

# **Service Object Manager**

For use with Microsoft's POS for .NET

## **User Guide**

**Developed by: SJJ Embedded Micro Solutions, LLC., San Diego Ca.**

Copyright © 2007 SJJ Embedded Micro Solutions, LLC., All Rights Reserved.

No part of this guide may be copied, duplicated, reprinted, and stored in a retrieval system by any means, mechanical or electronic, without the written permission of the copyright owner.

First Printing: January 2007

Published in the United States by

**SJJ Embedded Micro Solutions, LLC.**

11921 Tivoli Park Row #5  
San Diego, CA 92128 USA

[www.sjjmicro.com](http://www.sjjmicro.com)

Attempts have been made to properly reference all copyrighted, registered, and trademarked material. All copyrighted, registered, and trademarked material remains the property of the respective owners.

The publisher, author, and reviewers make no warranty for the correctness or for the use of this information, and assume no liability for direct or indirect damages of any kind arising from the information contained herewith, technical interpretation or technical explanations, for typographical or printing errors, or for any subsequent changes in this article.

The publisher and author reserve the right to make changes in this publication without notice and without incurring any liability.

All trademarks and copyrights are property of their respective owners.

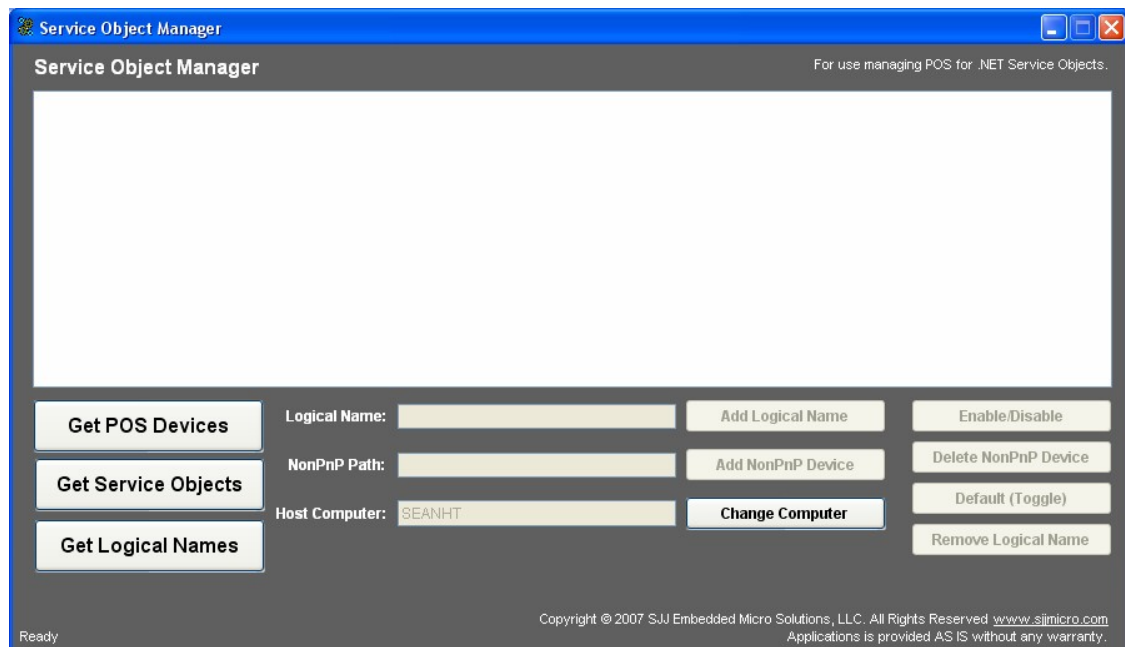
## Table of Contents

1	OVERVIEW .....	4
2	BASIC USAGE .....	5
2.1	<i>Get POS Devices</i> .....	5
2.1.1	Add Logical Name Button .....	6
2.1.2	Set NonPnP Path Button.....	6
2.1.3	Enable/Disable Button.....	6
2.1.4	Delete NonPnP Device Button.....	6
2.1.5	Default (Toggle) Button .....	6
2.2	<i>Get Service Objects</i> .....	6
2.2.1	Add NonPnP Device Button.....	7
2.3	<i>Get Logical Names</i> .....	7
2.3.1	Remove Logical Name Button .....	8
2.4	<i>Change Computer</i> .....	8

# 1 Overview

Service Objects are at the center of POS for .NET. The ability to setup and manage service objects are an important part of developing, installing, and maintaining POS for .NET applications. The POS for .NET SDK provides a command line tool called POSDM.EXE to manage service objects. Service Object Manager (SOM) is a graphical implementation of POSDM.EXE. SOM performs the same command set functionality of POSDM.EXE:

- Lists POS devices available to applications
- Lists service objects installed in the system
- Lists logical names for POS for .NET applications to use
- Allows you to set logical names for a service objects
- Enable and Disable a service objects
- You can set a service objects as default if two devices of the same type are present.
- Add and remove Non Plug and Play devices
- Manage systems over a network.



