BIR for purposes of Identification and Verification. The device must also be able to store a user defined payload and retrieve that payload upon successful Verification.

## **1.11 MICR (Magnetic Ink Character Recognition)**

A MICR device is used to read magnetic ink data from a document into the system for processing. Typically, the document to be scanned is a Check or voucher of some type.

The minimal functionality of a MICR is to read magnetic ink data from a document.

To test a POS Peripheral Device for this device class, please provide a sample document to be scanned.

## 2. Testing POS Device Peripherals with POSVerify

POSVerify is available for download from the webpage for the Microsoft Point of Service for .NET Service Object Verification Test on [www.veritest.com](http://www.veritest.com). POSVerify is a straightforward tool that tests POS Device Peripherals for selected Device Classes. Please note that most POS Peripheral Devices will be tested for a single device category. Only multi-functional POS Peripheral Devices that fall into multiple device categories may be tested for multiple device categories.

### 2.1 Connect and Configure the POS Device Peripheral

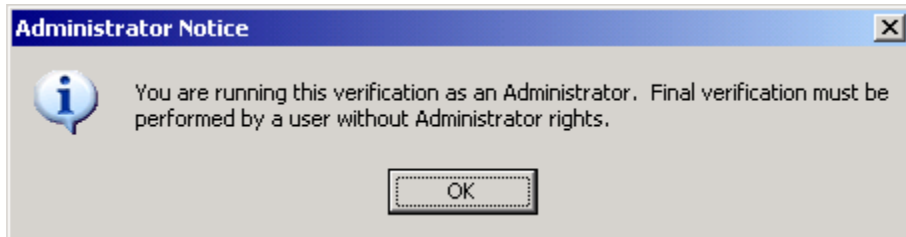
Connect the POS Device Peripheral to be tested to a system running the latest version of Microsoft Point of Service for .NET (you can download Microsoft Point of Service for .NET from the [Microsoft Download Center](#)). Confirm that the Windows Plug-and-Play sub-system successfully recognizes that the device has been added, correctly identifies it, and either automatically install the necessary device drivers or prompts for removable media containing them.

Complete the installation procedure for the device, then ensure that the service object for the device is also properly installed.

