



SII OPOS Application Programmer's Guide

Rev.05

[Products]

DSP-A01 Series

Seiko Instruments Inc.

Rev.01 July 2019
Rev.02 October 2019
Rev.03 August 2020
Rev.04 December 2021
Rev.05 July 2023


Copyright© 2019-2023 by Seiko Instruments Inc.
All rights reserved.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the U.S., Japan, and other countries.

Bluetooth® is a registered trademark of Bluetooth SIG, Inc.

Seiko Instruments Inc. (hereinafter referred to as "SII") has prepared this manual for use by SII personnel, licensees, and customers. The information contained herein is the property of SII and shall not be reproduced in whole or in part without the prior written approval of SII.

SII reserves the right to make changes without notice to the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical, arithmetic, or listing errors.

SII  is a trademark of Seiko Instruments Inc.

Introduction

This manual describes the display function of "SII OPOS" (hereinafter referred to as the "software") provided by Seiko Instruments Inc. (hereinafter referred to as "SII").

This software is OpenPOS control based on UnifiedPOS Retail Peripheral Architecture Version 1.14 for controlling Display or Display connected to the SII printers.

Target Products

This section lists the product supported by the software.

	Device Name	Description in This Manual
LineDisplay	DSP-A01	Display

See "UnifiedPOS Retail Peripheral Architecture Version 1.14" (hereinafter referred to as "UPOS V 1.14") before using this software.

Supported Driver

SII driver is required for using Display.
See "1.1 Configuration" for supported SII driver.

Notation in This Manual

The notation in this manual is described.

Operation and Display

In principle, this manual is written on the basis of the following conditions:

- Screenshots and display layouts of Windows 10
- Operating instructions with a mouse and a keyboard

Operating System Abbreviations

The operating system abbreviations used in this manual are listed below.

Operating System	Abbreviation
General Microsoft® Windows®	Windows
Microsoft® Windows® 11	Windows 11
Microsoft® Windows® 11 IoT Enterprise	
Microsoft® Windows® 10	Windows 10
Microsoft® Windows® 10 IoT Enterprise	

Terms

The terms used in this manual are defined as below.

Definition	Description
UPOS V1.14	"UnifiedPOS Retail Peripheral Architecture Version 1.14"
OPOS specification	Specification of the POS device interface defined in UPOS V1.14.
OPOS Control	The device driver which conforms to the ActiveX Control and provides the function to easily integrate the POS device (POS peripheral equipment) into the POS system built on Windows. The OPOS Control described in this manual supports the API defined in the OPOS specification.
Device class	A category of POS devices that shares a set of properties, methods, and events defined in the OPOS specification. In this manual, the device class of Line Display is used.
Device name	Name for identifying the POS device targeted for control which is specified when the OPOS Control is used. It is impossible to create a duplicated device name in the same device class.
Configuration program	The program that executes addition and setting change of devices provided by this software. When this software is installed, it will be installed as [SII OPOS Utility V2] on the computer.

Definition	Description
Logical device name	Alias of device name. Arbitrary name which is set in the configuration program.
Control Object	Provides a set of properties, methods, and events to an application for each device class.
Service Object	This is called by the Control Object to execute the OPOS prescribed function for each device.
Line Display Control	OPOS Control of the Line Display device class. Controlled device is Display.
Default	The value immediately after satisfying the availability condition.
Display command	Command for controlling Display described in "DSP-A01 SERIES CUSTOMER DISPLAY TECHNICAL REFERENCE".

Symbols

The symbols used in this manual are described below.

Caution

- ◆ Notes and limitations are described.

Reference

- Supplemental information and related matters are described.

Table of Contents

Chapter 1 Overview 1-1

1.1	Configuratioin.....	1-2
1.1.1	Structural Diagram.....	1-2
1.2	Operating Environment	1-4
1.2.1	System Environment.....	1-4
1.3	Display Settings	1-4
1.4	Limitations	1-5
1.4.1	Line Display Control	1-5

Chapter 2 Installation 2-1

2.1	Installation	2-1
2.2	Uninstallation	2-6
2.3	Silent Installation.....	2-7
2.4	Silent Uninstallation.....	2-8
2.4.1	MsiExec.exe File	2-8
2.4.2	SetupOpos.exe File	2-8

Chapter 3 How to Operate Configuration Program 3-1

3.1	Startup.....	3-1
3.2	Screen Layout.....	3-2
3.2.1	Menu Bar	3-2
3.2.2	Device View	3-3
3.2.3	Setting View	3-3
3.3	Functions	3-6
3.3.1	Addition of Device.....	3-6
3.3.2	Changing Device Settings.....	3-10
3.3.3	Deletion of Device	3-11
3.3.4	Adding and Deletng Logical Device Name.....	3-12
3.3.5	Device Interactive Test	3-15

5.1	Summary	5-1
5.1.1	Common Properties.....	5-1
5.1.2	Specific Properties.....	5-2
5.1.3	Common Methods	5-3
5.1.4	Specific Methods	5-4
5.1.5	Events	5-4
5.2	Display Data and Escape Sequences.....	5-5
5.2.1	Escape Sequence Operated When Specified.....	5-5
5.2.2	Escape Sequence Operated When Displayed.....	5-5
5.3	Common Properties	5-6
	BinaryConversion Property R/W	5-6
	CapCompareFirmwareVersion Property.....	5-7
	CapPowerReporting Property	5-7
	CapStatisticsReporting Property	5-7
	CapUpdateFirmware Property	5-7
	CapUpdateStatistics Property	5-8
	CheckHealthText Property.....	5-8
	Claimed Property.....	5-8
	ControlObjectDescription Property.....	5-9
	ControlObjectVersion Property	5-9
	DeviceDescription Property	5-9
	DeviceEnabled Property R/W	5-9
	DeviceName Property.....	5-10
	FreezeEvents Property R/W	5-10
	OpenResult Property	5-11
	PowerNotify Property R/W	5-11
	PowerState Property.....	5-11
	ResultCode Property	5-12
	ResultCodeExtended Property.....	5-12
	ServiceObjectDescription Property	5-13
	ServiceObjectVersion Property	5-13
	State Property	5-13
5.4	Specific Properties	5-14
	CapBrightness Property.....	5-14
	CapCharacterSet Property.....	5-14
	CapScreenMode Property	5-14
	CharacterSet Property R/W	5-15
	CharacterSetList Property.....	5-15
	Columns Property.....	5-16
	CursorColumn Property R/W	5-16
	CursorRow Property R/W	5-16

	CursorUpdate Property R/W	5-17
	DeviceBrightness Property R/W	5-17
	DeviceColumns Property	5-18
	DeviceRows Property	5-18
	DeviceWindows Property	5-18
	Rows Property	5-18
	ScreenMode Property R/W	5-19
	ScreenModeList Property	5-19
5.5	Common Methods.....	5-20
	CheckHealth Method	5-20
	ClaimDevice Method.....	5-20
	Close Method	5-21
	CompareFirmwareVersion Method	5-21
	DirectIO Method	5-21
	Open Method.....	5-23
	ReleaseDevice Method.....	5-24
	ResetStatistics Method	5-24
	RetrieveStatistics Method	5-24
	UpdateFirmware Method	5-24
	UpdateStatistics Method	5-25
5.6	Specific Methods.....	5-26
	ClearText Method	5-26
	DisplayText Method.....	5-26
	DisplayTextAt Method.....	5-27
Chapter 6	Registry Used by this Software	6-1
6.1	Line Display Control	6-1
Chapter 7	Header File	7-1
7.1	Line Display Header File	7-1

Chapter 1 Overview

This chapter describes the overview of the software.

This software is based on ActiveX control specifications. This software provides properties, methods and events to applications. This software is not shown on UI at an application execution. Only an application executes request processing through the methods and the properties. The application receives processing results through the return values of methods, parameters, properties and events. This software is implemented as an in-process server.

For the contents of the header file used by the control, see the "OPOS Internal Header Files" of "Appendix A OLE FOR RETAIL POS - OPOS IMPLEMENTATION REFERENCE" in UPOS V1.14.
In addition, the specific values of the control are defined. For these values, refer to "Chapter 7 Header File" in this manual.

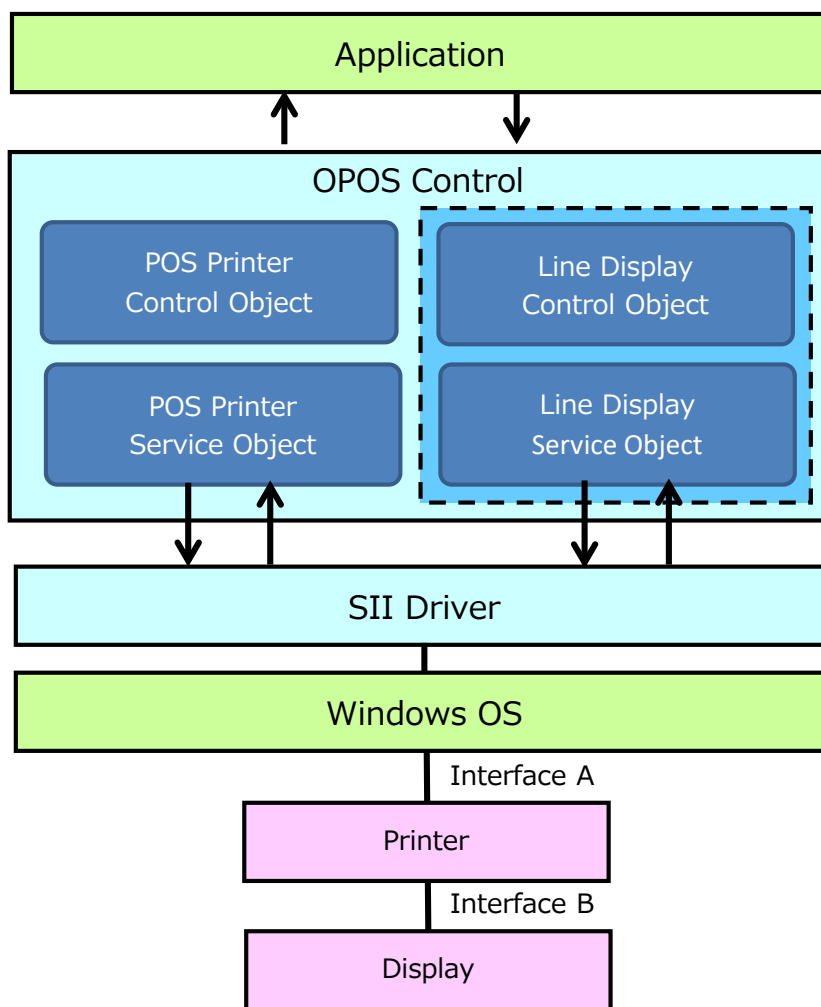
1.1 Configuration

1.1.1 Structural Diagram

The structure of the software is as follows, and the scope of this manual is indicated by broken lines. In addition, the supported SII driver is described.

There are 2 types for connecting Display.

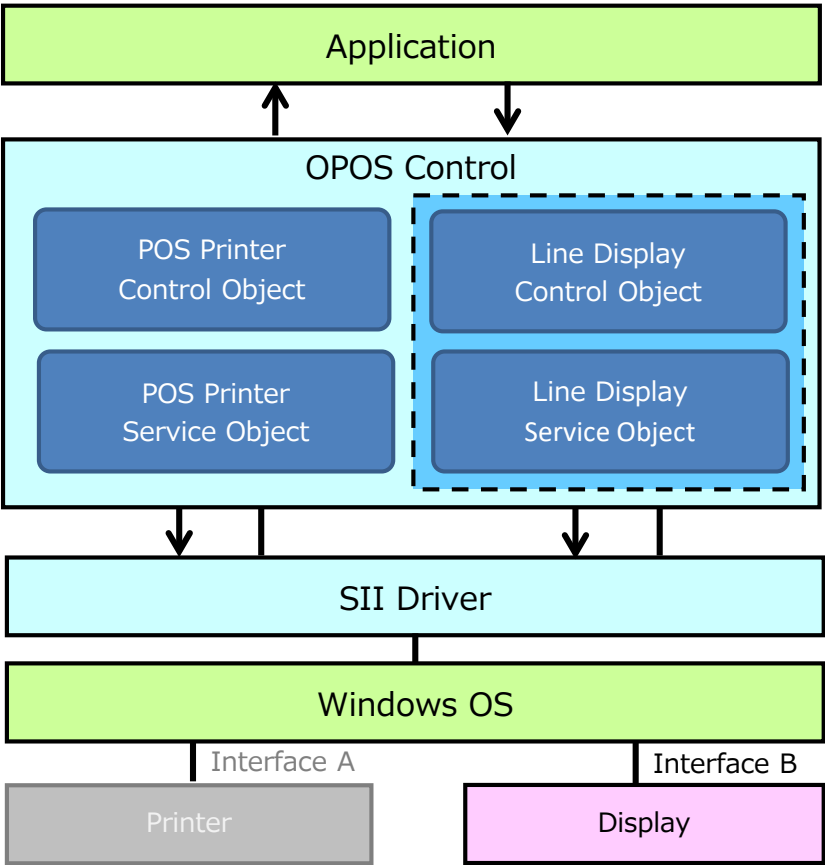
- (1) Configuration that use Display by connecting to SII printer (hereinafter described as "use via a printer")



- Supported driver

SII Driver	Interface A	Printer	Interface B	Display
"SII Printer Driver for Windows" for RP-F10/G10 series	USB Bluetooth TCP/IP	RP-F10 series	USB	DSP-A01 series

(2) Configuration that use Display alone (hereinafter described as "use alone")



- Supported driver

SII Driver	Interface A	Printer ^{*1}	Interface B	Display
"SII Printer Driver for Windows" for DSP-A01 series	-	-	USB	DSP-A01 series

^{*1}: SII printer that supports Display is as follows.
· RP-F10 series

1.2 Operating Environment

1.2.1 System Environment

The system environment of this software is shown below.

Item	Specifications
Operating System	Microsoft® Windows® 11 (64 bits) Microsoft® Windows® 11 IoT Enterprise (64 bits) Microsoft® Windows® 10 (32 bits and 64 bits) Microsoft® Windows® 10 IoT Enterprise (32 bits and 64 bits)
Driver	SII driver: "SII Printer Driver for Windows" for RP-F10/G10 Series or "SII Printer Driver for Windows" for DSP-A01 Series

1.3 Display Settings

The memory switches of the printer are set to [Value] in the following table when using the software.
See "DSP-A01 SERIES Customer Display USER'S GUIDE" for details about the memory switches.

MS	Function	Value	Note
1-1 to 1-8	Brightness Selection (Brightness)	00000000B : 10% 00000001B : 20% 00000010B : 30% 00000011B : 40% 00000100B : 50% 00000101B : 60% 00000110B : 70% 00000111B : 80% 00001000B : 90% 00001001B : 100%	Any one of [Value] on the left can be set by [DeviceBrightness] in the configuration program.

1.4 Limitations

The limitations of this software are described.

