Ifm efector Corporation
IO-Link providers

Version 1.0.0

User's Guide

November 30, 2020

NOTE:
This document is automatically translated.
The figures in the manual are based on Japanese.

[Revision History]

Version	Date	Description
1.0.0	2020-02-29	First edition
	2020-11-30	Described the explanation about the IODD file.

[Verifying Model]

Model	Version	POINTS OF CAUTION
AL1342	AL1x4x_cn_mo_v2.3.20	

Contents

1. Introduction

This document is the user's guide of the provider that connects to IO-Link Master (AL1342) of ifm efector Corporation, obtains the information of IO-Link Master, and inputs and outputs the data of IO-Link devices connected to the Master. Fig. 1-1 shows the overall configuration of one provider and device.

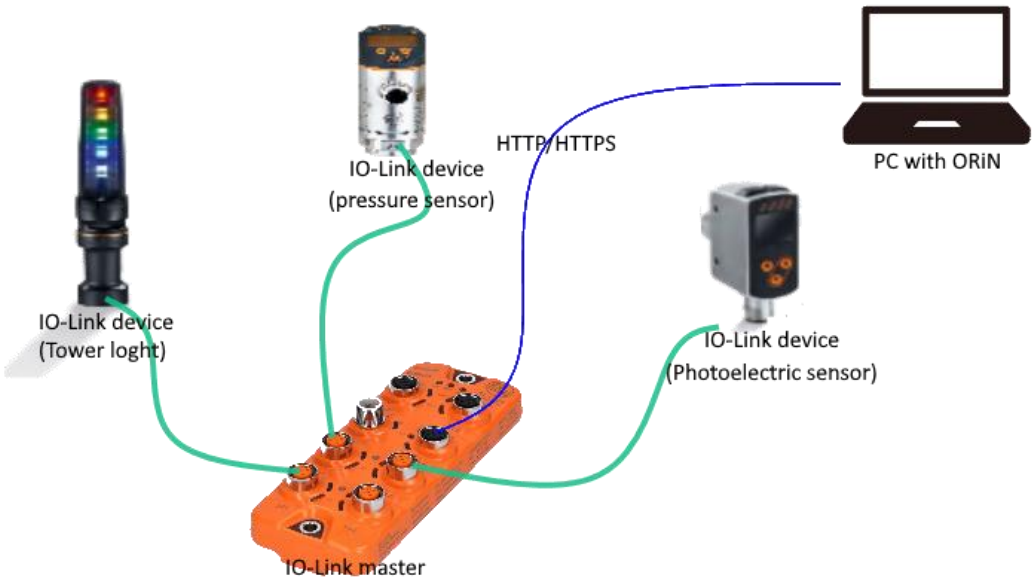


Fig. 1-1 Configuration Diagram

IO-Link masters and providers communicate by HTTP/HTTPS communication. Providers use IO-Link masters to exchange data even when retrieving and controlling IO-Link devices. This allows connections to be made without considering the manufacturer of IO-Link devices. Note that this only applies to IO-Link devices that support IODD file version 1.1.

1.1. Environment and version assumed by this document

It is assumed that the client PC runs on Windows and the target CNC is a company-named device type capable of Ethernet connection. The development environment of the PC can be developed if the programming environment supports Component Object Model (COM, Component Object Model).

1.2. Informative source

All of the programming examples in this book are given in Visual Basic 6. 0, but they can be developed in a variety of programming languages, including C++, Java, and.NET. For usage information, see ORiN2 Programming Guide.

ORiN 2 Programming Guide applies to the following files in ORiN2 SDK installation folder:

- ORiN2¥CAO¥Doc¥ORiN2_ProgrammersGuide_<lang>.pdf

※ Replace <lang> with the language-specific language-string for your environment.

Describes basic ORiN2, COM/DCOM knowledge and technologies required to develop provider-based applications, including examples.

IODD files required to connect to IO-Link devices can be downloaded from the website below.