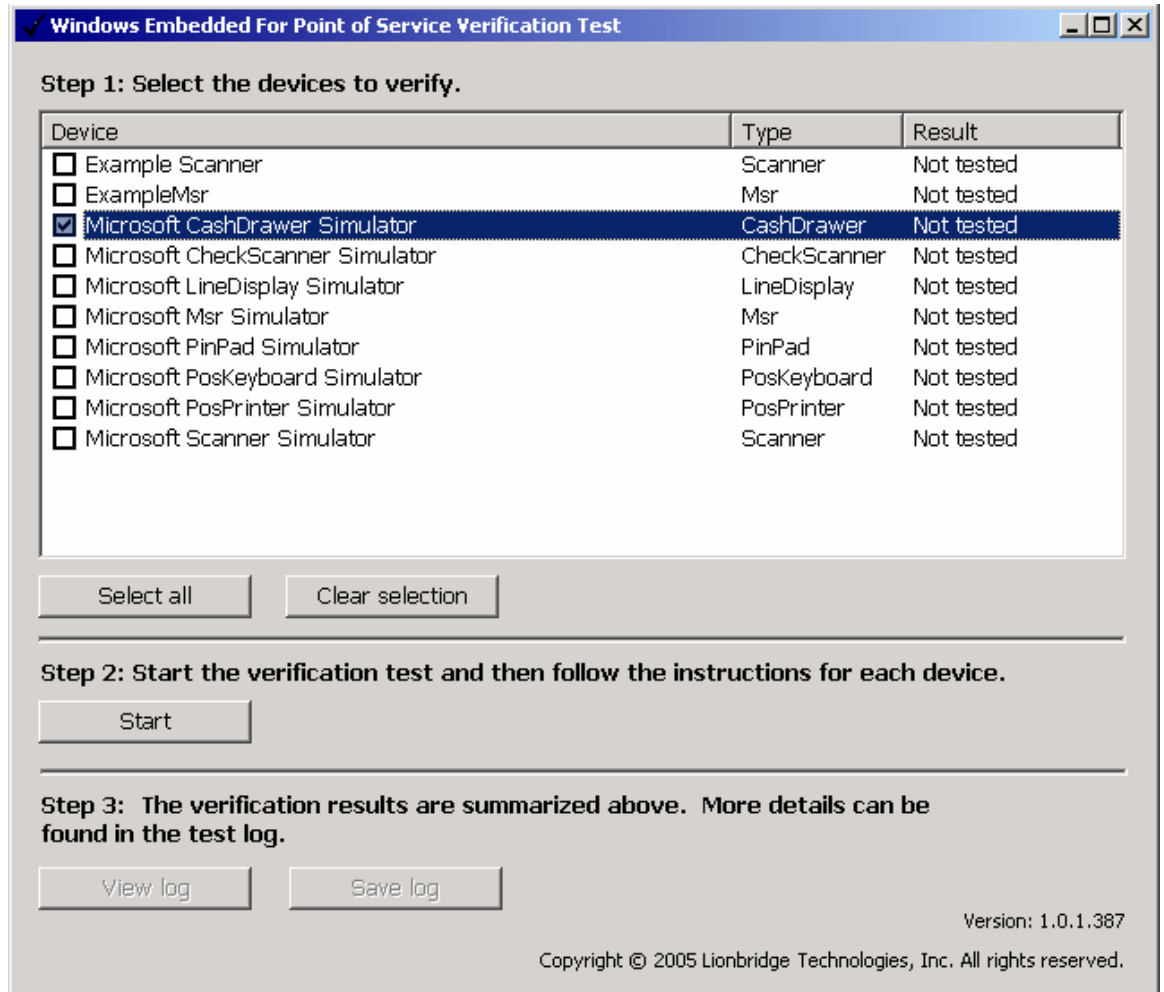
### 2.2 Confirm POSVerify identifies the POS Device Peripheral

Log in to the system as a user that does not have Administrator rights. Invoke the POSVerify test tool, and confirm that the POS Device Peripheral is correctly identified.

Please note that POSVerify can be invoked by an Administrative user, but it will issue the following warning:



Within POSVerify, select the POS Device Peripheral to be tested.



## 2.3 Run the Verification Test

Click on the "Start" button to run the verification test. POSVerify will issue instructions depending on the device category (or categories) of the POS Device Peripheral being tested. Follow these instructions.

For Example, when POSVerify runs the verification test for a Cash Drawer device, it will open a new window like the following:

**Cash Drawer Device Verification** [X]

A Cash Drawer device must respond to the "Open the drawer" command. The drawer status reported by the device is displayed below.

Cash Drawer status:

**Step 1: Verify the drawer is physically closed.**

---

**Step 2: Send the "Open the Drawer" command.**

---

**Step 3: Enter the result**

Did the cash drawer open?

Yes

No

---

**Step 4: Enter comments for the Log (this is optional).**

In this case, confirm that POSVerify correctly identifies the status of the Cash Drawer. Ensure the Cash Drawer is closed. Click on the "Open the drawer" button, and confirm the Cash Drawer opens.

Record the result within POSVerify, and click on the "OK" button to continue:

**Cash Drawer Device Verification**

A Cash Drawer device must respond to the "Open the drawer" command. The drawer status reported by the device is displayed below.

Cash Drawer status:

**Step 1: Verify the drawer is physically closed.**

---

**Step 2: Send the "Open the Drawer" command.**

---

**Step 3: Enter the result**

Did the cash drawer open?