1.4.1 Line Display Control

All interfaces of Line Display Device Class defined by OPOS specifications are provided but following limitations are applied.

- (a) The method and property settings related to Teletype Mode and Marquee Mode are not supported.
- (b) The following functions are not supported.
 - Display bitmap
 - Reverse video
 - Blink
- (c) All the following methods always return OPOS_E_ILLEGAL(106) after they are enabled.
 - **DisplayBitmap**
 - **SetBitmap**
- (d) The following events are not supported.
 - **StatusUpdateEvent**
 - **DirectIOEvent** (device-specific event)

Chapter 2 Installation

This chapter describes the procedure of installation and uninstallation of the software.

It is necessary to install the driver before installing this software.

For the installation procedure of the driver, see the installation part of "SII Printer Driver for Windows User's Guide" described in "1.1 Configuration".

Caution

- ◆ This installation requires logon to the computer with administrator privileges.
- ◆ When this software (Ver.1.31 or the previous version) has already been installed and the custom installation is to be selected, the software of Ver.1.31 or the previous version is necessary to be uninstalled. When Ver.1.32 or later software is installed without uninstalling Ver.1.31 or previous, the software is to be updated installation and the type of features to install cannot be selected.
See "2.2 Uninstallation" for uninstalling.

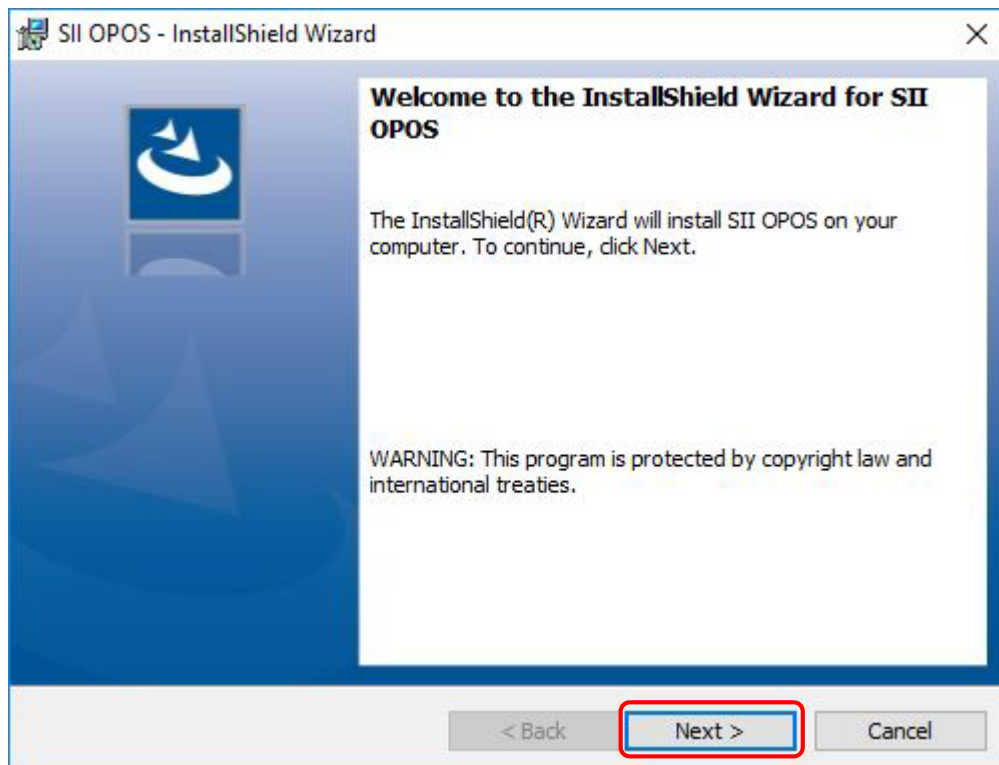
2.1 Installation

The installation of this software is described below.

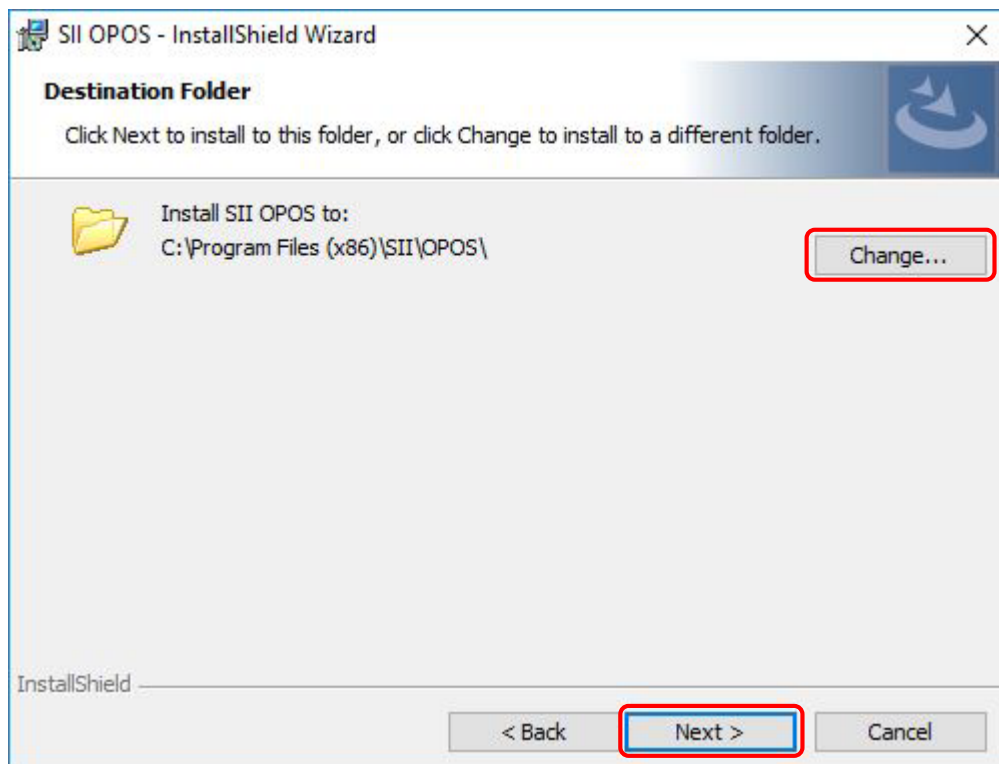
- (1) Start the setup program (SetupOpos.exe).
- (2) Select the language.
Then, click the [OK] button.



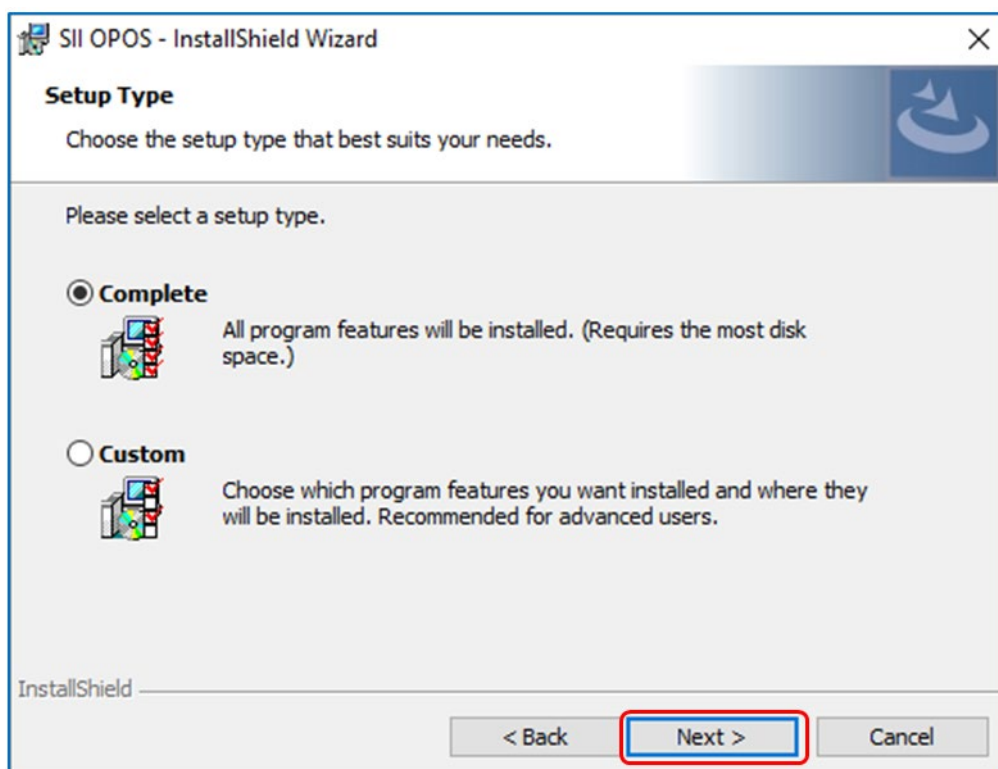
(3) When the installer starts up, click the [Next >] button.



(4) Specify the destination folder and click the [Next >] button.

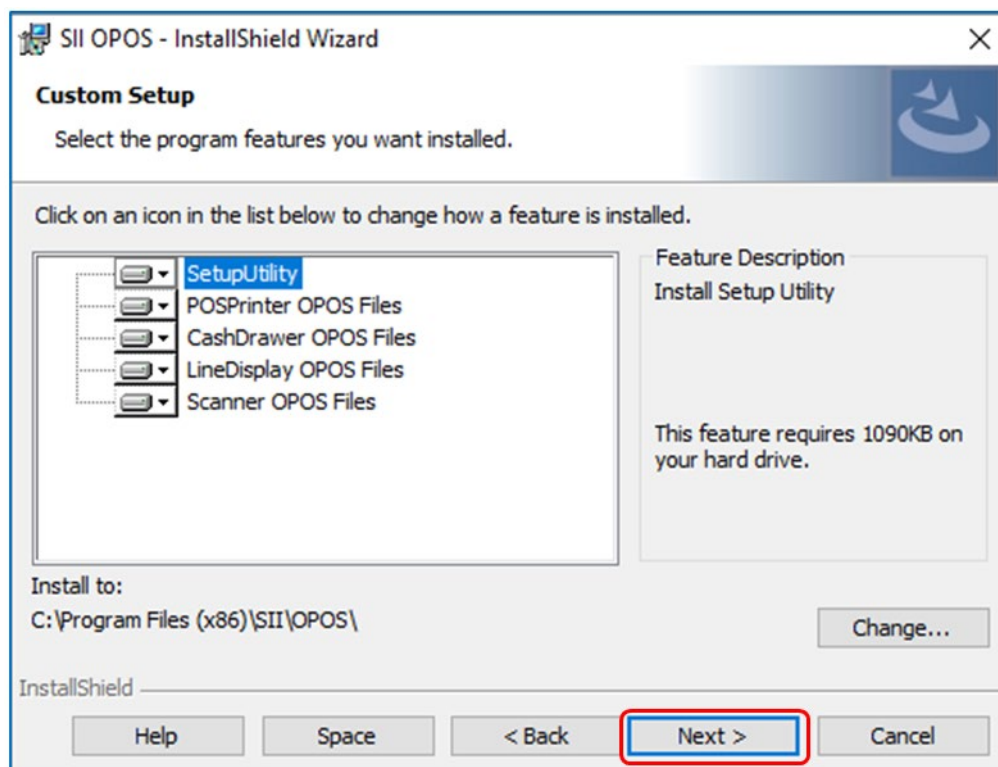


- (5) The window to select a setup type is displayed.
To install all features, select the “Complete” and click the [Next >] button.

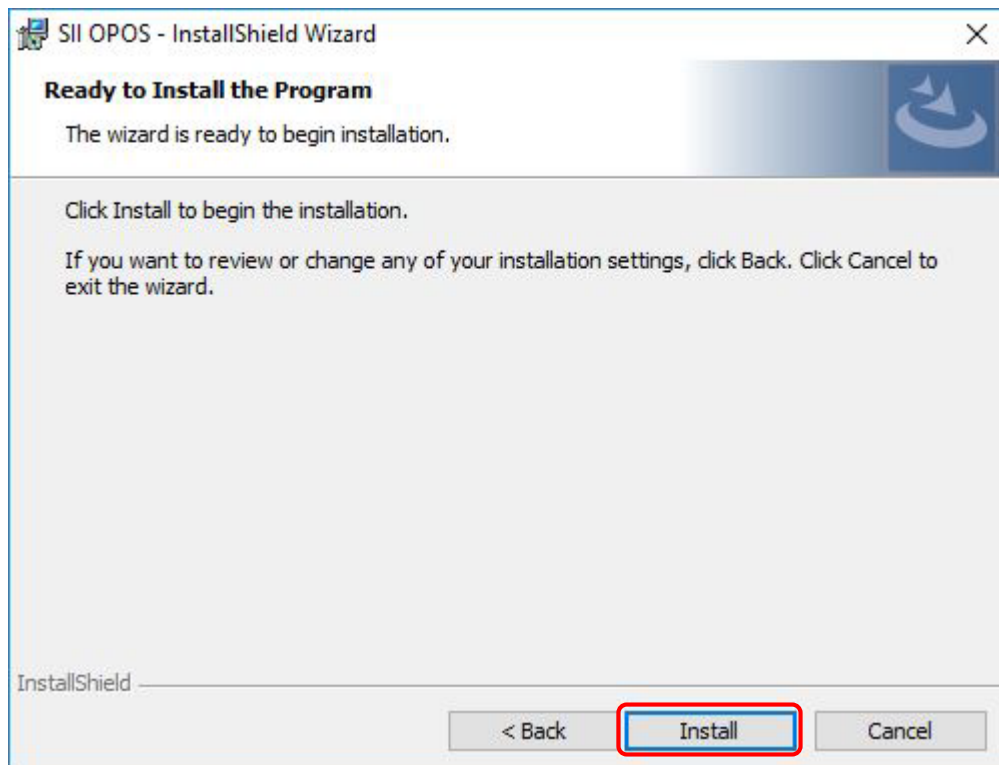


To select the installation features, select “Custom” and click the [Next >] button.