https://ioddfinder.io-link.comhttps://ioddfinder.io-link.com/

2. Setting Up Your Environment for Application Development

2.1. Connecting IO-Link masters to providers

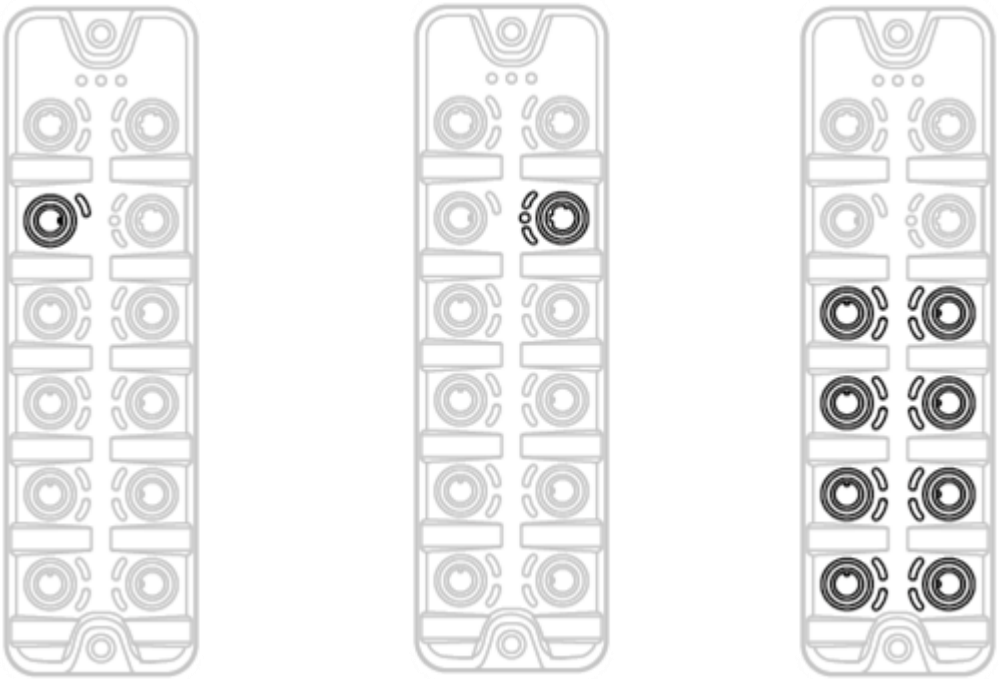


Fig. 2-1 shows the power, IoT, and IO-Link ports from left to right.

2.1.1. Connecting the Power Supply Ports of IO-Link Masters

Connect the power connector cable to the power supply port.

2.1.2. Connecting IO-Link Masters to Communication Ports

Connect the communication connector cable to IoT port, and connect the terminal on the other side to an empty port of the switching hub connected to the client PC or directly to the Lan port of the client PC.

2.1.3. Connecting IO-Link Masters to IO-Link Devices

Specify IO-Link cables to which IO-Link devices you want to connect to IO-Link ports are connected.

2.2. Setting up the PC development environment

2.2.1. Automated provider installation

If you have ORiN2 SDK installed on your computer, you are ready to connect to the operating environment (line time).

Please prepare a programming environment that supports Component Object Model (COM, Component Object Model) such as Microsoft Visual Studio 6. 0, 2003/2005/2010/2010, LabVIEW separately.

2.2.2. Installing a Provider Manually

Registry registration is required to install the provider manually.

Registry registration

To register the registry, start the command prompt with administrator privileges and execute regsvr32 command.

Also, before the CAO engine can run, you must have a valid ORiN2 SDK license for each PC. See the section "Adding and Removing Licenses" in ORiN2 SDK User's Guide.

Table 2-1 Provider information

File name	CaoProvifmIO-Link.dll
ProgID	CaoProv.ifm.IO-Link
Registry registration	Regsvr32 CaoProvifmIO-Link.dll
Deletion of Registry Registration	Regsvr32 /u CaoProvifmIO-Link.dll

3. Command Reference

3.1. List of Methods/Properties

Table 3-1 List of Methods and Properties

Category	Methods/Properties[1]		Function	Reference
CaoWorkspace				
	AddController	M	Configuring Connections to IO-Link Masters	P.9
CaoController				
	AddExtension	M	Configuring IO-Link Devices	P.11
	Extensions	P	Obtaining the extension board collection held by the controller	P.12
	VariableNames	P	Get a list of connectable variable names	P.13
	AddVariable	M	Adding Variable Objects	P.13
	Variables	P	Retrieving a Collection of Variables Held by a Controller	P.13
CaoExtenstion				
	AddVariable	M	Adding Variable Objects	P.14
	VariableNames	P	Get a list of connectable variable names	P.14
	Variables	P	Retrieving a Collection of Variables Held by a Task	P.14
CaoVariable				
	Value	P	Get/Set Value	P.15

3.2. Method Properties

3.2.1. CaoWorkspace classes

3.2.1.1. AddController method

Add controller objects to CaoWorkspace. The provider connects to the corresponding IO-Link Master using the access point information and communication method specified here. The following are the specifications for AddController method:

Providers do not verify connectivity with IO-Link masters during AddController. Therefore, even if AddController is successful, the connection with IO-Link master is not guaranteed. Note that the connection and communication with IO-Link master are actually made and communicated when Value properties of CaoController variables are accessed.

Format

CaoController AddController

(

[1] M: Methods, P: Properties, and E: Events, respectively.

```
        "<controller name>",            // Controller name (optional)
        "CaoProv.ifm.IO-Link",          // Provider name (fixed)
        "<machine name>",               // Provider execution machine name (not used)
        "<options>"                     // Optional option string
)
```

OPTIONS

The following options are specified for option strings: The option string is a string consisting of the following options separated by a comma (,). The underlined item in the value range column in the table indicates the default value when the option is omitted.

OPTIONS	Required	Description	Value Range
Server	○	Specify the destination of IO-Link master to be connected.	---
SSL	-	Specify whether to perform SSL communication with a 2 value.	False: HTTP connections True:HTTPS connections
User	-	Specify the username for connecting to HTTPS. This option is required only if you specify True for SSL. If you specify False for SSL, this option is ignored. When this option is omitted, "administrator" is used.	Any character string
Password	-	Specify the passwords for connecting to HTTPS. This option is required only if you specify True for SSL. If you specify False for SSL, this option is ignored. When this option is omitted, "password" is used.	Any character string
Timeout	-	Specify the connection timeout (ms).	1 - Default: 500ms