Yes

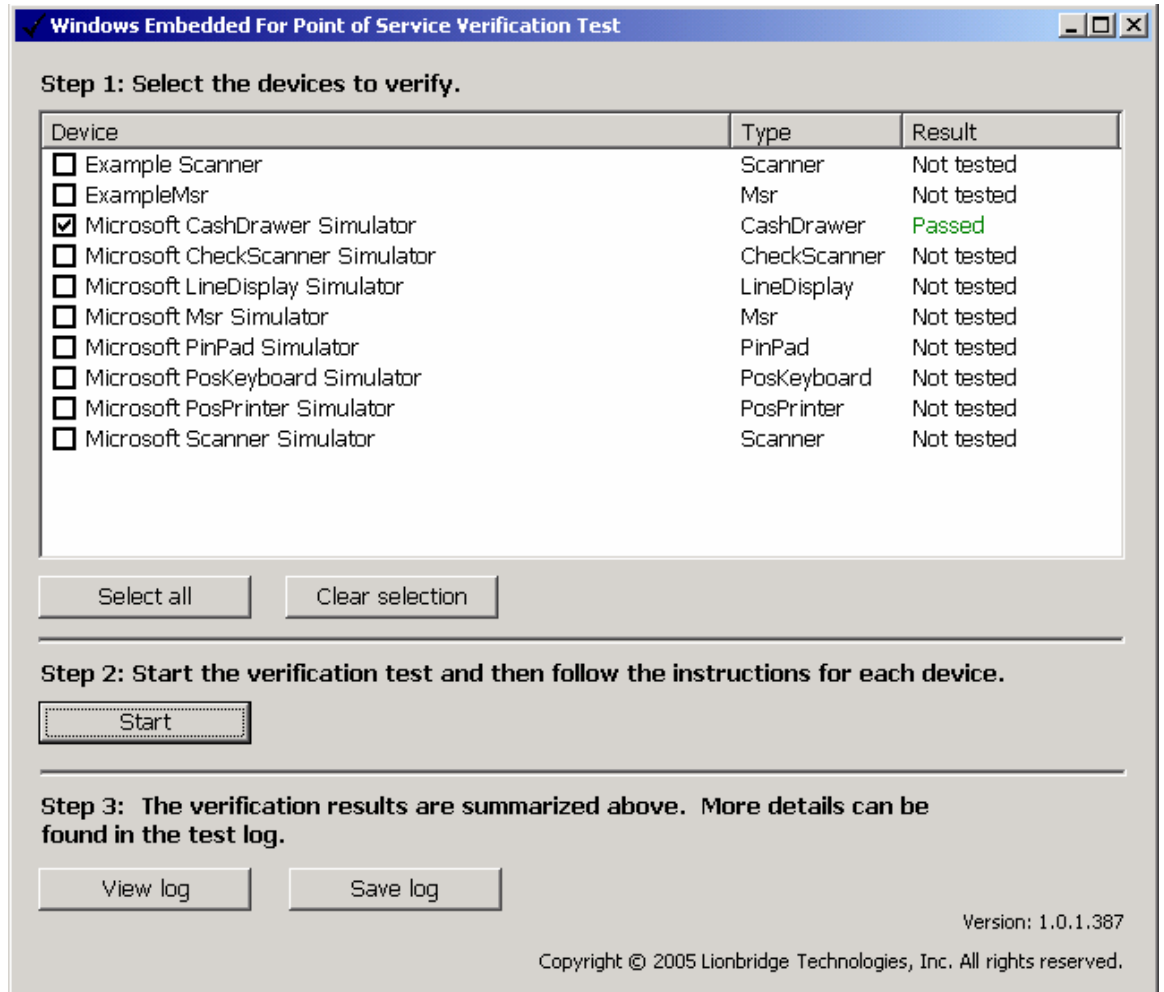
No

---

**Step 4: Enter comments for the Log (this is optional).**

## 2.4 View Results and Log File

Upon completion of all test runs, the overall test results for each device and device type are listed on the main Window in POSVerify.