Unlike POSDM, you can visually see the details of each service object and highlight the individual SO you want to control, thus saves time in having to type the full command line.

**Requirements:** .NET Framework 2.0 and POS for .NET 1.1 or higher must be installed on the system to use SOM.

Before using SOM, you should already be familiar with POS for .NET, UPOS, and the role of service objects / OPOS drivers. More information on POS for .NET can be found in the book: *Windows® Embedded for Point of Service / POS for .NET Step-by-Step*. Available at [www.sjjmicro.com](http://www.sjjmicro.com)

**Warning:** SOMgr.EXE v1.0.0.0 is a free utility, and it is supplied as is without any warranty. Use the Service Object manager at your own risk.

## 2 Basic Usage

Service Object Manager is a simple point and click tool. The buttons provide all the command functionality. The first set of buttons on the far left, are the list buttons. The buttons on the right hand side are the command and computer change buttons. You need to first get a list before performing any actions on service object or logical name. Different command buttons become available.

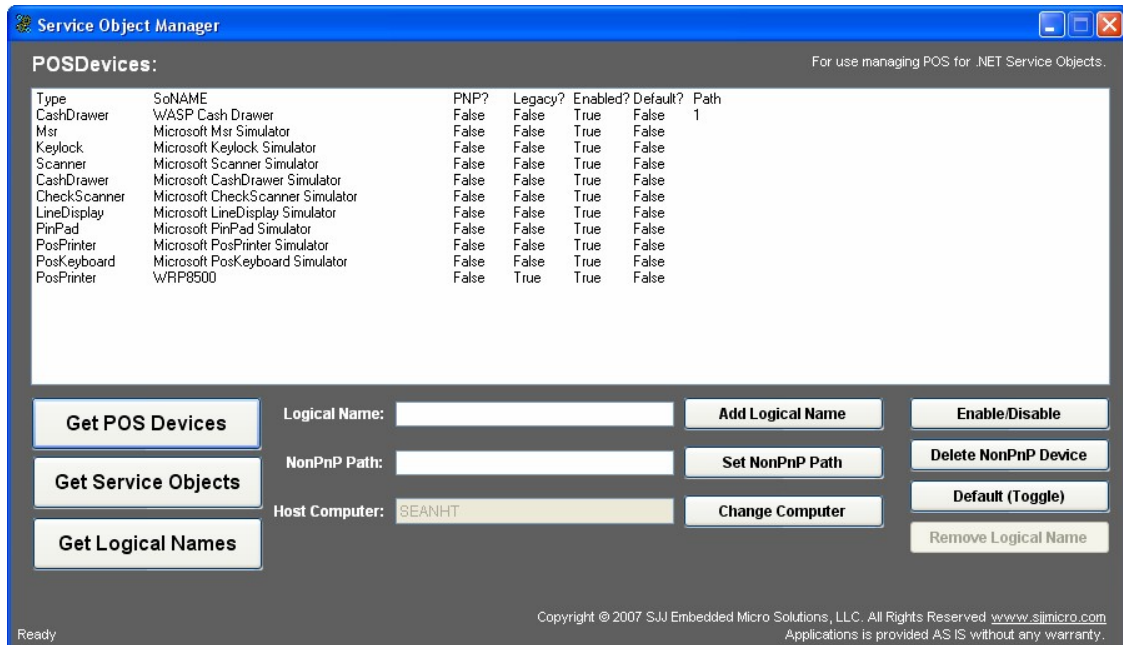
List Button	Function
Get POS Devices	Provides a list of POS Devices available
Get Service Objects	Lists the physical Service Object location
Get Logical Names	Lists the logical names assigned to a service object

The lists are custom versions of the POSDM commands. The *POS for .NET Management Guide* that comes with the POS for .NET SDK explains the basic foundation for the functions used by SOM. Only the major POSDM commands have been implemented. Please review the Pos for .NET help files for information on other commands available.

**Note:** The lists in the list box have been aligned using padding functions. This means that some data might get out of alignment because of spacing. This is a know issue and doesn't interfere with the basic commands.