Examples of use

The following examples assume that the following code exists:

```
Dim engine As CaoEngine     ' Engine objects
Dim workspace As CaoWorkspace       ' Workspace objects
Dim controller As CaoController         ' Controller objects

Set engine = New CaoEngine
Set workspace = engine.Workspaces.Item(0)

' Connection method
Private Sub Connect()
        ' for HTTP connections
        Set controller = workspace.AddController("Master1", _
                                                  "CaoProv.ifm.IO-Link", _
                                                  "", _
                                                  "Server=192.168.0.1")

        ' for HTTPS connections
        Set controller = workspace.AddController("Master1", _
                                                  "CaoProv.ifm.IO-Link", _
```

```
                                        """, _
                                        "Server=192.168.0.1, SSL=True, User=usr, Password=pass")
End Sub

' Disconnect method
Private Sub Diconnect()
    ' remove CaoController from CaoWorkspace
    Call workspace.Controllers.Remove(controller.Index)
    ' Clear CaoController
    Set controller = Nothing

    ' remove CaoWorkspace from CaoEngine
    Call engine.Workspaces.Remove(workspace.Index)
    ' Clear CaoWorkspace
    Set workspace = Nothing

    ' Clear CaoEngine
    Set engine = Nothing
End Sub
```

3.2.1.2. Connection in Security Mode On

If the security mode of IO-Link master to be connected is on, specify True in the SSL option, and specify the user ID and password in UserId and Password options respectively.

When the security mode of IO-Link Master is switched from Off to On, user IDs and passwords must be set first using ifm electronic's LR DEVICE application.

3.2.2. CaoController classes

3.2.2.1. AddExtension method

Set the providers to IO-Link devices that connect to (or are connected to) IO-Link masters. The following are the specifications for AddExtension method:

During AddExtension, providers do not verify connectivity with devices and the existence of port numbers. Therefore, successful AddExtension does not guarantee connectivity with IO-Link devices. Actual connections and communications occur when accessing Value properties of CaoExtension variables.

Format

CaoExtension AddExtension

```
(
        "<extension name>",         // Extension name (optional)
        "<options>"                 // Option string
)
```

OPTIONS

The following options are specified for option strings: The option string is a string consisting of the following options separated by a comma (,).

OPTIONS	Required	Description
Port	○	Specify the port number to which IO-Link devices are connected. Specify a port number that exists on IO-Link master.
IODD	-	Specify IODD file corresponding to IO-Link devices. See page 12 for specifying IODD file.12 This option can be omitted. If this option is omitted, the process data and IO-Link device setting information in 3.3. 2 CaoExtension Class Variables cannot be acquired.3.3.2CaoExtension class-variable

Examples of use

```
'Connection
Call Connect

' Add IO-Link devices
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
                                        "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")

Call controller.Extensions.Remove(extension.Index)
Set caoExt = Nothing

'Disconnect
Call Disconnect
```

3.2.2.1.1. How to specify IODD files

IODD files can be specified as relative or absolute paths. See Appendix D for the IODD file.

- Relative path specification

  When a relative path is specified, IODD file is read based on the following directory.

  Reference path

  "Provider Installation Directory"/XML

- To specify an absolute path

  Specify the absolute paths of IODD files to be read.

3.2.2.2. Extensions Properties

Get the collections of CaoExtension the controllers hold.

Examples of use

```
'Connection
Call Connect
```

```
'Obtain Extended Board Collection
Dim extensions as CaoExtensions
Set extensions = controller.Extensions

'Obtain Expansion Board
Dim extension as CaoExtension
Set extension = extensions.Item(0)

'Disconnect
Call Disconnect
```

3.2.2.3. VariableNames Properties

Get a list of variable names that can be connected. The names of the retrieved variables can be used as the first arguments to AddVariable method.AddVariable method

Examples of use

```
'Connection
Call Connect

' Get file name list
Dim variables as Variant
Variables = controller.VariableNames

'Disconnect
Call Disconnect
```

3.2.2.4. AddVariable method

Adds variable objects to CaoController. Only those variables listed in "3.3.1 CaoController Variables" can be used.

AddVariable specifications are shown below.

Format

```
CaoVariable AddVariable
(
        "<variable name>",          // Variable name
        "<options>"                 // Optional option string
)
```

3.2.2.5. Variables Properties

Retrieves the variable collection held by the controller.

Examples of use

```
'Connection
Call Connect

' Get Variable Collection
```

```
Dim variables as CaoVariables
Set variables = controller.Variables

' Get Variable
Dim variable as CaoVariable
Set variable = variables.Item(0)

'Disconnect
Call Disconnect
```

3.2.3. CaoExtension classes

3.2.3.1. AddVariable method

Adds variable objects to CaoController. Only the variable names shown in 3.3.2 can be used.

AddVariable specifications are shown below.

Format

CaoVariable AddVariable

```
(
        "<variable name>",          // Variable name
        "<options>"                 // Optional option string
 )
```

3.2.3.2. VariableNames Properties

Get a list of variable names that can be connected. The names of the retrieved variables can be used as the first arguments to AddVariable method.AddVariable method

Examples of use

```
'Connection
Call Connect

' Adding an expansion board
Dim caoExt As CaoExtension
Set caoExt = controller.AddExtension("Snsor1", _
                                     "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")

' Get Variable Name List
Dim variables as Variant
Variables = caoExt.VariableNames
```

3.2.3.3. Variables Properties

Get the variable collection held by the expansion board.

Examples of use

```
'Connection
Call Connect
```

```
' Adding an expansion board
Dim caoExt As CaoExtension
Set caoExt = controller.AddExtension("Snsor1", _
                                     "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")

' Get Variable Collection
Dim variables as CaoVariables
Set variables = caoExt.Variables

' Get Variable
Dim variable as CaoVariable
Set variable = variables.Item(0)
```

3.2.4. CaoVariable classes

3.2.4.1. Value Properties

Retrieves and sets data from the port of the connected IO-Link master or IO-Link devices connected to that port. The behavior depends on the variable name. For more information, see 3.3.