POSVerify records the test results of the individual test cases in a log file in XML format, along with general environmental information concerning the system, and service object. Please view the log file for detailed information, particularly for fail issues.

```
<?xml version="1.0" encoding="us-ascii" standalone="yes" ?>
- <VerificationLog Time="7/20/2005 1:31:21 PM">
  <Detail>Using a delay of 100 ms between verification of state changes</Detail>
- <Device Time="7/20/2005 1:31:21 PM">
  <Compatibility>CompatibilityLevel1</Compatibility>
  <Description>Microsoft Cash Drawer Simulator</Description>
  <ServiceObjectName>Microsoft CashDrawer Simulator</ServiceObjectName>
  <ServiceObjectVersion>1.0.2.0</ServiceObjectVersion>
  <HardwareDescription />
  <HardwareId />
  <HardwarePath />
  <IsDefault>False</IsDefault>
  <ManufacturerName>Microsoft Corporation</ManufacturerName>
  <LogicalNames />
  <Type>CashDrawer</Type>
  <UposVersion>1.8</UposVersion>
- <Tests Time="7/20/2005 1:31:21 PM">
```

```

<Method Name="Open" Result="Verified" />
<Property Name="DeviceName" Type="read-only" Result="Verified" Value="Microsoft CashDrawer Simulator" />
<Property Name="DeviceDescription" Type="read-only" Result="Verified" Value="Microsoft CashDrawer Simulator" />
<Property Name="ServiceObjectDescription" Type="read-only" Result="Verified" Value="Microsoft Cash Drawer Simulator" />
<Property Name="ServiceObjectVersion" Type="read-only" Result="Verified" Value="1.8" />
<Property Name="CapPowerReporting" Type="read-only" Result="Verified" Value="None" />
<Property Name="CapStatisticsReporting" Type="read-only" Result="Verified" Value="True" />
<Property Name="CapUpdateStatistics" Type="read-only" Result="Verified" Value="True" />
<Method Name="Claim" Result="Verified" />
<Property Name="Claimed" Type="read-only" Result="Verified" Value="True" />
<Property Name="State" Type="read-only" Result="Verified" Value="Idle" />
<Property Name="FreezeEvents" Type="read-write" Result="Verified" Value="True" />
<Property Name="DeviceEnabled" Type="read-write" Result="Verified" Value="True" />
<Method Name="CheckHealth" Result="Verified" />
<Property Name="PowerState" Type="read-only" Result="Verified" Value="Unknown" />
<Property Name="PowerNotify" Type="read-only" Result="Verified" Value="Disabled" />
<Method Name="RetrieveStatistics" Result="Verified" />
<Method Name="UpdateStatistics" Result="Verified" />
<Method Name="ResetStatistics" Result="Verified" />
<Method Name="Release" Result="Verified" />
<Method Name="Close" Result="Verified" />
- <MinimalFunctionality Time="7/20/2005 1:31:22 PM" Type="CashDrawer">
  <Event Type="Status" Time="7/20/2005 1:31:58 PM" Status="1" />
  <Detail>The cash drawer is Open</Detail>
  <Result>Verified</Result>
  </MinimalFunctionality>
</Tests>
<Result>PASSED</Result>
</Device>
</VerificationLog>

```

## 2.5 Troubleshooting POSVerify

By default, POSVerify uses a delay of 100 ms between verification of state changes of the POS Device Peripheral being tested. This delay may be too short for some types of devices. If you find that POSVerify does not recognize the state change of a device, you can increase the delay with a start-up switch.

For example, to increase the delay to 500 ms, invoke POSVerify as follows:

```
POSVerify.exe /delay=500
```

The delay can range from 0 to 5000 ms. Please note that the test is not threaded, so you will notice delays and possibly drawing issues with POSVerify's user interface if you increase the delay to 1000 ms or higher.