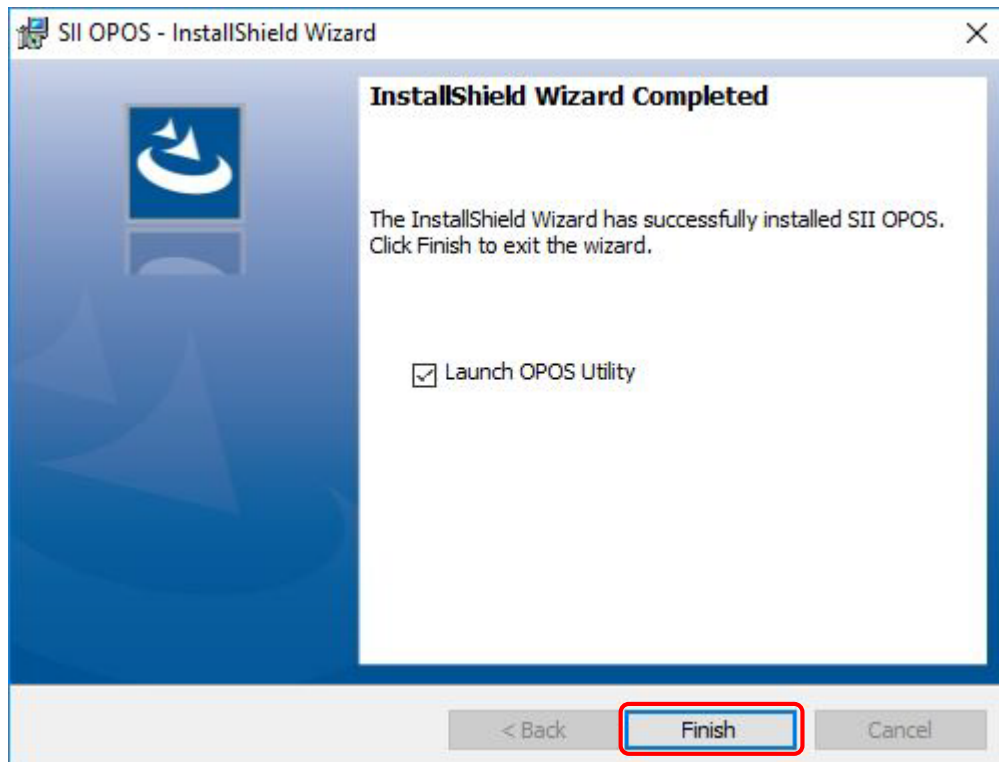
The Custom Setup wizard is displayed. Select the features to install and click the [Next >] button.



- (6) Click the [Install] button.



- (7) Click the [Finish] button. When clicking the [Finish] button with the "Launch OPOS Utility" checkbox on, the setup program ends and the configuration program (OPOS Utility V2) starts up.



This configuration program adds and deletes devices, and make settings for each device. After setting is completed, the setting data will be saved in the registry.

Please see "Chapter 3 How to Operate Configuration Program" about the configuration program. For the details about the registry, please refer to "Chapter 6 Registry Used by This Software".

2.2 Uninstallation

When the software is no longer used, click "Uninstall a program" in [Programs and Features] in the Control Panel. When the [Uninstall or change a program] window is displayed, select [SII OPOS], and click the [Uninstall] button.

2.3 Silent Installation

There are two installation ways in silent mode: by installing all features or by specifying features. The command to be used for the installation in silent mode is the following.

```
SetupOpos.exe /s /v"/qn [ADDLOCAL=StandardFiles[,POSPrinter][,CashDrawer][,LineDisplay][,Scanner]]"
```

The examples of the command are described below.

- **For installing all features in silent mode:**

```
SetupOpos.exe /s /v"/qn"
```

- **For installing POSPrinter and CashDrawer in silent mode:**

```
SetupOpos.exe /s /v"/qn ADDLOCAL=StandardFiles,POSPrinter,CashDrawer"
```

Caution

- ◆ Do not input space after the /v (slash v) of the argument.
- ◆ To specify a features by using ADDLOCAL, the writing of StandardFiles is necessary.
- ◆ To specify a features by using ADDLOCAL, do not include blank characters.
- ◆ When this software (Ver.1.31 or the previous version) has already been installed and the custom installation is to be selected, uninstall the software of Ver.1.31 or the previous version.

Reference

- The GUI of the installer is not displayed at the installation in silent mode but the dialog to confirm the account control of Windows is displayed.
- The commands are case-sensitive.

2.4 Silent Uninstallation

The procedure of the uninstallation in silent mode is described below.

Execution file	Description
MsiExec.exe	This file is executable in spite of the versions of setup program (SetupOpos.exe) and installed software. See "2.4.1 MsiExec.exe File" for details.
SetupOpos.exe	This file is executable when the versions of your setup program (SetupOpos.exe) and installed software are the same. See "2.4.2 SetupOpos.exe File" for details.

2.4.1 MsiExec.exe File

Startup the command prompt in administrator mode and execute following command.

```
MsiExec.exe /qn /x {9C13F697-C8FC-4B9C-BF2E-95DB8C9E6773}
```

Caution

- ◆ Uninstallation is not executed if the uninstallation is not executed by the command prompt in administrator mode.

2.4.2 SetupOpos.exe File

Startup the command prompt in administrator mode and execute the following command.

```
SetupOpos.exe /s /x /v"/qn"
```

Caution

- ◆ Do not input space after the /v (slash v) of the argument.

Reference

- When the file is not executed by the command prompt in administrator mode, the dialog to confirm the account control of Windows is displayed.
- The uninstallation using the MsiExec.exe file can be tried when the uninstallation is unable.

Chapter 3 How to Operate Configuration Program

This chapter describes the configuration program provided by this software.

3.1 Startup

The startup procedure of the configuration program is described.

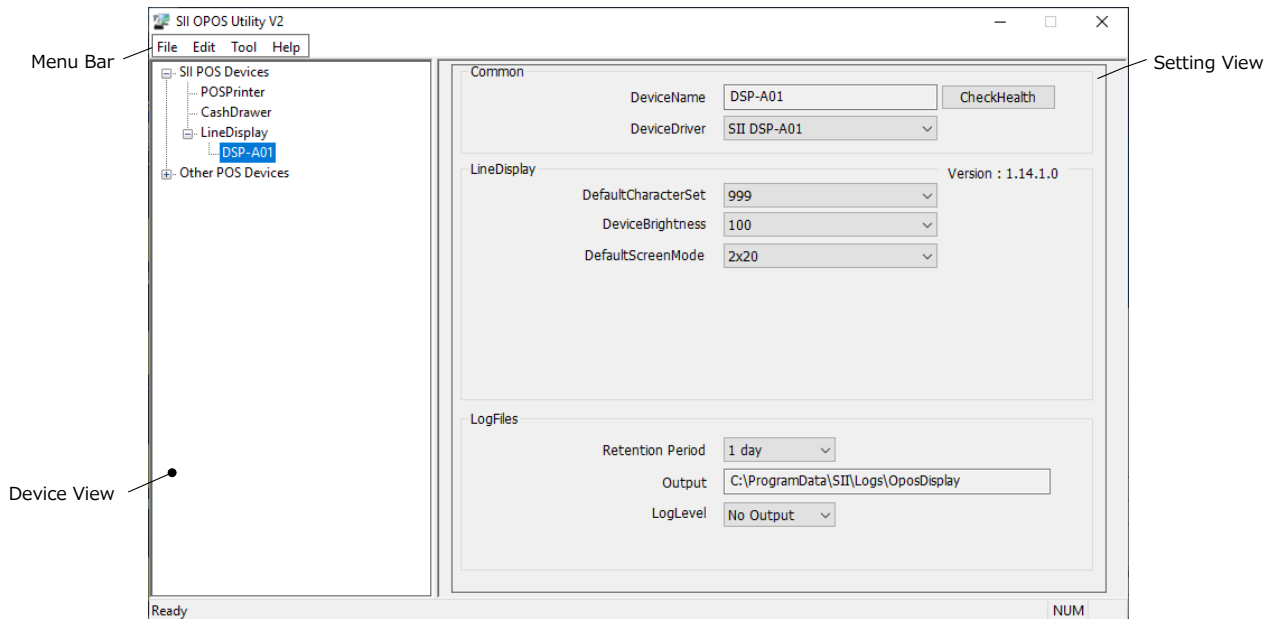
- For Windows 11:
Select [All apps] - [OPOS Utility V2] from the Start menu, and then the configuration program starts up.
- For Windows 10:
Select [SII OPOS] - [OPOS Utility V2] from the Start menu, and then the configuration program starts up.

Caution

- ◆ This software operates using a printer driver. The printer driver is required to be installed on the computer for using this software.
- ◆ Using this software requires logon to the computer with administrator privileges.

3.2 Screen Layout

The screen layout of the configuration program is described.



Item	Description
Menu Bar	The menu bar of the configuration program. See "3.2.1 Menu Bar" for items in the menu bar.
Device View	The type, the name, and the logical name of the device registered in the system are displayed in a tree.
Setting View	Displays setting contents of the device selected in "Device View". See "3.2.3(1) Display setting items" for items of device.

3.2.1 Menu Bar

Item		Description
File	Exit	Ends the configuration program.
	Add Device...	Adds a new device.
	Add Logical Device Name...	Adds a new logical device name.
Edit	Delete	Deletes the device being selected.
	CheckHealth	Executes an interactive test for the device being selected.
Help	About SIIOPosUtility...	Displays the version information of the configuration program.

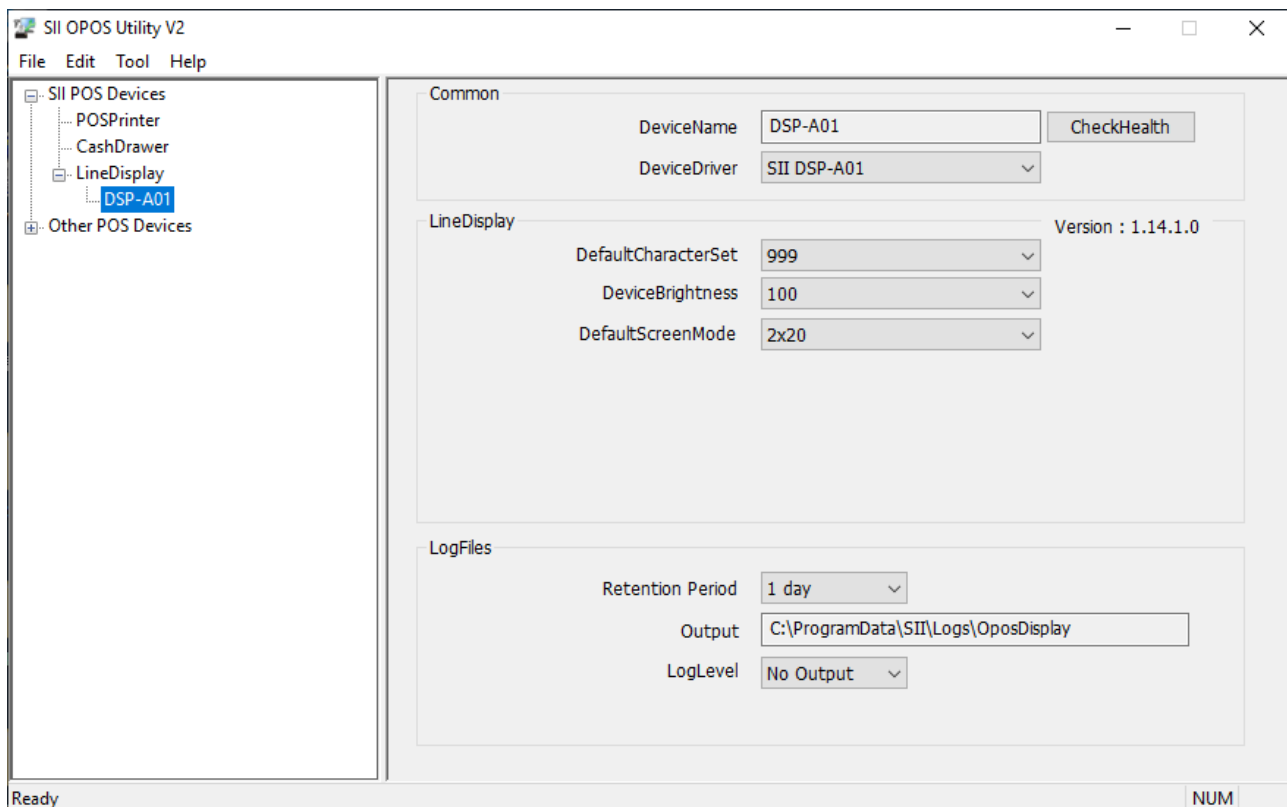
3.2.2 Device View

Name	Description
SII POS Devices	Displays SII devices. When a device name is selected in "Device View", the device can be deleted and the logical device name can be added.
Other POS Devices	Displays devices other than SII devices. Device settings cannot be changed or deleted.

3.2.3 Setting View

(1) Display setting items

The items displayed in "Setting View" and setting contents are described below.



Item	Description	Setting Content (" " : Default)
Common		
DeviceName	Device name	Device name of Display selected in "Device View".
CheckHealth	Executes an interactive test for the device being selected.	-
DeviceDriver	SII driver used for communication with Display	-

Item	Description	Setting Content (" " : Default)
LineDisplay		
DefaultCharacterSet	Character set type CharacterSet is initialized with this value. See CharacterSet for details.	437 737 850 852 855 857 858 860 863 865 866 932*1 999*2 1250 1251 1252 1253 1254
DeviceBrightness	Brightness setting	0 10 20 30 40 50 60 70 80 90 100
DefaultScreenMode	Screen mode of the device Number of rows × number of columns	2×20 5×20 2×40 5×40 8×40
LogFiles		
Retention Period	Retention period for log files Log files past the retention period are deleted when logs are output. The actual retention period may be longer by 1 day at maximum.	1 day 3 days 10 days 30 days 90 days
Output	Log output directory (Unchangeable) The log output directory and the file name are as follows. Output Directory: <System Drive>:\ProgramData\SII\Logs\LineDisplay The output directory cannot be changed. File Name: <yyyyMMdd>.log The maximum size of the log file is 32 MB. When the log file exceeds the maximum size, the file name is changed to <yyyyMMdd_hhmmssfff>.log, and a new <yyyyMMdd>.log is created.*3	

Item		Description	Setting Content (" " : Default)
	LogLevel	Log output level of Line Display Service Object	<p>No Output: No logs are output.</p> <p>Error: Error logs at execution are output.</p> <p>Trace: Detailed operation history and dump logs are output.</p>

*1: Default for Japanese

*2: Default for English

*3: The symbols used for the file name mean as follows. Each value comes from the Windows system clock.

yyyy : Year
 MM : Month
 dd : Day
 hh : Hour
 mm : Minute
 ss : Second
 fff : Millisecond

3.3 Functions

The functions of the configuration program are described.

3.3.1 Addition of Device

The procedure for adding a device is described.