3.3. List of Variables

Defines a list of variables that can be used in each class. Variables refer to objects of CaoVariable classes.

3.3.1. CaoController variables

List of 3-2 CaoController Variables

Variable name	Description	Reference
IO-Link Master Basic Info		
@ProductCode	Obtain the part number.	P.16
@Vendor	Get the manufacturer.	P.16
@DeviceFamily	Gets the device family.	P.17
@HWRevision	Get the hardware version.	P.17
@SerialNumber	Get the serial number.	P.17
@SWRevision	Get the firmware version.	P.18
@BootLoaderRevision	Get the bootloader version.	P.18
@ExtensionRevisions	Get the firmware and bootloader version.	P.18
@FieldbusType	Gets the fieldbus type.	P.19
@IOLinkMasterName	Gets/sets IO-Link master name.	P.19
IO-Link master control data		
@IOLinkMasterVersion	Get the software version of IO-Link Master.	P.20
@IOLinkMasterSWType	Get IO-Link master software type.	P.20

@ContainerMaxSize	Get the max size (byte) of a container area.	P.20
@ContainerChunkSize	Get the size (byte) of the data segment.	P.21
@ContainerSize	Get the size (in byte) of the container area.	P.21
Diagnostic data		
@Temperature	Gets IO-Link master temperature in Celsius.	P.22
@Voltage	Gets the voltage (V) of IO-Link master.	P.22
@Current	Gets the current (A) of IO-Link master.	P.22
@SuperVisionStatus	Retrieve device-supplied diagnostic information.	P.23
Provider information		
@Version	Get the version of the provider.	P.23

3.3.1.1. IO-Link Master Basic Info

3.3.1.1.1. @ProductCode

Gets the part number as a string.

Data Type

Type Description	
VT_BSTR	Part Number

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@ProductCode")
' Value retrieval
Dim strVal As String
StrVal = var.Value
```

3.3.1.1.2. @Vendor

Get the manufacturer as a string.

Data Type

Type Description	
VT_BSTR	Manufacturer

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@Vendor")
```

```
' Value retrieval
Dim strVal As String
StrVal = var.Value
```

3.3.1.1.3. @DeviceFamily

Gets the device family as a string.

Data Type

Type Description	
VT_BSTR	Device family

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@DeviceFamily")
' Value retrieval
Dim strVal As String
StrVal = var.Value
```

3.3.1.1.4. @HWRevision

Gets the hardware version as a string.

Data Type

Type Description	
VT_BSTR	Hardware Version

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@HWRevision")
' Value retrieval
Dim strVal As String
StrVal = var.Value
```

3.3.1.1.5. @SerialNnumber

Get the serial number as a string.

Data Type

Type Description	
VT_BSTR	Device family

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@SerialNnumber")
' Value retrieval
Dim strVal As String
StrVal = var.Value
```

3.3.1.1.6. @SWRevision

Get the firmware version as a string.

Data Type

Type Description	
VT_BSTR	Firmware Version

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@SWRevision")
' Value retrieval
Dim strVal As String
StrVal = var.Value
```

3.3.1.1.7. @BootLoaderRevision

Get the bootloader version as a string.

Data Type

Type Description	
VT_BSTR	Boot loader version

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@BootLoaderRevision")
' Value retrieval
Dim strVal As String
StrVal = var.Value
```

3.3.1.1.8. @ExtensionRevisions

Get the firmware and bootloader version as a string.

Data Type

Type Description	
VT_BSTR	Firmware and Bootloader Version

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@ExtensionRevisions")
' Value retrieval
Dim strVal As String
StrVal = var.Value
```

3.3.1.1.9. @FieldbusType

Gets the fieldbus type as a string.

Data Type

Type Description	
VT_BSTR	Fieldbus type

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@FieldbusType")
' Value retrieval
Dim strVal As String
StrVal = var.Value
```

3.3.1.1.10. @IOLinkMasterName

Gets/sets IO-Link master name.

Data Type

Type Description	
VT_BSTR	IO-Link Master Names

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@IOLinkMasterName")
' Value retrieval
Dim strVal As String
```

```
StrVal = var.Value
```

3.3.1.2. IO-Link master control data

3.3.1.2.1. @IOLinkMasterVersion

Gets IO-Link master software version as a string.

Data Type

Type Description	
VT_BSTR	Software versions of IO-Link masters

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@IOLinkMasterVersion")
' Value retrieval
Dim strVal As String
StrVal = var.Value
```

3.3.1.2.2. @IOLinkMasterSWType

Gets IO-Link master software type as a string.

Data Type

Type Description	
VT_BSTR	IO-Link Master Software Types

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@IOLinkMasterSWType")
' Value retrieval
Dim strVal As String
StrVal = var.Value
```

3.3.1.2.3. @ContainerMaxSize

Get the max size (byte) of a container area.

Data Type

Type Description	
VT_I4	Maximum size of the container area

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@ContainerMaxSize")
' Value retrieval
Dim val As Long
val = var.Value
```

3.3.1.2.4. @ContainerChunkSize

Get the size (byte) of the data segment.

Data Type

Type Description	
VT_I4	Size of the data segment (byte)

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@ContainerMaxSize")
' Value retrieval
Dim val As LOng
val = var.Value
```

3.3.1.2.5. @ContainerSize

Get the size (in byte) of the container area.