### 2.1 Get POS Devices

The Get POS Devices button lists physical POS devices available in the system. Command buttons for Enable/Disable/ Delete NonPnP Device, Default (Toggle), Add logical Name, and Set NonPnPPath all become active.



The column headings are defined as follows:

Column Header	Description
Type	POS Device Type
SoName	The actual Service Object Name
PnP?	Will be True if the SO supports Plug and Play
Legacy?	True = OPOS driver / False = POs for .NET Service Object
Enabled?	True if the Service object is enabled. False if the Service object is disabled.
Default?	If TRUE, the device is the default in a POS Device category
Path	Hardware path for the service object

To perform an action on a service object, click on a service object in the list box and click the desired command button. For example, if you have the POS for .NET SDK installed, you will see several simulator services objects in the list. Click on the MSR simulator service object and click the Enable/Disable button. The MSR will be disabled. Repeat the action to enable the MSR simulator.

**Note:** OPOS drivers will also be lists, but you must use OPOS configuration tools to manage them.

### 2.1.1 Add Logical Name Button

Logical names provide a common ID for applications to open devices for access. You can assign a Logical Name to a service object. Highlight the service object in the list, enter a logical name in the text box, and press the Add Logical Name button. The list box will change and list all the logical names assigned to service objects including the logical name you just added.

### 2.1.2 Set NonPnP Path Button

Many POS devices are based on old non-Plug and Play technology like RS-232 serial ports. Service objects can be written to have either a hard coded path to the hardware or have some flexibility to be attached to any COM port. The Set NonPnP Path button allows you to set the COM port for those service objects designed with the flexibility. You will have to follow the service object documentation in order to enter the correct COM path.

### 2.1.3 Enable/Disable Button

The Enable/Disable button simply enables or disables a highlighted service object.

### 2.1.4 Delete NonPnP Device Button

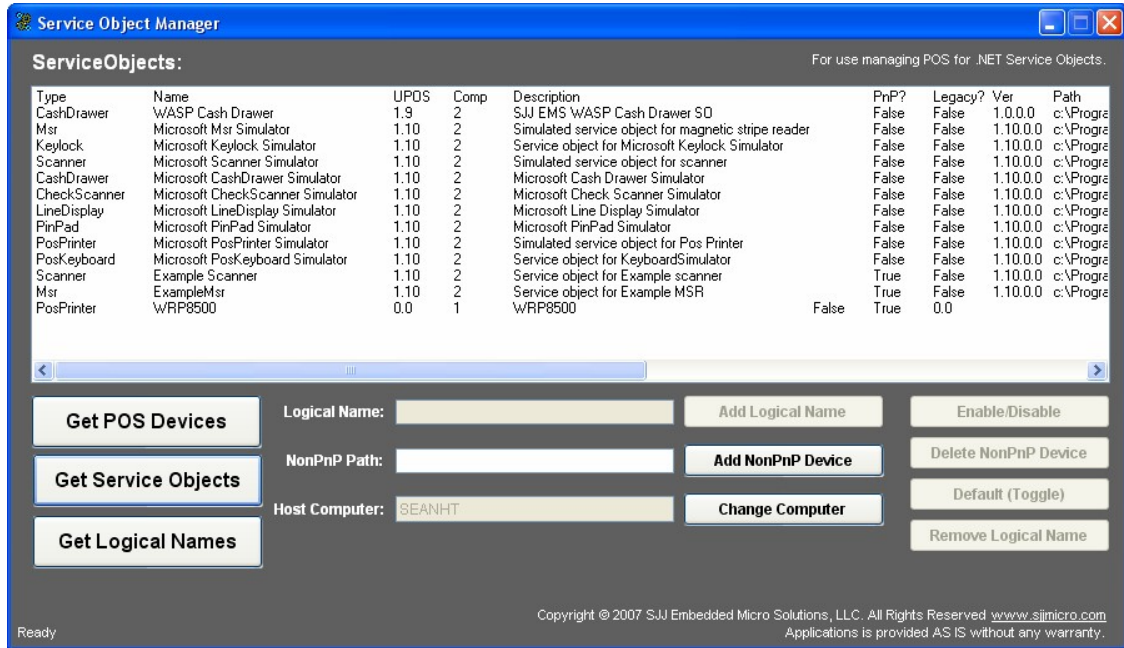
This button allows you to delete the path for a Non-Plug and Play service object.