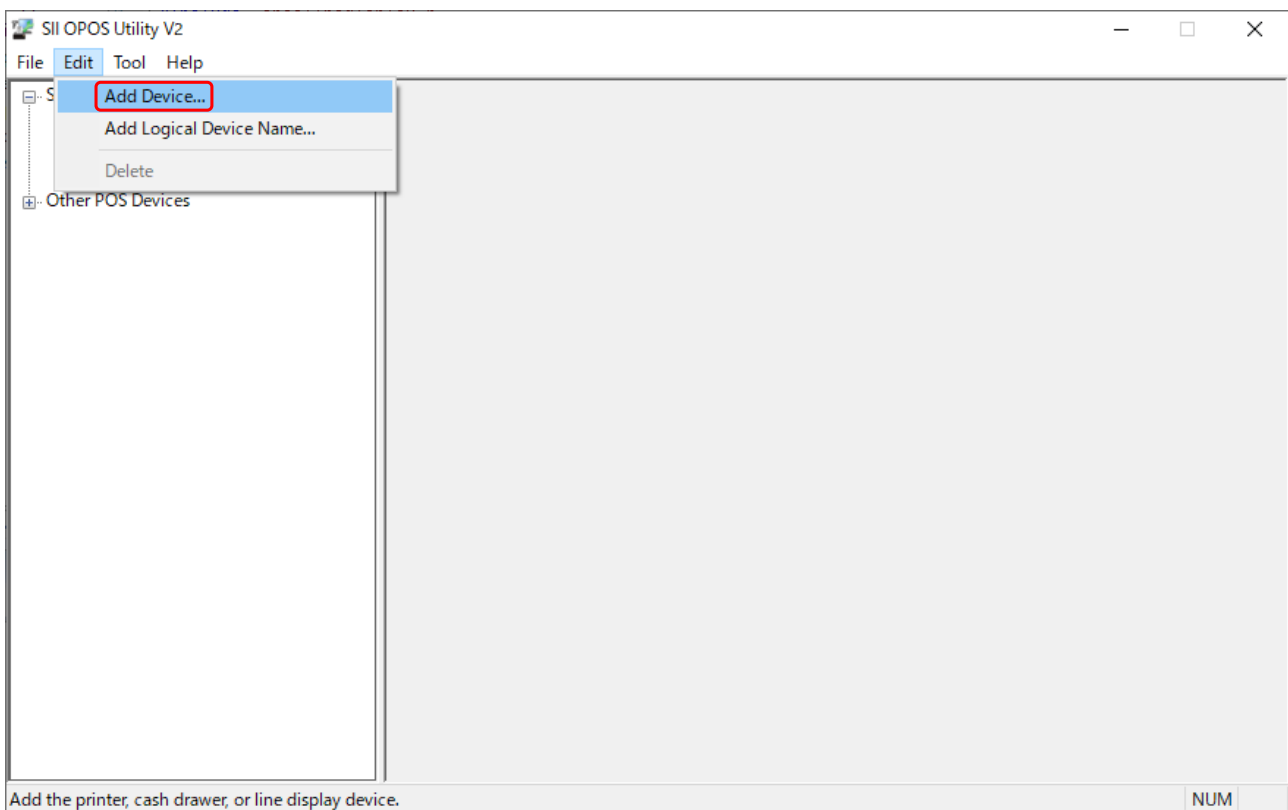
When the configuration program is started up immediately after installing this software, a device needs to be added since no device has been added.

When a new device is added, it is necessary to install the driver for the communication port to be used in advance. For installation of the driver, see "SII Printer Driver for Windows User's Guide" described in "1.1 Configuration".

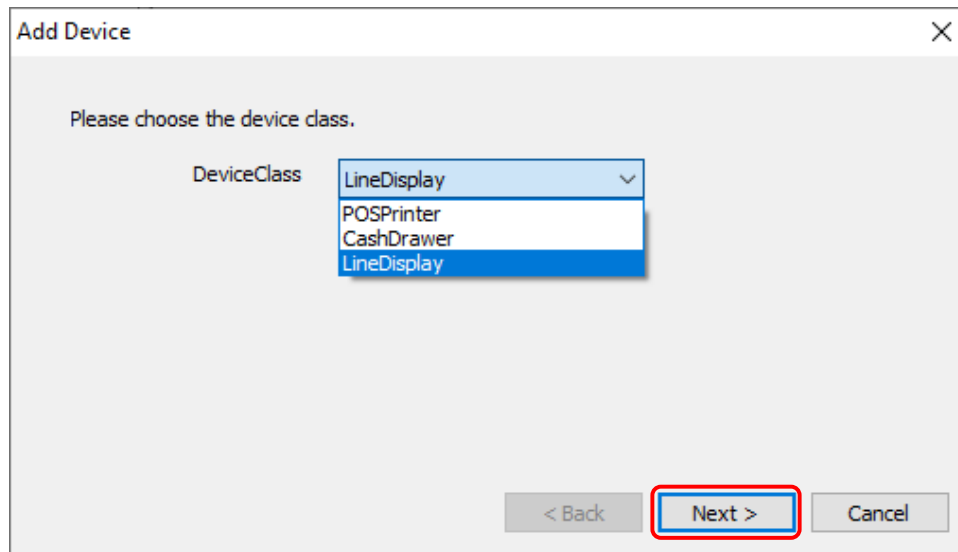
Maximum 99 Displays can be added.

(1) Addition of Display

- 1) When the configuration program starts, the following window is displayed. Select [Edit] – [Add Device...] from "Menu Bar".

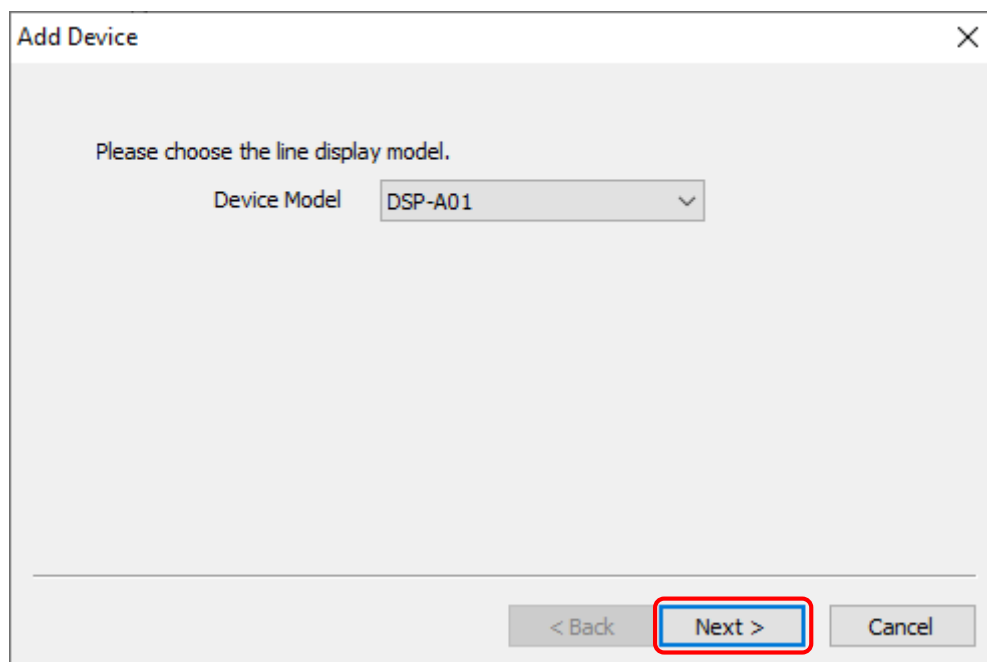


- 2) Select "LineDisplay" for [DeviceClass], then click the [Next >] button.



The screenshot shows a dialog box titled "Add Device" with a close button (X) in the top right corner. Inside the dialog, the text "Please choose the device class." is displayed. Below this text is a label "DeviceClass" followed by a dropdown menu. The dropdown menu is open, showing four options: "LineDisplay", "POSPrinter", "CashDrawer", and "LineDisplay". The first "LineDisplay" option is highlighted in blue. At the bottom of the dialog, there are three buttons: "< Back", "Next >", and "Cancel". The "Next >" button is highlighted with a red rectangle.

- 3) Select Display to be added from [Device Model], and click the [Next >] button.



The screenshot shows a dialog box titled "Add Device" with a close button (X) in the top right corner. Inside the dialog, the text "Please choose the line display model." is displayed. Below this text is a label "Device Model" followed by a dropdown menu. The dropdown menu is open, showing one option: "DSP-A01". At the bottom of the dialog, there are three buttons: "< Back", "Next >", and "Cancel". The "Next >" button is highlighted with a red rectangle.

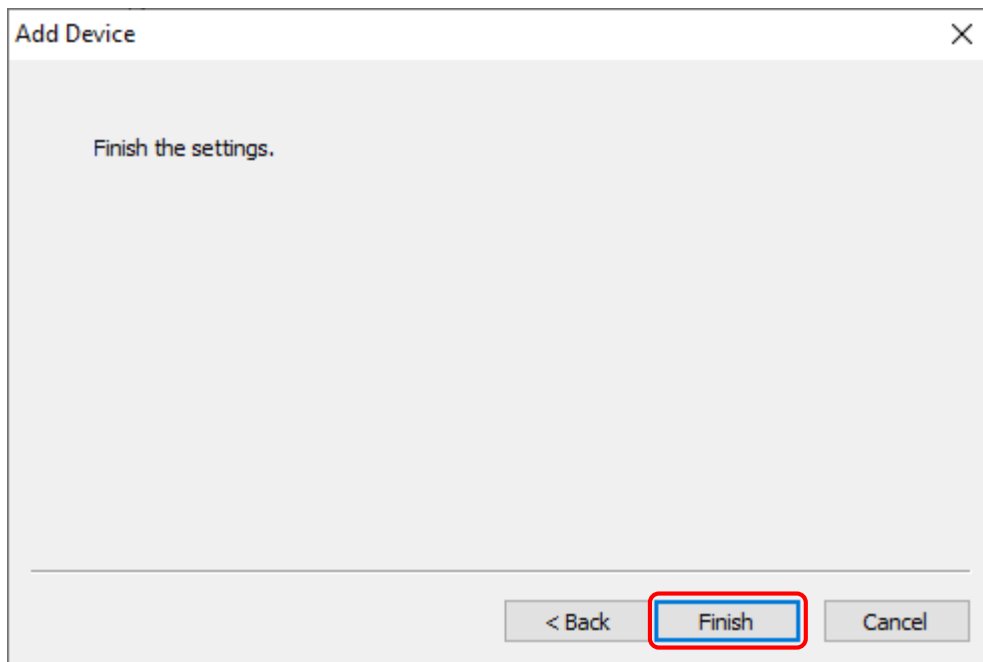
- 4) Select the driver to be used for the device selected in the description 3) from [Device Driver].
When Display is used via a printer, select the driver compatible with connected printer.
When Display is used alone, select the driver compatible with Display.
Click the [Next >] button after selecting the driver. [New DeviceName] is set automatically.

The screenshot shows a dialog box titled "Add Device" with a close button (X) in the top right corner. The main text reads "Please set the line display device driver." Below this, there are two fields: "New DeviceName" with the value "DSP-A01" and "Device Driver" with a dropdown menu showing "SII DSP-A01". At the bottom, there are three buttons: "< Back", "Next >" (highlighted with a red rectangle), and "Cancel".

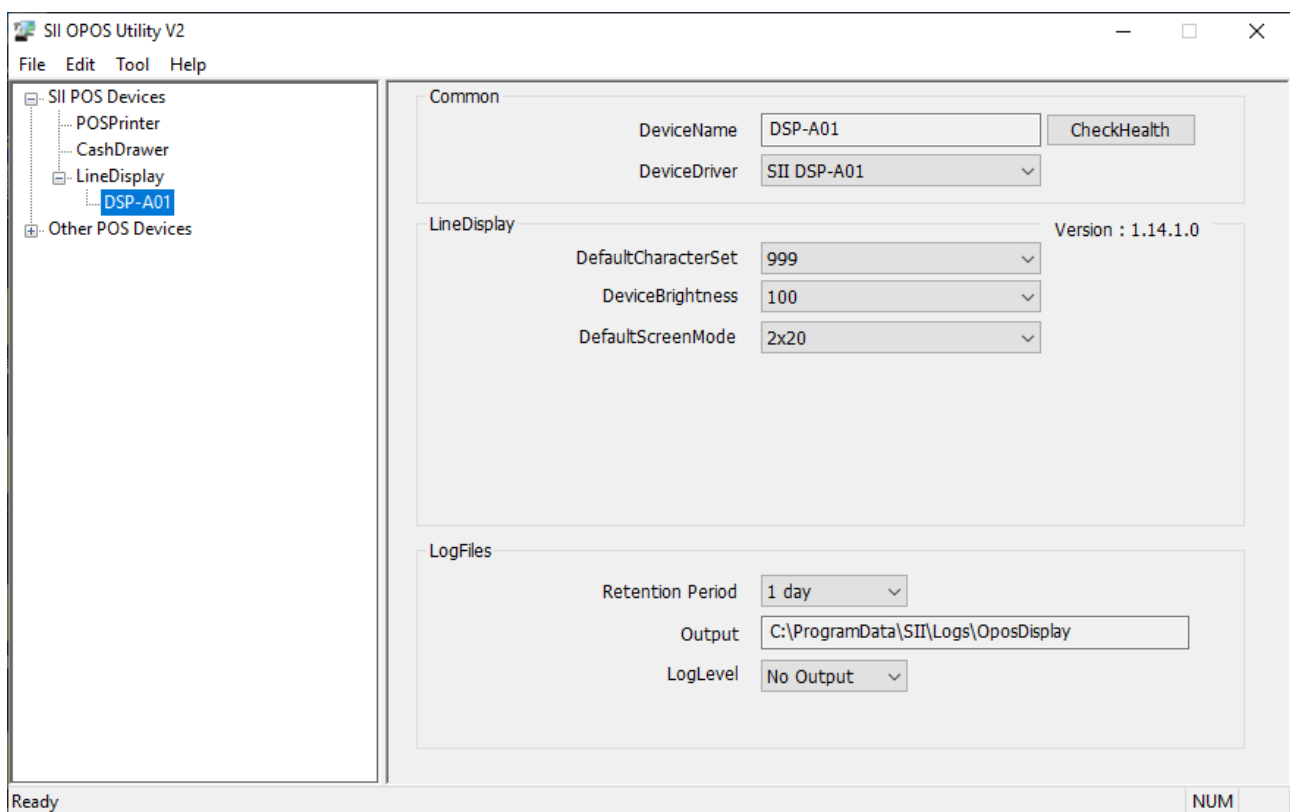
- 5) Enter or select the settings of Display, and then click the [Next >] button.

The screenshot shows the same "Add Device" dialog box, but now it displays "LineDisplay settings." Below this, there are three dropdown menus: "DefaultCharacterSet" set to "999", "DeviceBrightness" set to "100", and "DefaultScreenMode" set to "2x20". At the bottom, the same three buttons are present: "< Back", "Next >" (highlighted with a red rectangle), and "Cancel".

6) Click the [Finish] button.



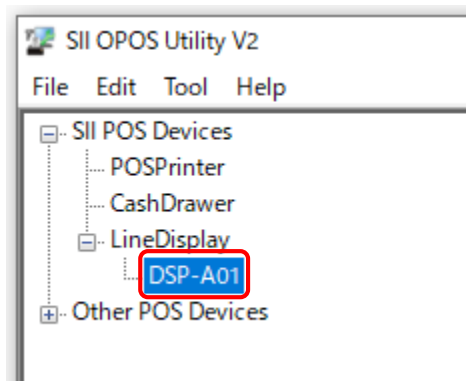
7) Confirm the contents in "Setting View".