Data Type

Type Description	
VT_I4	Size of the container area

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@ContainerSize")
' Value retrieval
Dim val As Long
val = var.Value
```

3.3.1.3. Diagnostic data

3.3.1.3.1. @Templerature

Obtain the temperatures (°C) of IO-Link masters.

Data Type

Type Description	
VT_I4	Temperatures of IO-Link masters (°C)

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@Templerature")
' Value retrieval
Dim val As Long
val = var.Value
```

3.3.1.3.2. @Voltage

Gets the voltage (V) of IO-Link master.

Data Type

Type Description	
VT_I4	IO-Link master voltages (V)

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@Voltage")
' Value retrieval
Dim val As Long
val = var.Value
```

3.3.1.3.3. @Current

Gets the current (A) of IO-Link master.

Data Type

Type Description	
VT_I4	Current of IO-Link masters (A)

Examples of use

```
'Connection
```

```
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@Current")
' Value retrieval
Dim val As Long
val = var.Value
```

3.3.1.3.4. @SuperVisionStatus

Retrieve device-supplied diagnostic information.

Data Type

Type	Description
VT_I4	Device-supplied diagnostic information

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@SuperVisionStatus")
' Value retrieval
Dim val As Long
val = var.Value
```

3.3.1.4. Provider information

3.3.1.4.1. @VERSION

Get the version of the DLL.

Data Type

Type	Description
VT_BSTR	Get the version of the DLL. *.*.*

Examples of use

```
'Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@VERSION")
' Value retrieval
Dim value As String
Value = var.Value
```

3.3.2. CaoExtension class-variable

List of 3-3 CaoExtension Class Variables

Variable name	Description	Reference
Configuring IO-Link Ports		
@SendProcDataPort	Gets and sets the port number used for sending process data to LR_SMARTOBSERVER.	P.25
@TransCycleTimePreset	Gets and sets the number of microseconds for transferring data on IO-Link ports.	P.25
@TransCycleTimeAct	Gets the current cyclic time for transferring data between IO-Link masters and IO-Link devices on a port.	P.26
@Mode	Gets or sets the operation mode of IO-Link ports.	P.26
@ComSpeed	Gets the datatransfer rate of IO-Link ports.	P.27
@ValidationDataStorage	Gets and sets how new IO-Link devices are validated when they are connected.	P.27
@ValidationVendorId	Gets/sets VendorID to be verified.	P.28
@ValidationDeviceId	Gets/sets DeviceID to be verified.	P.29
@DataStorageMaxSize	Get the max size (byte) of the data storage area.	P.29
@DataStorageChunkSize	Get the size (byte) of the data segment.	P.29
@DataStorageSize	Get the size of the data storage area (byte).	P.29
Process data		
@Pin2Input	Gets the Pin 2 values of IO-Link ports.	P.29
@ProcessDataIn	Obtain IO-Link input value (input process data) of pin 4 of IO-Link port.	P.30
@ProcessDataOut	Gets or sets IO-Link output value (output process data) of pin 4 of IO-Link port.	P.33
IO-Link Devices Settings		
<IODD FileVariables>	Obtain/configure IO-Link device configuration information. You can specify the variables that are defined in SpecialistRoleMenuSet of IODD file. Variable names can be obtained using 3.2.3.2.	P.37
IO-Link Devices		
@Status	Get status of IO-Link devices.	P.42
@VendorId	Get the manufacturer id of IO-Link devices.	P.42
@DeviceId	Get the device id of IO-Link device.	P.43
@ProductName	Obtain the product name of IO-Link device.	P.43
@Serial	Get the serial number of IO-Link devices.	P.44

@ApplicationTag	Gets/sets the application tag.	P.44

3.3.2.1. Configuring IO-Link Ports

3.3.2.1.1. @SendProcDataPort

Gets/sets the port number used to send process data to LR_SMARTOBSERVER. Parameters can only be edited if connection to Modbus TCP plc is interrupted. You can only set values on ports that do not have IO-Link devices attached.

Data Type

Type Description	
VT_I4	Port number to use when sending process data to LR_SMARTOBSERVER Value range: 0-65525

Examples of use

```
'Connection
Call Connect

Addition of the 'Extension
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
                                        "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")

'Add Variable
Dim var As CaoVariable
Set var = extension.AddVariable("@SendProcDataPort")

' Value retrieval
Dim value As Long
value = var.Value

' Value Setting
var.Value = 63234
```

3.3.2.1.2. @TransCycleTimePreset

Gets and sets the number of microseconds for transferring data on IO-Link ports. You can only set values on ports that do not have IO-Link devices attached.

Data Type

Type Description	
VT_I4	Cycle time for transferring data on IO-Link ports in microseconds Value range: 0-132800, where 0 is the fastest cycle time

Examples of use

```
'Connection
```

```
Call Connect

Addition of the 'Extension
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
                                        "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")

'Add Variable
Dim var As CaoVariable
Set var = extension.AddVariable("@TransCycleTimePreset")

' Get/Set Value
Dim value As Long
value = var.Value
If (value <> 0) Then
    var.Value = 0
EndIf
```

3.3.2.1.3. @TransCycleTimeAct

Gets the current cyclic time (in microseconds) for transferring data between IO-Link masters and IO-Link devices.

Data Type

Type Description	
VT_I4	Current cyclic time (in microseconds) for transferring data between IO-Link masters and IO-Link devices Value range: 1-132800

Examples of use

```
'Connection
Call Connect

Addition of the 'Extension
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
                                        "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")

'Add Variable
Dim var As CaoVariable
Set var = extension.AddVariable("@TransCycleTimeAct")

' Get/Set Value
Dim value As Long
value = var.Value
```

3.3.2.1.4. @Mode

Gets and sets the operational mode of IO-Link ports. Parameters can only be edited if connectivity to Modbus TCP plc is interrupted.