### 2.1.5 Default (Toggle) Button

Sets a highlighted service object as the default device for it's device type. PosExplorer.GetDevice(type) method will return the default device even if there is more than one device of the type available. If you have more than one scanner in the system, you can set one scanner to be the default that applications will always open.

## 2.2 Get Service Objects

The Get Service Objects lists all the available service object assemblies and OPOS drivers installed in the system. Because there is flexibility in where service objects binaries can be installed, the list also provides the actual file/folder path to the dll binary, which can be very helpful in debugging a system. Only the Add NonPnP Device command button is enabled.



The column headings are defined as follows:

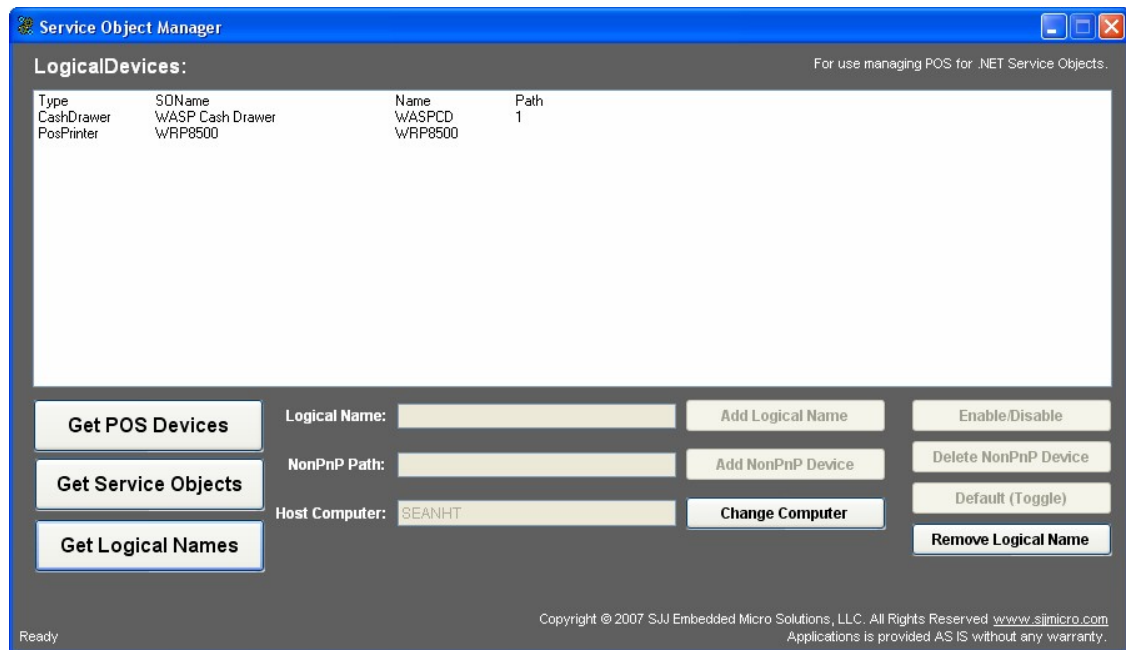
Column Header	Description
Type	POS Device Type
Name	The actual Service Object Name
UPOS	What is UPOS version. A version of 0.0 implies a OPOS driver.
COMP	Lists the major version of POS for .NET the service object is compatible with.
Description	Short description of the service object
PnP?	Will be True if the SO supports Plug and Play
Legacy?	True = OPOS driver / False = POS for .NET Service Object
Ver	The version number of the service object
Path	File location path for the service object binaries

**2.2.1 Add NonPnP Device Button**

Performs the same action as Set NonPnP Device button to set the hardware path for a Non-Plug and Play service object device, but also adds the device as an available device for POS for .NET applications.

**2.3 Get Logical Names**

When you click on the Get Logical Names button, the list box will display all the logical names associated with a service object. The only button enabled is the Remove Logical Name button. If OPOS drivers are installed, you will see logical names defined for OPOS drivers. SOM has no control over OPOS drivers. You must use OPOS utilities to manage these drivers.



The columns are defined as follows:

Column Header	Description
Type	POS Device Type
SoName	The actual Service Object Name
Name	The Logical Name assigned to the service Object
Path	Hardware path for the service object

### 2.3.1 Remove Logical Name Button

If you want to remove a logical name from the list, highlight the service object and click the Remove Logical name button. The list will refresh and display the updated information.

## 2.4 Change Computer

SOM also can be used to manage other computers on the network. The Change Computer button allows you to switch to different POS computers on the network. All you have to enter is the name of the computer. No UNC, “//”, or other long path names are required. The remote computer must be enabled to support remote management calls.