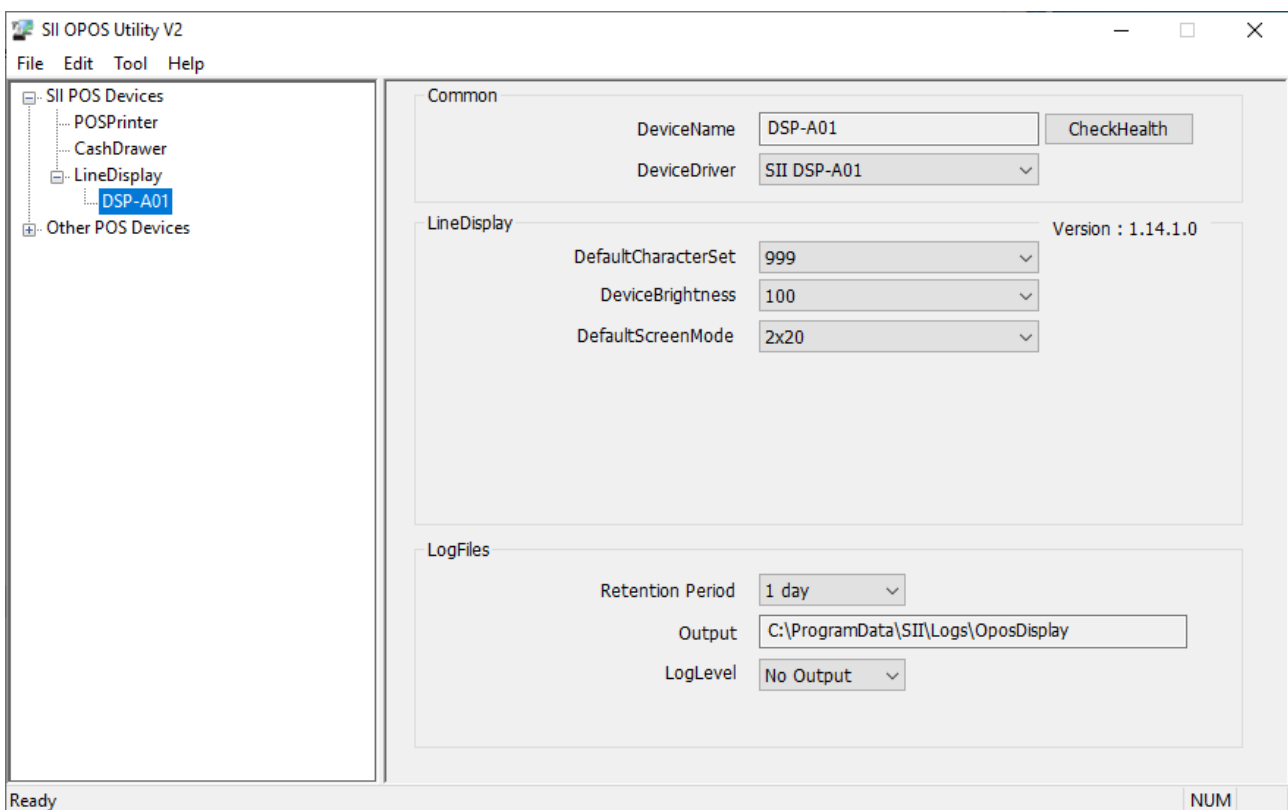
3.3.2 Changing Device Settings

The procedure for changing settings of the added device is described.

- 1) Select the device name from "Device View".



- 2) "Setting View" is displayed in editable state. Change the contents.



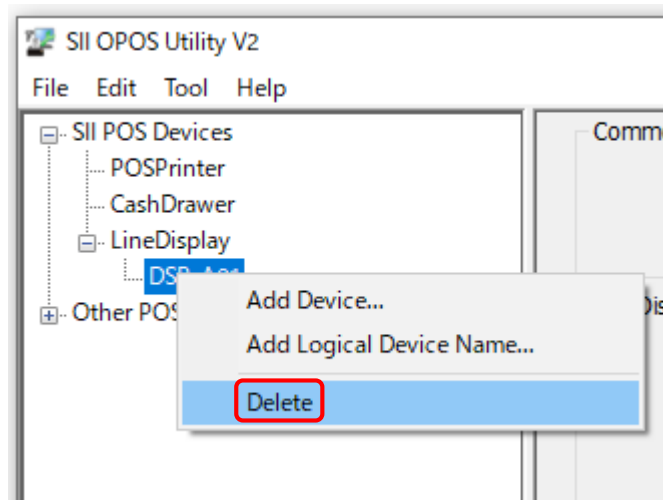
Reference

- When another device is selected or the configuration program is finished, the setting contents are saved in the registry.

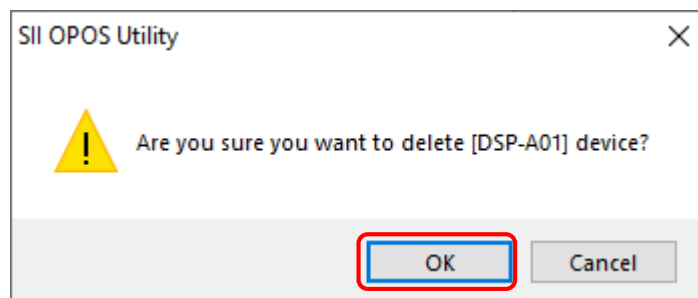
3.3.3 Deletion of Device

The procedure for deleting the added device is described below.

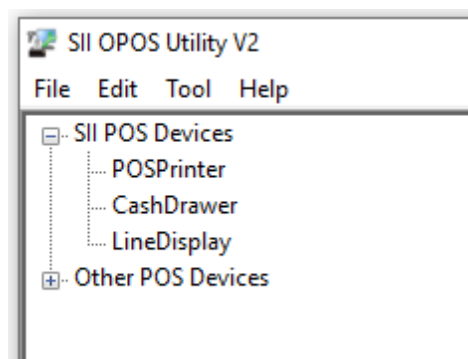
- 1) Select the device to be deleted from "Device View", and select [Delete] from the right-click menu.



- 2) Confirm the device name, and click the [OK] button.



- 3) Confirm that the selected device has been deleted from the "Device View".

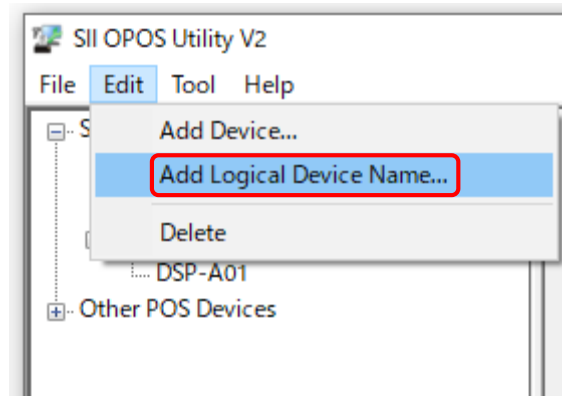


3.3.4 Adding and Deleting Logical Device Name

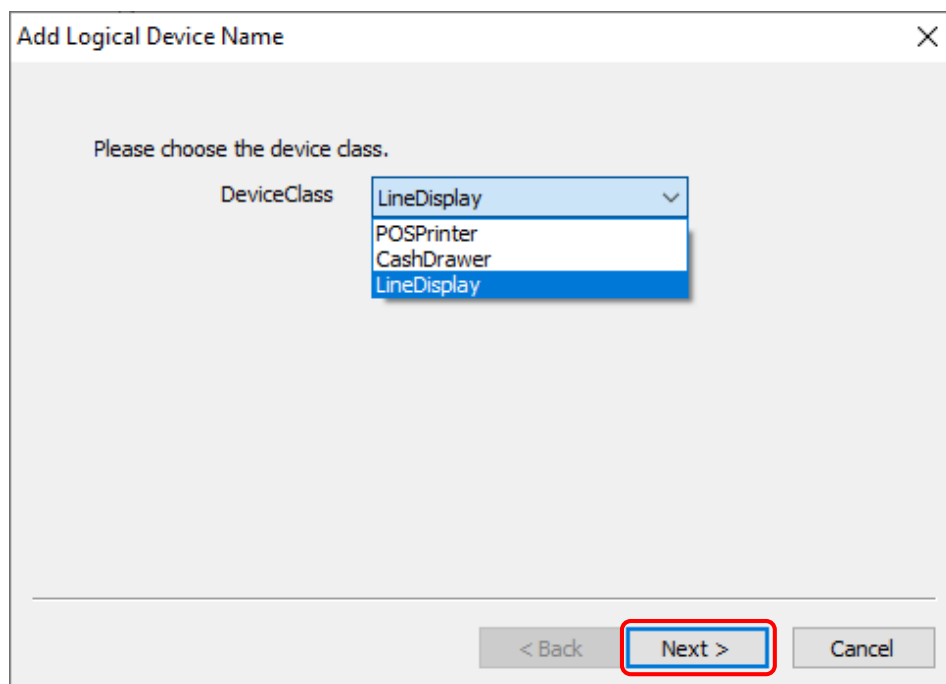
The procedures for adding and deleting the logical device name are described below.

(1) Addition of logical device name

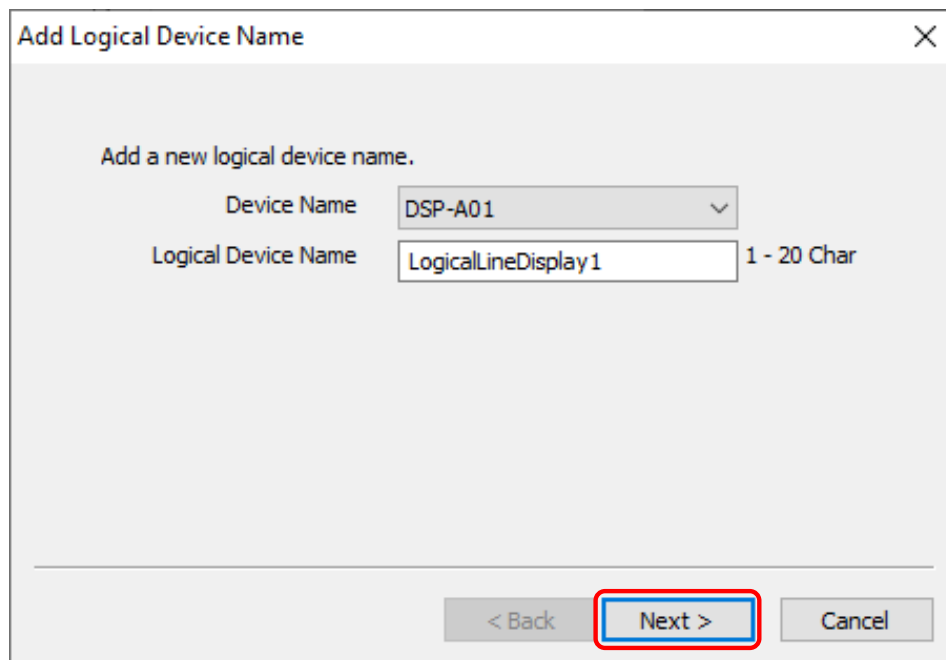
- 1) Start the configuration program, and select [Edit] – [Add Logical Device Name...] from "Menu Bar".



- 2) Select the target device in [DeviceClass], and click the [Next >] button.



- 3) Select the target device name in [Device Name], and enter the [Logical Device Name]. Click the [Next >] button.



Add Logical Device Name

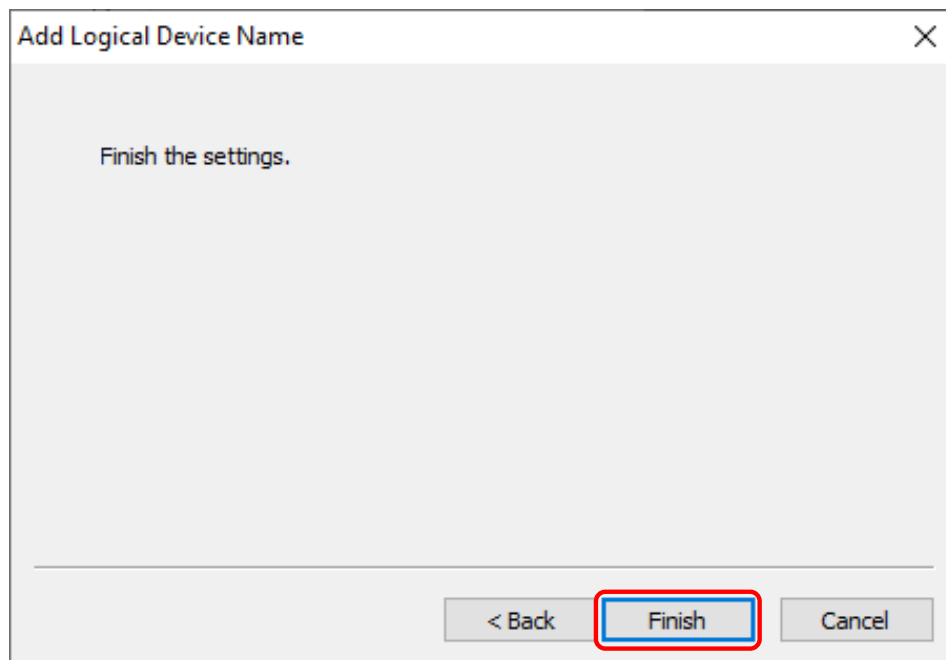
Add a new logical device name.

Device Name DSP-A01

Logical Device Name LogicalLineDisplay1 1 - 20 Char

< Back Next > Cancel

- 4) Click the [Finish] button.

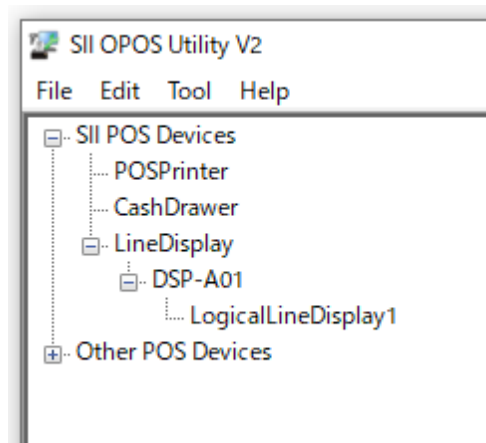


Add Logical Device Name

Finish the settings.

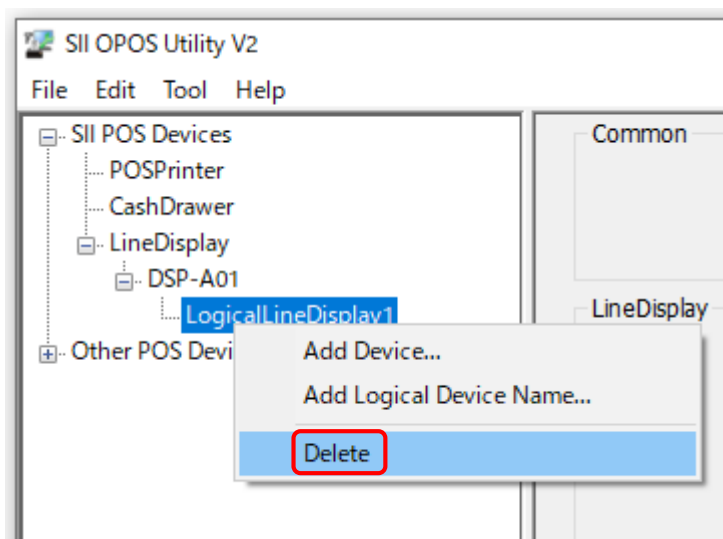
< Back Finish Cancel

- 5) Confirm the contents of "Device View".