Data Type

Type Description	
VT_I4	Operating mode. 0: deactivated 1: Digital input (DI) 2: Digital output (DO) 3: IO-Link

3.3.2.1.5. @ComSpeed

Gets the datatransfer rate of IO-Link ports.

Data Type

Type Description	
VT_I4	Rate at which IO-Link ports are transmitted

3.3.2.1.6. @ValidationDataStorage

Gets/sets how new IO-Link devices are verified when they are connected. Parameters can only be edited if connectivity to Modbus TCP plc is interrupted. Following variable values

Data Type

Type Description	
VT_I4	How to verify. For details, refer to the attached table.

@ValidationDataStorage Values	Verifying IO-Link Devices	Saving Parameter Values	Recovering Parameter Values
0	Do not validate.	-	-
1	IODD Ver1. Verify that it is compliant with 0.	-	-
2	IODD Ver1. 1 Compliance is verified. It also verifies IO-Link device with the vendor ID and device ID specified by @ValidationVendorId,@ValidationDeviceId.	-	-
3	IODD Ver1. 1 Compliance is verified. It also verifies IO-Link device with the vendor ID and device ID specified by	Parameter values are saved	If identical IO-Link devices are connected,

	@ValidationVendorId,@ValidationDeviceId.	automatically. The changes to the current parameter values are saved.	they will be recovered with the factory-default settings.
4	IODD Ver1. 1 Compliance is verified. It also verifies IO-Link device with the vendor ID and device ID specified by @ValidationVendorId,@ValidationDeviceId.	Parameter values are not saved automatically.	If identical IO-Link devices are connected, they will be recovered with the factory-default settings.

Examples of use

Describes how to set the port-1 validation method to IODD1. 1 validation mode and restrict connections other than Vendor: ifm electronic gmbh, Device: OGD596.

```
'Connection
Call Connect

Addition of the 'Extension
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", "Port=1")

'Add Variable
Dim validationDataStorage As CaoVariable
Set validationDataStorage = extension.AddVariable("@ValidationDataStorage")
Dim varidationVendorId As CaoVariable
Set varidationVendorId = extension.AddVariable("@ValidationVendorId")
Dim varidationDeviceId As CaoVariable
Set varidationDeviceId = extension.AddVariable("@ValidationDeviceId")

'SETTING
ValidationDataStorage.Value = 2 'Compliant with Ver1. 1
ValidationVendorId.Value = 310 ' ifm electronic gmbh
ValidationDeviceId.Value = 1066 ' OGD596
```

3.3.2.1.7. @ValidationVendorId

Gets/sets the vendor ID to be verified when @ValidationDataStorage is 2 or greater. When @ValidationDataStorage is 2 or more, if the vendor ID of the connected IO-Link device differs from the vendor ID of this variable, IO-Link master and IO-Link device cannot be connected. Parameter can be edited only when

the connection to Modbus TCP plc is interrupted.

Data Type

Type Description	
VT_I4	Vendor ID to verify

3.3.2.1.8. @VaridationDeviceId

Gets/sets DeviceID to validate when @ValidationDataStorage is greater than or equal to 2. When @ValidationDataStorage is 2 or more, if the device ID of the connected IO-Link device differs from the value of this variable, IO-Link master and IO-Link device cannot be connected. Parameter can be edited only when the connection to Modbus TCP plc is interrupted.

Data Type

Type Description	
VT_I4	Device ID to verify

3.3.2.1.9. @DataStorageMaxSize

Get the max size (byte) of the data storage area.

Data Type

Type Description	
VT_I4	Max Data Storage Area Size (byte)

3.3.2.1.10. @DataStorageChunkSize

Get the size (byte) of the data segment.

Data Type

Type Description	
VT_I4	Size of the data segment (byte)

3.3.2.1.11. @DataStorageSize

Get the size of the data storage area in byte.

Data Type

Type Description	
VT_I4	Size of the data storage area (byte)

3.3.2.2. Process data

3.3.2.2.1. @Pin2Input

Acquires the digital-in value of Pin 2 of IO-Link port.

Data Type

Type	Description
VT_I4	Pin 2 digital-in values of IO-Link ports

Examples of use

```
'Connection
Call Connect

Addition of the 'Extension
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
                                        "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")

'Add Variable
Dim var As CaoVariable
Set var = extension.AddVariable("@Pin2Input")

' Value retrieval
Dim value As Long
value = var.Value
```

3.3.2.2.2. @ProcessDataIn

Obtain IO-Link input value (input process data) of pin 4 of IO-Link port. The datatype depends on IODD files specified in CaoController::AddExtension.

Data Type

Type	Description
VT_VARIANT	The @ProcessDataIn datatype depends on the specified IODD files. For the types of the specified IODD files "IODevice.ProfileBody.ProcessDataCollection.ProcessDataIn" See.

Sample IODD Files

The following figure defines the input process data in IODD file "ifm-00042A-20180717-IODD1. 1.xml".

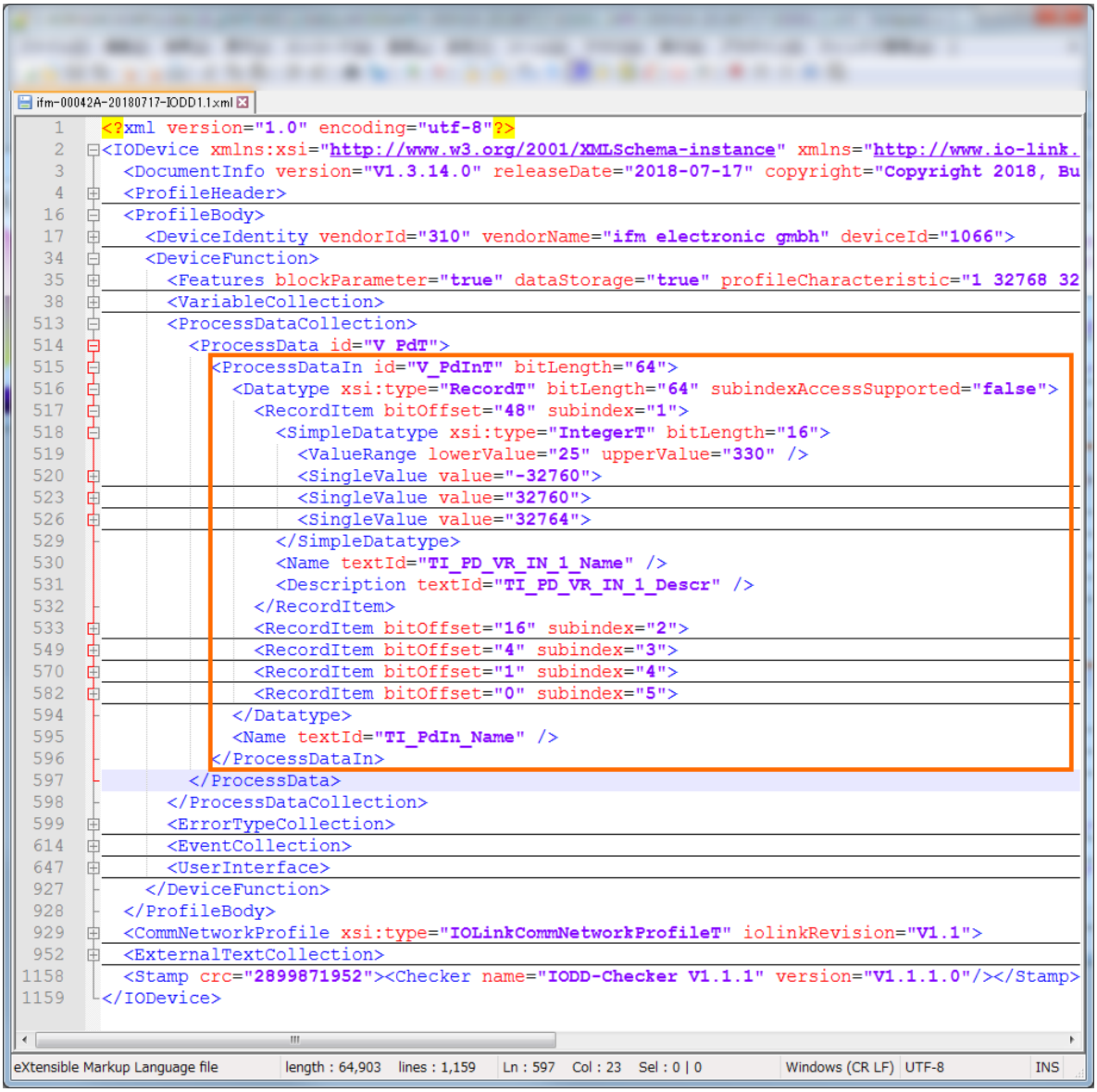


Fig. 3-1 Actual ProcessDataIn definitions

You can see that ProcessDataIn is of type RecordT and there are five records.

Data Type Examples

Examples of data types when ifm-00042A-20180717-IODD1. 1.xml is specified are shown below.

Type	Description

Type Description		
VT_VARIANT VT_ARRRAY		In the examples, because the process data type is RecordT, the data type corresponding to RecordItem type corresponding to each subindex is held. Note that the data is sorted in ascending subindex order, not in the order defined in IODD files. The data names are strings defined in the "Name" of each "RecordItem".
0	VT_I2	Name: Distance
1	VT_I2	Name: Reflectivity
2	VT_UI1	Name: Device Status
3	VT_BOOL	Data name: OUT2
4	VT_BOOL	Data name: OUT1

Examples of use

The following example acquires the input process data from the ifm optical sensor "OGD596" connected to Port 1.

```
'Connection
Call Connect

Addition of the 'Extension
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
                                        "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")

'Add Variable
Dim var As CaoVariable
Set var = extension.AddVariable("@ProcessDataIn")

'Get Value
Dim value As Variant
value = var.Value

' Get Distance
Dim distance As Long
distance = value(0)

' Get Reflectivity
Dim reflectivity As Long
reflectivity = value(1)

' Get DeviceStatus
Dim deviceStatus As Byte
deviceStatus = value(2)

' Get OUT2
Dim out2 As Boolean
out2 = value(3)
```

```
' Get OUT1
Dim out1 As Boolean
out1 = value(4)
```

3.3.2.2.3. @ProcessDataOut

Gets or sets IO-Link output value (output process data) of pin 4 of IO-Link port. The datatype depends on IODD files specified in CaoController::AddExtension.

Data Type

Type Description	
VT_VARIANT	The @ProcessDataOut datatype depends on the specified IODD files. For the types of the specified IODD files "IODevice.ProfileBody.ProcessDataCollection.ProcessDataOut" See.