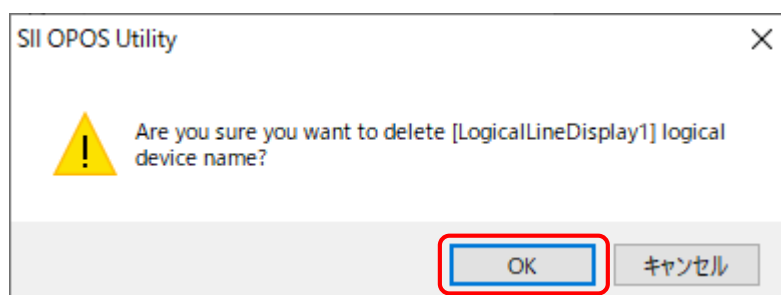


- (2) Deletion of Logical Device Name

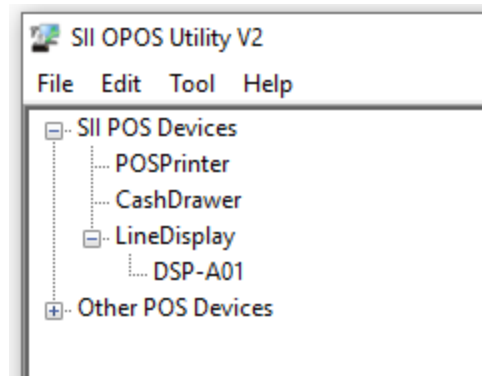
- 1) Select the logical device name to be deleted from "Device View", and select [Delete] from the right-click menu.



- 2) Confirm the logical device name, and click the [OK] button.



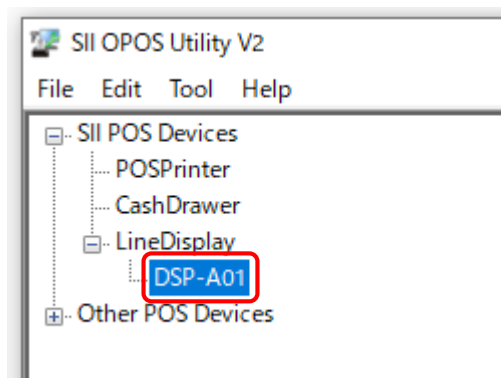
- 3) Confirm that the selected logical device name has been deleted from the "Device View".



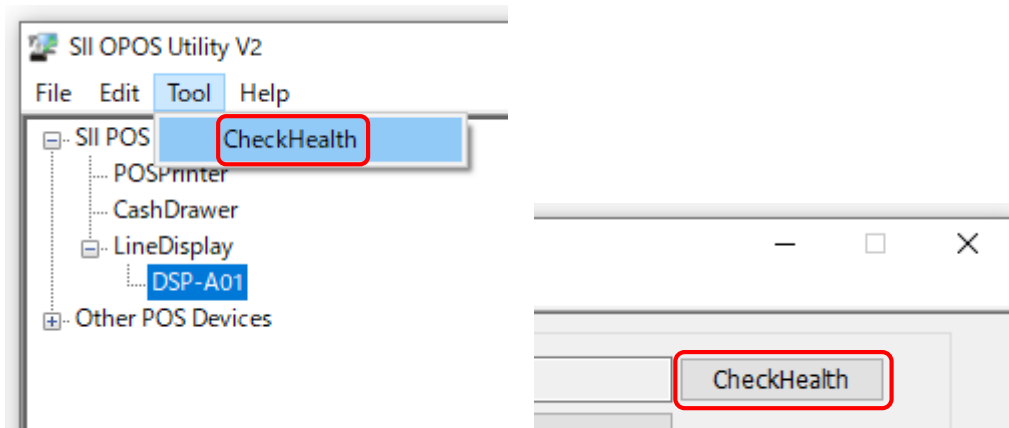
3.3.5 Device Interactive Test

In the configuration program, an interactive test can be performed for the device selected in "Device View". The procedure of the interactive test is described below.

- 1) Select the device name or the logical device name for which the interactive test is performed from "Device View".



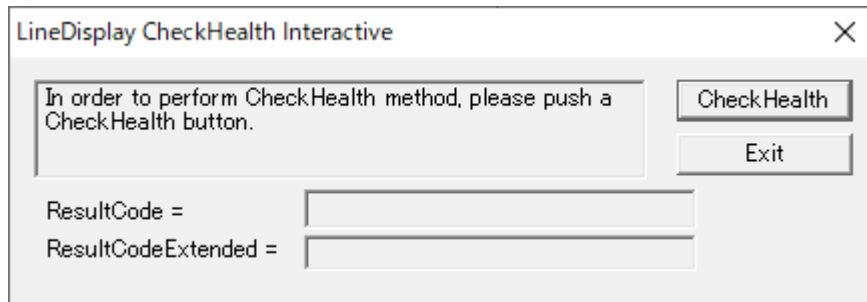
- 2) Select [Tool] menu - [CheckHealth], or click [CheckHealth] button in "Setting View".



- 3) The preparation for the interactive test is started.

[When the preparation for the interactive test succeeded]

- 4) The Display CheckHealth Interactive dialogue to perform the interactive test is displayed.

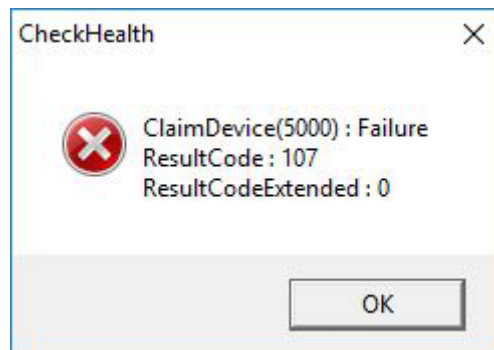


To start the interactive test, click the [CheckHealth] button.

To exit the interactive test, click the [Exit] button.

[When the preparation for the interactive test failed]

- 4) The error message is displayed.



Chapter 4 OPOS Control

This chapter describes the use procedure of the OPOS control.
Use the software in the following procedure for the application.

- (1) **Open:** Link the Control Object to the Service Project.
- (2) **ClaimDevice:** Call to gain exclusive access to the device.
- (3) **DeviceEnabled:** Set to TRUE to make the device operational.
- (4) Use the device (Properties, methods, and events).
- (5) **DeviceEnabled:** Set to FALSE to disable the device.
- (6) **ReleaseDevice:** Release exclusive access to the device.
- (7) **Close:** Release the Service Object from the Control Object.

Line Display Control supports Immediate Mode only.

The interface of the methods and properties related to Teletype Mode and Marquee Mode is provided, but the operation is not supported.

Display controlled by Line Display Control is a device used exclusively.

Chapter 5 Line Display Control Interface Specifications

This chapter describes the Line Display Control properties, methods and events implemented in this software.

5.1 Summary

5.1.1 Common Properties

Property Name	Type	Access	Availability Condition	Default
BinaryConversion	Long	R/W	Open	OPOS_BC_NONE(0)
CapCompareFirmwareVersion	Boolean	R	Open	FALSE
CapPowerReporting	Long	R	Open	OPOS_PR_NONE(0)
CapStatisticsReporting	Boolean	R	Open	FALSE
CapUpdateFirmware	Boolean	R	Open	FALSE
CapUpdateStatistics	Boolean	R	Open	FALSE
CheckHealthText	String	R	Open	""
Claimed	Boolean	R	Open	FALSE
ControlObjectDescription	String	R	-	"OPOS LineDisplay Control 1.14.xxx [Public, by CRM/MCS]"
ControlObjectVersion	Long	R	-	1014xxx
DeviceDescription	String	R	Open	"SII DSP-A01 Line Display"
DeviceEnabled	Boolean	R/W	Open & Claim	FALSE
DeviceName	String	R	Open	"DSP-A01 Line Display"
FreezeEvents	Boolean	R/W	Open	FALSE
OpenResult	Long	R	-	OPOS_SUCCESS(0)
PowerNotify	Long	R/W	Open	OPOS_PN_DISABLED(0)
PowerState	Long	R	Open	OPOS_PS_UNKNOWN(2000)
ResultCode	Long	R	-	OPOS_SUCCESS(0)
ResultCodeExtended	Long	R	Open	0
ServiceObjectDescription	String	R	Open	"SII DSP-A01 LineDisplay Service Object, Copyright(C) 20xx Seiko Instruments Inc."
ServiceObjectVersion	Long	R	Open	1014xxx
State	Long	R	-	OPOS_S_CLOSED(1)

5.1.2 Specific Properties

(When DefaultScreenMode=2×20, CharacterSet=999)

Property Name	Type	Access	Availability Condition	Default
CapBrightness	Boolean	R	Open	TRUE
CapCharacterSet	Long	R	Open	DISP_CCS_KANJI(11)
CapScreenMode	Boolean	R	Open	TRUE
CharacterSet	Long	R/W	Open, Claim, & Enable	999*1
CharacterSetList	String	R	Open	"437,737,850,852,855,857,858,860,863,865,866,932,999,1250,1251,1252,1253,1254"
Columns	Long	R	Open	The value of DeviceColumns
CursorColumn	Long	R/W	Open	0
CursorRow	Long	R/W	Open	0
CursorUpdate	Boolean	R/W	Open	TRUE
DeviceBrightness	Long	R/W	Open, Claim, & Enable	100*1
DeviceColumns	Long	R	Open	20
DeviceRows	Long	R	Open	2
DeviceWindows	Long	R	Open	0
Rows	Long	R	Open	The value of DeviceRows
ScreenMode	Long	R/W	Open & Claim	1*1
ScreenModeList	String	R	Open	"2x20, 5x20, 2x40, 5x40, 8x40"

*1: Can be modified by the configuration program.

The following specific properties are provided but the operation is not supported.

Property Name	Type	Access	Availability Condition	Default
BlinkRate	Long	R/W*1	Open	0
CapBitmap	Boolean	R	Open	FALSE
CapBlink	Long	R	Open	DISP_CB_NOBLINK(0)
CapBlinkRate	Boolean	R	Open	FALSE
CapCursorType	Long	R	Open	DISP_CCT_NONE(0x00000000)
CapCustomGlyph	Boolean	R	Open	FALSE
CapDescriptors	Boolean	R	Open	FALSE
CapICharWait	Boolean	R	Open	FALSE
CapHMarquee	Boolean	R	Open	FALSE
CapMapCharacterSet	Boolean	R	Open	FALSE
CapReadBack	Long	R	Open	DISP_CRB_NONE(0x00000000)
CapReverse	Long	R	Open	DISP_CR_NONE(0x00000000)
CapVMarquee	Boolean	R	Open	FALSE
CurrentWindow	Long	R/W*1	Open	0

Property Name	Type	Access	Availability Condition	Default
CursorType	Long	R/W ^{*1}	Open	DISP_CT_NONE(0)
CustomGlyphList	String	R	Open	""
DeviceDescriptors	Long	R	Open	0
GlyphHeight	Long	R	Open	24
GlyphWidth	Long	R	Open	24
InterCharacterWait	Long	R/W ^{*1}	Open	0
MapCharacterSet	Boolean	R/W ^{*1}	Open	FALSE
MarqueeFormat	Long	R/W ^{*1}	Open	DISP_MF_WALK(0)
MarqueeRepeatWait	Long	R/W ^{*1}	Open	0
MarqueeType	Long	R/W ^{*1}	Open	DISP_MT_NONE(0)
MarqueeUnitWait	Long	R/W ^{*1}	Open	0
MaximumX	Long	R	Open	0
MaximumY	Long	R	Open	0

^{*1}: OPOS_SUCCESS(0) is stored in **ResultCode** only when the default is set.
When a value other than the default is set, OPOS_E_ILLEGAL(106) is stored in **ResultCode**.

5.1.3 Common Methods

Method Name	Availability Condition
CheckHealth	Open, Claim, & Enable
ClaimDevice	Open
Close	Open
CompareFirmwareVersion	Open, Claim, & Enable
DirectIO	Open, Claim, & Enable ^{*1}
Open	-
ReleaseDevice	Open & Claim
ResetStatistics	Open, Claim, & Enable
RetrieveStatistics	Open, Claim, & Enable
UpdateFirmware	Open, Claim, & Enable
UpdateStatistics	Open, Claim, & Enable

^{*1}: The availability condition differs from that of UPOS V 1.14.

5.1.4 Specific Methods

Method Name	Availability Condition
ClearText	Open, Claim, & Enable
DisplayText	Open, Claim, & Enable
DisplayTextAt	Open, Claim, & Enable

The following specific properties are provided but the operation is not supported. OPOS_E_ILLEGAL(106) is returned and also stored in **ResultCode**.

Event Name	Occurrence Condition
ClearDescriptors	Open, Claim, & Enable
CreateWindow	Open, Claim, & Enable
DefineGlyph	Open, Claim, & Enable
DestroyWindow	Open, Claim, & Enable
DisplayBitmap	Open, Claim, & Enable
ReadCharacterAtCursor	Open, Claim, & Enable
RefreshWindow	Open, Claim, & Enable
ScrollText	Open, Claim, & Enable
SetBitmap	Open, Claim, & Enable
SetDescriptor	Open, Claim, & Enable

5.1.5 Events

Events are not supported.

5.2 Display Data and Escape Sequences

5.2.1 Escape Sequence Operated When Specified

Name	Data	Remarks
Display bitmap	ESC #B	Not supported.

5.2.2 Escape Sequence Operated When Displayed

Name	Data	Remarks
Reverse video	ESC vC	Not supported.
Blink	ESC kC	Not supported.
Normal	ESC N	Not supported.

5.3 Common Properties

BinaryConversion Property R/W

Syntax **LONG BinaryConversion;**

Remarks OPOS passes multi-character input and output using BStrings. BStrings may be safely used for text data. As the BStrings are passed between the application and the OPOS Control, OLE may execute language-specific translations to or from Unicode. When BStrings are used to pass binary data, then these translations may alter the data such that the data byte in a BString character at the application does not match the corresponding byte at the Control. This mismatch is more likely when BString pointers are used, since the Unicode characters are presented to the application and/or OPOS Control, and a language difference between them may cause misinterpretation.

Characters between 0x00 and 0x7F may be sent without fear of language-specific translation. Only characters between 0x80 and 0xFF sometimes cause incorrect translations.

The values of **BinaryConversion** are as follows.

Value	Meaning
OPOS_BC_NONE(0)	Data is placed one byte per BString character, with no conversion.
OPOS_BC_NIBBLE(1)	Each byte is converted into two characters (This option provides for the fastest conversion between binary and ASCII characters). First character = 0x30 + bits 7-4 of the data byte. Second character = 0x30 + bits 3-0 of the data byte. Example: Byte value 154 = 0x9A is converted to the characters 0x39 0x3A (= string "9:"). Note that this conversion is not the more common hexadecimal ASCII, which would have converted 154 to 0x39 0x41 (= string "9A").
OPOS_BC_DECIMAL(2)	Each byte is converted into three characters. VAL (string) may be used on each 3 characters to convert from ASCII to binary. RIGHT ("^^"+STR(byte),3) may be used to produce 3 ASCII characters from each byte, where '^' represents the space character. Example 1: Byte value 154 = 0x9A becomes the characters 0x31 0x35 0x34 (= string "154"). Example 2: Byte value 8 becomes the characters 0x30 0x30 0x38 (= string "008").

When **BinaryConversion** is on (that is, not OPOS_BC_NONE(0)) and the property or method parameter description specifies that **BinaryConversion** applies, before setting the property or passing the method parameter, convert the string data into the format specified by the **BinaryConversion** value.

This property is initialized to OPOS_BC_NONE(0) by **Open**.

Return When this property is set, one of the following values is placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	An illegal value was specified.

CapCompareFirmwareVersion Property

Syntax **BOOL CapCompareFirmwareVersion;**

Remarks Indicates the function that compares firmware version.
The following table shows the valid property values.

Value	Meaning
FALSE	This property is not supported.

This property is initialized to FALSE by **Open**.

CapPowerReporting Property

Syntax **LONG CapPowerReporting;**

Remarks Identifies the reporting capabilities of the device.
The following table shows the valid property values.

Value	Meaning
OPOS_PR_NONE	This property is not supported.

This property is initialized to OPOS_PR_NONE(0) by **Open**.

CapStatisticsReporting Property

Syntax **BOOL CapStatisticsReporting;**

Remarks Identifies the reporting capabilities of the device.
The following table shows the valid property values.

Value	Meaning
FALSE	No statistical data regarding the device is available.

This property is initialized to FALSE by **Open**.

CapUpdateFirmware Property

Syntax **BOOL CapUpdateFirmware;**

Remarks Indicates whether the device supports firmware updating.
The following table shows the valid property values.

Value	Meaning
FALSE	This property is not supported.

This property is initialized to FALSE by **Open**.

CapUpdateStatistics Property

Syntax **BOOL CapUpdateStatistics;**

Remarks Indicates the function that some or all device statistics can be reset.
The following table shows the valid property values.

Value	Meaning
FALSE	There are no statistics that the application can reset/change.

This property is initialized to FALSE by **Open**.

CheckHealthText Property

Syntax **BSTR CheckHealthText;**

Remarks Holds the results of the immediately preceding call to the **CheckHealth**.
The following examples show the results of diagnosis.

Value	Meaning
"Internal HCheck: Successful"	Succeeded in the health check without using the device.
"Internal HCheck: Failure"	Failed in the health check without using the device.
"External HCheck: Successful"	Succeeded in the communication confirmation with Display and displaying using the device.
"External HCheck: Failure"	Failed in the communication confirmation with Display and displaying using the device.
"Interactive HCheck: Successful"	Succeeded in the interactive test of the device.
"Interactive HCheck: Failure"	Failed in the interactive test of the device.
"Interactive HCheck: Canceled"	For the interactive test of the device, the dialog is closed without testing.

This value is initialized to an empty string before the first call to **CheckHealth**.

Claimed Property

Syntax **BOOL Claimed;**

Remarks Indicates whether the device is claimed for exclusive access.
The following table shows the valid property values.

Value	Meaning
TRUE	The device is claimed for exclusive access
FALSE	The device is released for sharing with other applications.

This property is initialized to FALSE by **Open**.

ControlObjectDescription Property

Syntax **BSTR ControlObjectDescription;**

Remarks Identifies the Control Object by this property.
"OPOS LineDisplay Control 1.14.xxx [Public, by CRM/MCS]" is set.

This property is always readable.

ControlObjectVersion Property

Syntax **LONG ControlObjectVersion;**

Remarks This property holds the Control Object version number.

This property is always readable.

DeviceDescription Property

Syntax **BSTR DeviceDescription;**

Remarks This property identifies devices and related information.
The following table shows the valid property values.

Display	Default
DSP-A01	"SII DSP-A01 Line Display"

DeviceEnabled Property R/W

Syntax **BOOL DeviceEnabled;**

Remarks Selects Enable/Disable of the device.
The following table shows the valid property values.

Value	Meaning
TRUE	The device has been placed in an operational state. If changed to TRUE, then the device is brought to an operational state.
FALSE	The device has been disabled. If changed to FALSE, then the device is physically disabled.

The application must set this property to TRUE before using output devices.

This property is initialized to FALSE by **Open**.

Return When this property is set, one of the following values is placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_NOTCLAIMED(103)	To enable an exclusive use device, its exclusive access must be obtained previously.
OPOS_E_NOHARDWARE(107)	When the device is used via a printer: It is in one of the following states. <ul style="list-style-type: none">· The printer is not powered on· The printer and the host are not connected· Display is not connected to the printer (DeviceEnabled only) When the device is used alone: Display and the host are not connected.
OPOS_E_FAILURE(111)	The device cannot be enabled. Setting information and the information from device may be different.

DeviceName Property

Syntax **BSTR DeviceName;**

Remarks This property identifies devices and related information.
The following table shows the valid property values.

Display	Default
DSP-A01	"DSP-A01 Line Display"

FreezeEvents Property R/W

Syntax **BOOL FreezeEvents;**

Remarks Selects whether to notify events.
The following table shows the valid property values.

Value	Meaning
FALSE	This property is not supported.

This property is initialized to FALSE by **Open**.

OpenResult Property

Syntax **LONG OpenResult;**

Remarks Holds additional details about the most recent **Open**. The following table shows the valid property values.

Value	Meaning
OPOS_SUCCESS(0)	Successful open.
OPOS_OR_ALREADYOPEN(301)	Already open.
OPOS_OR_REGBADNAME(302)	The registry does not contain a key for the specified device name. Or, the device name is not specified.
OPOS_OR_REGPROGID(303)	Could not read the device name key's default, or could not convert the Programmatic ID it holds into a valid Class ID.
OPOS_OR_CREATE(304)	Could not create a Service Object instance, or could not get its IDispatch interface.
OPOS_OR_BADIF(305)	The Service Object does not support one or more of the methods required by its release. The setting of device name may be different from the Service Object.

This property is initialized to OPOS_SUCCESS(0) by **Open**.

PowerNotify Property R/W

Syntax **LONG PowerNotify;**

Remarks Contains the type of power notification selection made by the application. The following table shows the valid property values.

Value	Meaning
OPOS_PN_DISABLED(0)	This property is not supported.

This property is initialized to OPOS_PN_DISABLED (0) by **Open**.

PowerState Property

Syntax **LONG PowerState;**

Remarks Contains the current power condition of the device. The following table shows the valid property values.

Value	Meaning
OPOS_PS_UNKNOWN(2000)	This property is not supported.

This property is initialized to OPOS_PS_UNKNOWN(2000) by **Open**.

ResultCode Property

Syntax **LONG ResultCode;**

Remarks This property is set by each method. It is also set when a writable property is set.

This property is always readable. Before **Open** is called, it returns the value OPOS_E_CLOSED(101).

The following table shows the valid property values.

Value	Meaning
OPOS_SUCCESS(0)	Successful operation.
OPOS_E_CLOSED(101)	Attempt was made to access a closed device. This error is not mentioned in the description of property and method.
OPOS_E_NOTCLAIMED(103)*1	Attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
OPOS_E_NOSERVICE(104)	The Control cannot communicate with the Service Object. The software must be re-installed the software. See "2.2 Uninstallation" for the uninstallation procedure. See "2.1 Installation" for the installation procedure.
OPOS_E_DISABLED(105)*1	Cannot execute operation while device is disabled.
OPOS_E_ILLEGAL(106)	Attempt was made to execute an illegal or unsupported operation with the device, or an invalid parameter value was used.
OPOS_E_NOHARDWARE(107)	When the device is used via a printer: It is in one of the following states. <ul style="list-style-type: none">• The printer is not powered on• The printer and the host are not connected When the device is used alone: Display and the host are not connected.
OPOS_E_NOEXIST(109)	The file name (or other specified value) does not exist.
OPOS_E_FAILURE(111)	The device cannot execute the requested procedure, even though the device is connected to the system, powered on, and on-line.
OPOS_E_TIMEOUT(112)	The Service Object timed out waiting for a response from the device, or the data was unable to be transmitted to the device within the timeout.

*1: When multiple values of OPOS_E_NOTCLAIMED(103) and OPOS_E_DISABLED(105) are valid, the value of OPOS_E_NOTCLAIMED(103) has the high priority.

ResultCodeExtended Property

Syntax **LONG ResultCodeExtended;**

Remarks This property is initialized to 0 by **Open**. The extended error code is not supported.

Reference **ResultCode**

ServiceObjectDescription Property

Syntax **BSTR ServiceObjectDescription;**

Remarks Contains a string for identifying the service object to this property.
The following table shows the valid property values.

Display	Default
DSP-A01	"SII DSP-A01 LineDisplay Service Object, Copyright (C) 20xx Seiko Instruments Inc."

ServiceObjectVersion Property

Syntax **LONG ServiceObjectVersion;**

Remarks This property holds the Service Object version number.

This property is initialized by **Open**.

State Property

Syntax **LONG State;**

Remarks Contains the current state of the Control.
The following table shows the valid property values.

Value	Meaning
OPOS_S_CLOSED(1)	The Control is closed.
OPOS_S_IDLE(2)	The Control is in a good state and is not busy.

This property is always readable.
This property is initialized to OPOS_S_IDLE(2) by **Open**.

5.4 Specific Properties

CapBrightness Property

Syntax **BOOL CapBrightness;**

Remarks Indicates whether the brightness can be controlled.
The following table shows the valid property values.

Value	Meaning
TRUE	The brightness control is supported.

This property is initialized to TRUE by **Open**.

CapCharacterSet Property

Syntax **LONG CapCharacterSet;**

Remarks Holds the character setting that can be displayed.
This property has the following value.

Value	Meaning
DISP_CCS_KANJI(11)	The character setting supports Code Page932, including ASCII characters 0x20 through 0x7F and the one-byte katakana characters 0xA1 through 0xDF. It also includes the Shift-JIS code characters defined in JIS 1st and 2nd levels.

This property is initialized to DISP_CCS_KANJI(11) by **Open**.

CapScreenMode Property

Syntax **BOOL CapScreenMode;**

Remarks Indicates whether the device can support changing the screen mode (for example, the number of text rows and columns).
The following table shows the valid property values.

Value	Meaning
TRUE	The screen mode changing is supported.

This property is initialized to TRUE by **Open**.

CharacterSet Property R/W

Syntax **LONG CharacterSet;**

Remarks Holds the character set for displaying characters.
One of the following values is set to this property:

Value	Meaning
437	Selects Code Page437 character set.
737	Selects Code Page737 character set.
850	Selects Code Page850 character set.
852	Selects Code Page852 character set.
855	Selects Code Page855 character set.
857	Selects Code Page857 character set.
858	Selects Code Page858 character set.
860	Selects Code Page860 character set.
863	Selects Code Page863 character set.
865	Selects Code Page865 character set.
866	Selects Code Page866 character set.
932	Selects Katakana as the Code Page932 character set (Shift-JIS Code).
PTR_CS_ANSI (999) / PTR_CS_WINDOWS (999)	Sets the Windows ANSI characters.*1
1250	Selects Code Page1250 character set.
1251	Selects Code Page1251 character set.
1252	Selects Code Page1252 character set.*1
1253	Selects Code Page1253 character set.
1254	Selects Code Page1254 character set.

*1: Windows ANSI character set is equal to Code Page1252 character set.

For this property, the default can be changed by setting of the configuration program.
This property is initialized to the value of character set which is set in [DefaultCharacterSet] of the configuration program when the device is first enabled.

Return When this property is set, one of the following values is placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
Other values	See the item of ResultCode .

CharacterSetList Property

Syntax **BSTR CharacterSetList;**

Remarks Holds the character set numbers.
"437,737,850,852,855,857,858,860,863,865,866,932,999,1250,1251,1252,1253,1254" is set.

This property is initialized to the above value by **Open**.

Columns Property

Syntax **LONG Columns;**

Remarks Holds the number of columns for this window.
This property is the same as **DeviceColumns**.

This property is initialized to the value of **DeviceColumns** by **Open**.

CursorColumn Property R/W

Syntax **LONG CursorColumn;**

Remarks Holds the column in the current window to which the next displayed character will be output.

Legal values range from 0 through **Columns**. (See the description of **CursorColumn = Columns** in **DisplayText**.)

This property is initialized to 0 by **Open** and **ClearText**.

If **CursorUpdate** is true, it is also updated when **DisplayText** or **DisplayTextAt** is called.

Return When this property is set, one of the following values is placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	An invalid cursor column value was specified.

CursorRow Property R/W

Syntax **LONG CursorRow;**

Remarks Holds the row in the current window to which the next displayed character will be output.

Legal values range from 0 through 1 less than **Rows**.

This property is initialized to 0 by **Open** and **ClearText**.

If **CursorUpdate** is true, it is also updated when **DisplayText** or **DisplayTextAt** is called.

Return When this property is set, one of the following values is placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	An invalid cursor column value was specified.

CursorUpdate Property R/W

Syntax **BOOL CursorUpdate;**

Remarks **CursorRow** and **CursorColumn** will be updated to point to the character beyond the last character output when characters are displayed using the **DisplayText** or **DisplayTextAt** method. The following table shows the valid property values.

Value	Meaning
TRUE	CursorRow and CursorColumn will be updated to point to the character beyond the last character output when characters are displayed using the DisplayText or DisplayTextAt method.
FALSE	The cursor properties will not be updated when characters are displayed.

This property is initialized to TRUE by **Open**.

Return When this property is set, following value is placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.

DeviceBrightness Property R/W

Syntax **LONG DeviceBrightness;**

Remarks Holds the device brightness value, expressed as a percentage between 0 and 100. The following table shows the valid property values.

Value	Meaning
0	0 : 0% (blank)
1 to 10	10: 10% intensity
11 to 20	20: 20% intensity
21 to 30	30: 30% intensity
31 to 40	40: 40% intensity
41 to 50	50: 50% intensity
51 to 60	60: 60% intensity
61 to 70	70: 70% intensity
71 to 80	80: 80% intensity
81 to 90	90: 90% intensity
91 to 100	100: 100% intensity

For this property, the default can be changed by setting of the configuration program. This property is initialized to the value of brightness which is set in [DeviceBrightness] of the configuration program when the device is first enabled.

Return When this property is set, one of the following values is stored in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	An invalid value was used: not in the range 0 to 100.

DeviceColumns Property

Syntax **LONG DeviceColumns;**

Remarks Holds the number of columns on this device.

This property is initialized by **Open**.

This property is updated to the number of columns of the value which is set in [DefaultScreenMode] of the configuration program by calling **ClaimDecive** immediately after **Open**.

In addition, this property is updated when the index value of **ScreenMode** is changed.

DeviceRows Property

Syntax **LONG DeviceRows;**

Remarks Holds the number of rows on this device.

This property is initialized by **Open**.

This property is updated to the number of rows of the value which is set in [DefaultScreenMode] of the configuration program by calling **ClaimDecive** immediately after **Open**.

In addition, this property is updated when the index value of **ScreenMode** is changed.

DeviceWindows Property

Syntax **LONG DeviceWindows;**

Remarks Holds the maximum window number supported by this device.
The following table shows the valid property values.

Value	Meaning
0	Only the device window is supported. No windows may be created.

This property is initialized to 0 by **Open**.

Rows Property

Syntax **LONG Rows;**

Remarks Holds the number of rows for this window.
This property is the same as **DeviceRows**.

This property is initialized to the same value of **DeviceRows** by **Open**.

ScreenMode Property R/W

Syntax **LONG ScreenMode;**

Remarks Contains the screen mode value of the device.
The values of this property are index values contained in **ScreenModeList**.
The following table shows an example of index values.

Index Value	Meaning
0	Default screen mode
1	2×20
2	5×20
3	2×40
4	5×40
5	8×40

This property can only be updated when the device is opened and claimed, but not enabled.

For this property, the default can be changed by setting of the configuration program.

This property is initialized to the value which is set in [DefaultScreenMode] of the configuration program by **Open**.

ScreenModeList Property

Syntax **BSTR ScreenModeList;**

Remarks Contains the comma-delimited list of row-column pairs that are supported by the device.

This property is initialized to "2×20, 5×20, 2×40, 5×40, 8×40" by **Open**.

5.5 Common Methods

CheckHealth Method

Syntax **LONG CheckHealth (LONG *Level*);**

The *Level* indicates the type of health check to be executed on the device. The following values may be specified:

Value	Meaning
OPOS_CH_INTERNAL(1)	Executes an internal test without changing the physical state of the device. Checks whether Display is ready to show.
OPOS_CH_EXTERNAL(2)	Executes a display test after confirming the communication with Display.
OPOS_CH_INTERACTIVE(3)	Executes an interactive test of the device. The Service Object displays the modal dialog and executes a display test.

Remarks Calls to test the state of a device. A text description of the results of this method is stored in **CheckHealthText**. **CheckHealth** is always synchronous.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	Indicates that the health checking procedure was initiated properly and, when possible to determine, indicates that the device is healthy. However, the health of many devices can only be determined by a visual inspection of the test results.
OPOS_E_ILLEGAL(106)	Unsupported <i>Level</i> is specified.
Other values	See the item of ResultCode .

ClaimDevice Method

Syntax **LONG ClaimDevice (LONG *Timeout*);**

The *Timeout* gives the maximum number of milliseconds to wait for exclusive access to be satisfied.

If the parameter is 0, the method returns the appropriate status immediately even though it cannot gain exclusive access to the device.

If OPOS_FOREVER(-1) is set, the method waits until exclusive access is satisfied.

Remarks Call this method to request exclusive access to the device. The Line Display cannot be used until exclusive access is obtained. When successful, **Claimed** is changed to TRUE.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	Exclusive access has been granted. Claimed is now TRUE. It is also returned if this application has already gained the exclusive access to the device.
OPOS_E_ILLEGAL(106)	An invalid <i>Timeout</i> is specified.
OPOS_E_TIMEOUT(112)	Another application has exclusive access to the device and the <i>Timeout</i> (in milliseconds) has elapsed before the device is released. Or, the device did not become a processable state before the <i>Timeout</i> (in milliseconds) has elapsed.

Close Method

Syntax **LONG** Close ();

Remarks Releases the device and its resources.
When **DeviceEnabled** is TRUE, the device is first disabled.
When **Claimed** is TRUE, exclusive access to the device is first released.
Do not execute this method while the event is in progress (or in the event handler).

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	Device has been disabled and closed.
Other values	See the item of ResultCode .

CompareFirmwareVersion Method

Syntax **LONG** CompareFirmwareVersion (**BSTR** *FirmwareFileName*, **Long** *result*);

Remarks This method is not supported.

Return The following value is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

DirectIO Method

Syntax **LONG** DirectIO (**LONG** *Command*, **LONG*** *pData*, **BSTR*** *pString*);

Parameter	Description
<i>Command</i>	Command number. Specific values assigned by the Service Object.
<i>pData</i>	Pointer to additional numeric data. Specific values vary by <i>Command</i> and Service Object.
<i>pString</i>	Pointer to additional string data. Specific values vary by <i>Command</i> and Service Object. The format of this data depends on the value of BinaryConversion . See BinaryConversion for details.

Remarks The following functions are supported.

- Binary data transmission
- Specified file sending

- **Binary data transmission**

Any Display command can be sent.

Parameter	Description
<i>Command</i>	DISP_DI_SET_BINARYDATA(601)
<i>pData</i>	Not used
<i>pString</i>	<p>IN Transmission data</p> <p>A string is specified in hexadecimal.</p> <p>Example: When US "LD" 31h 08h 00h 00h 00h "TXW" 31h 32h 33h 34h 35h is transmitted by the Display command "Input Text Data" "1F4C4431080000005458573132333435"</p> <p>See "DSP-A01 SERIES CUSTOMER DISPLAY TECHNICAL REFERENCE" for details of the Display command.</p>

- **Specified file sending**

A specified file can be sent.

Parameter	Description
<i>Command</i>	DISP_DI_SET_FILE(602)
<i>pData</i>	Not used
<i>pString</i>	<p>IN Sending file name</p> <p>A full path or relative path is specified as a character string.</p> <p>The file formats that can be specified are BIN, XML, JPEG and PNG.</p> <p>The file size that can be sent varies depending on the remaining memory capacity of the Display's user area and the file format.</p> <p>See "DSP-A01 SERIES CUSTOMER DISPLAY TECHNICAL REFERENCE" for details.</p> <p>Example: "C:\Temp\binary.bin"</p> <p style="padding-left: 40px;">"C:\Temp\Template.xml"</p> <p style="padding-left: 40px;">"C:\Temp\Image.jpg"</p>

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	DirectIO is successfully completed.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not performed. Execute ClaimDevice .
OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled to TRUE.
OPOS_E_ILLEGAL(106)	It is in one of the following states. <ul style="list-style-type: none">Parameter has an error.An unsupported file format is specified.An invalid command number is specified.
OPOS_E_NOHARDWARE(107)	When the device is used via a printer: It is in one of the following states. <ul style="list-style-type: none">The printer is not powered onThe printer and the host are not connected When the device is used alone: Display and the host are not connected.
OPOS_E_NOEXIST(109)	The specified file does not exist.
OPOS_E_FAILURE(111)	A communication error has occurred.

Open Method

Syntax **LONG Open (BSTR DeviceName);**

DeviceName specifies the device name to open. Specify the registered device name (such as "DSP-A01") or the logical device name of Display to execute this method.

Remarks Call this method to open the device.
When **Open** is successful, the common properties and other class-specific properties are initialized.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	Successful open.
OPOS_E_NOSERVICE(104)	Could not establish a connection to the corresponding Service Object.
OPOS_E_ILLEGAL(106)	The control is already open.
OPOS_E_NOEXIST(109)	The specified <i>DeviceName</i> is not found. This also includes when <i>DeviceName</i> is set as an empty string.
OPOS_E_FAILURE(111)	Initialization of the OPOS Driver is failed.

Note The value of **ResultCode** after calling **Open** may not be the same as **Open** return value for the following two cases.

When the OPOS Control is closed and **Open** fails:

- ResultCode** will continue to return OPOS_E_CLOSED(101).

When the OPOS Control is already open:

- Open** will return OPOS_E_ILLEGAL(106), but **ResultCode** may continue to return the value held before **Open**.

ReleaseDevice Method

Syntax **LONG ReleaseDevice ();**

Remarks Call this method to release exclusive access to the device.
If **DeviceEnabled** is TRUE and the device is an exclusive-use device, then the device is first disabled. Do not execute this method while the event is in progress (or in the event handler).

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	Exclusive access has been released. Claimed is now FALSE.
OPOS_E_ILLEGAL(106)	The application does not have exclusive access to the device.
Other values	See the item of ResultCode .

ResetStatistics Method

Syntax **LONG ResetStatistics (BSTR *StatisticsBuffer*);**

Remarks This method is not supported.

Return The following value is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

RetrieveStatistics Method

Syntax **LONG RetrieveStatistics (BSTR **pStatisticsBuffer*);**

Remarks This method is not supported.

Return The following value is returned by the method and also placed in **ResultCode**.

Parameter	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

UpdateFirmware Method

Syntax **LONG UpdateFirmware (BSTR *FirmwareFileName*);**

Remarks This method is not supported.

Return The following value is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

UpdateStatistics Method

Syntax **LONG UpdateStatistics (BSTR *StatisticsBuffer*);**

Remarks This method is not supported.

Return The following value is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

5.6 Specific Methods

ClearText Method

Syntax **LONG ClearText ();**

Remarks Clears the current window to blanks. **CursorRow** and **CursorColumn** is set to 0. Clears all bitmaps displayed in the window.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
Other values	See the item of ResultCode .

DisplayText Method

Syntax **LONG DisplayText (BSTR Data, LONG Attribute);**

Parameter	Meaning
<i>Data</i>	The string of characters to display. It consists of displayable characters and escape sequences, line feed (LF) and carriage return (CR). Refer to the values and meanings of special characters in <i>Data</i> below for LF, CR. The format of this data depends upon the value of the BinaryConversion . See BinaryConversion for details.
<i>Attribute</i>	Supports only DISP_DT_NORMAL.

The values and meanings of special characters within *Data* are as follows.

Value	Meaning
LF	Changes the next character's output position to the beginning of the next row. Scroll the window if the current row is the last row of the window.
CR	Changes the next character's output position to the beginning of the current row.

Remarks The characters in *Data* are processed beginning at the location specified by **CursorRow** and **CursorColumn**.

Character processing continues to the next row when the end of a window row is reached. If the end of the window is reached with additional characters to be processed, then the window is scrolled upward by 1 row.

If **CursorUpdate** is TRUE, **CursorRow** and **CursorColumn** are updated to point to the character position following the last character of *Data*.

Scrolling will not occur when the last character of *Data* is placed at the end of a row. In this case, If **CursorUpdate** is TRUE, **CursorRow** is set to the row containing the last character, and **CursorColumn** is set to **Columns** (that is, to one more than the final character of the row).

This stipulation ensures that Display does not scroll when a character is written into its last position. Instead, the Control Object will wait until another character is written before scrolling the window.

This method updates the window and view port immediately.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	<i>Attribute</i> is invalid.
Other values	See the item of ResultCode .

DisplayTextAt Method

Syntax **LONG DisplayTextAt (LONG Row, LONG Column, BSTR Data, LONG Attribute);**

Parameter	Meaning
<i>Row</i>	The start row for the text.
<i>Column</i>	The start column for the text.
<i>Data</i>	The string of characters to display. It consists of displayable characters and escape sequences, line feed (LF) and carriage return (CR). Refer to the values and meanings of special characters in <i>Data</i> below for LF, CR. The format of this data depends upon the value of the BinaryConversion . See BinaryConversion for details.
<i>Attribute</i>	Supports only DISP_DT_NORMAL.

The values and meanings of special characters within *Data* are as follows.

Value	Meaning
LF	Changes the next character's output position to the beginning of the next row. Scroll the window if the current row is the last row of the window.
CR	Changes the next character's output position to the beginning of the current row.

Remarks The characters in *Data* are processed beginning at the window location specified by *Row* and *Column*.

Character processing continues to the next row when the end of a window row is reached. If the end of the window is reached with additional characters to be processed, then the window is scrolled upward by 1 row.

If **CursorUpdate** is true, then **CursorRow** and **CursorColumn** are updated to point to the character position following the last character of *Data*.

Scrolling will not occur when the last character of *Data* is placed at the end of a row. In this case, If **CursorUpdate** is true, then **CursorRow** is set to the row containing the last character, and **CursorColumn** is set to **Columns** (that is, to one more than the final character of the row).

This stipulation ensures that Display does not scroll when a character is written into its last position. Instead, the Control Object will wait until another character is written before scrolling the window.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	Row or Column is out of range, or <i>Attribute</i> is invalid.
Other values	See the item of ResultCode .

Chapter 6 Registry Used by This Software

The registry set by the configuration program is described as follows.

Although the registry values can be set manually, set them using the configuration program. The contents of registry are read at the time when this software executes the **Open** method. Therefore, changing the value during operation of this software will not be reflected in the operation. In order to reflect new setting values, call the **Open** method after calling the **Close** method in this software.

6.1 Line Display Control

[Registry key]

HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\LineDisplay\DeviceNameKey

The device name key depends on Display. It is added by the configuration program for each Display device.

Display	Device Name key	Meaning
DSP-A01	"DSP-A01" and others	Device name of Display which is added by the configuration program.

[Registry value list]

Value Name	Value	Meaning
Default	OPOS.LINEDISPLAY.SO.SII.DSP.1	-
SODLLPath	C:\Program Files\SII\OPOS \LineDisplaySO.dll	File name of Line Display Service Object (Full path)
DefaultCharacterSet	932 (Japanese) 999 (English)	Default of CharacterSet
Description	SII DSP-A01 Line Display Service Object, Copyright (C) 20xx Seiko Instruments Inc.	Details of dedicated Line Display Service Object
DriverName	SII DSP-A01 SII RP-F10/G10	Supported driver name
LogRetentionPeriod	1	Retention period of log file 1: 1 day 3: 3 days 10: 10 days 30: 30 days 90: 90 days

Value Name	Value	Meaning
LogLevel	-1	Log output level of Line Display Service Object -1: No output 0: Error 4: Trace
DeviceBrightness	10	Brightness setting 0: 0% (blank) 1: 10% 2: 20% ... 10: 100%
DefaultScreenMode	1	Screen mode of the device 1: 2×20 2: 5×20 3: 2×40 4: 5×40 5: 8×40
Version	1.14.xx	Version of dedicated Line Display Service Object

Chapter 7 Header File

7.1 Line Display Header File

The header file used in this software is described below.
The constants defined uniquely in this software are as follows.

- **DISP_DI_SET_BINARYDATA**
- **DISP_DI_SET_FILE**

Header file: siidispcommon.h

```
////////////////////////////////////
//
// SIIDispCommon.h
//
// Line Display header file for OPOS Applications.
//
////////////////////////////////////
#ifndef SIIDISPCOMMON_H
#define SIIDISPCOMMON_H

////////////////////////////////////
// Parameter Constants of "DirectIO" Method
////////////////////////////////////
const LONG DISP_DI_SET_BINARYDATA = 601;
const LONG DISP_DI_SET_FILE      = 602;
#endif //!defined(SIIDISPCOMMON_H)
```