Sample IODD Files

The following figure defines the output process data in IODD file "Balluff-BNI_IOL-802-102-Z036-20150730-IODD1. 1.xml".

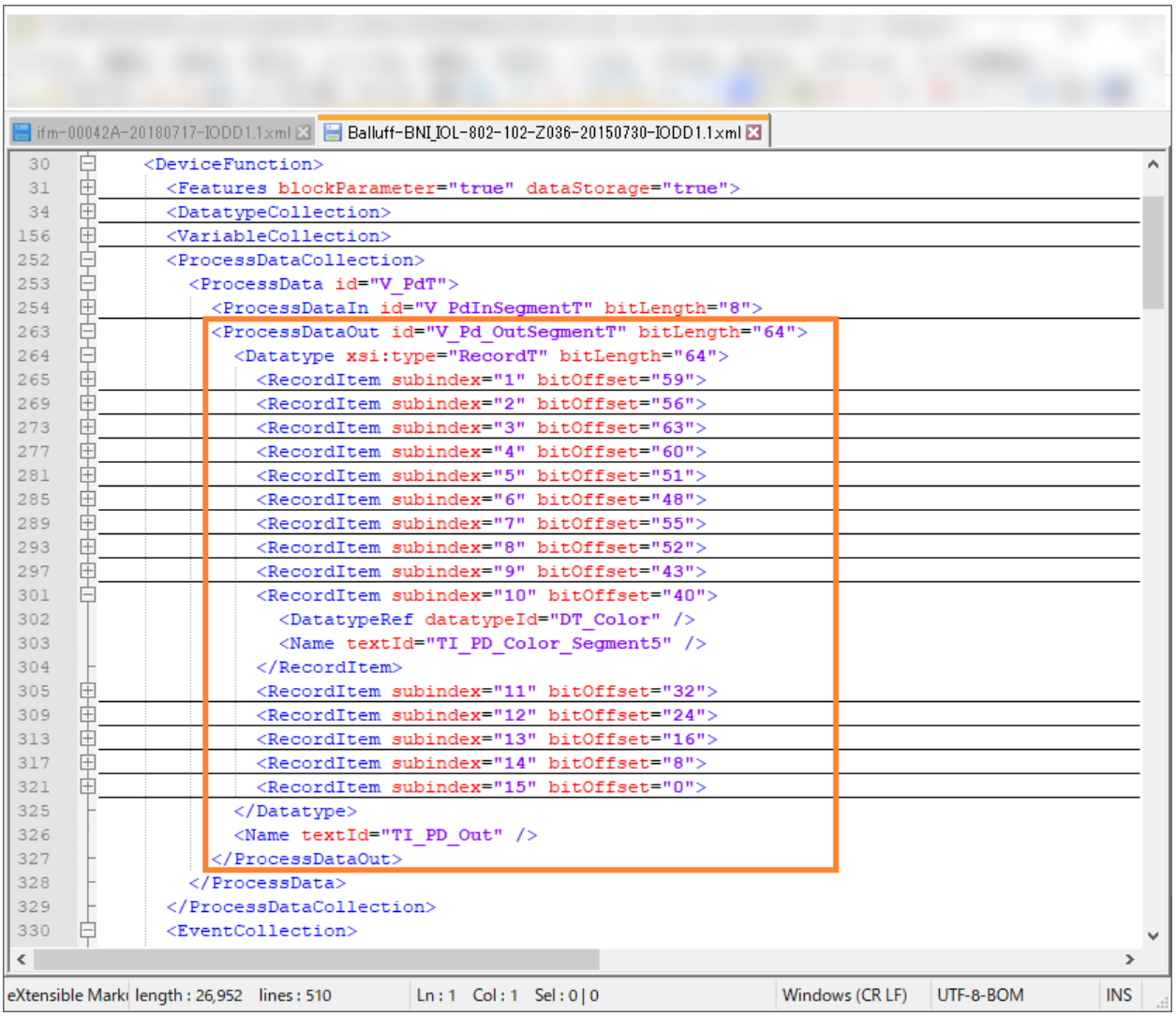


Fig. 3-2 Actual ProcessDataOut definitions

You can see that ProcessDataOut is of type RecordT and there are 15 records.
The data type of each record refers to DataTypeCollection element.

Data Type Examples

Balluff-BNI_IOL-802-102-Example data type when Z036-20150730-IODD1. 1.xml is specified

Type Description	
VT_VARIANT VT_ARRRAY	In the examples, because the process data type is RecordT, the data type corresponding to RecordItem type corresponding to each subindex is held. Note that the data is sorted in ascending subindex order, not in the order defined in IODD files. The data names are strings defined in the "Name" of each "RecordItem".

Type Description		
0	VT_BOOL	Name: Segment 1 blink/dominant FALSE: No Blinking TRUE: Blinking
1	VT_UI1	Name: Segment 1 color/Background color 0: OFF 1: Green 2: Red 3: Yellow 4: Blue 5: Orange 6: User-defined 7: White
2	VT_BOOL	Name: Segment 2 blink/dominant FALSE: No Blinking TRUE: Blinking
3	VT_UI1	Name: Segment 2 color/Running color 0: OFF 1: Green 2: Red 3: Yellow 4: Blue 5: Orange 6: User-defined 7: White
4	VT_BOOL	Name: Segment 3 blink/dominant FALSE: No Blinking TRUE: Blinking

Type Description		
5	VT_UI1	Name: Segment 3 color/Running color 0: OFF 1: Green 2: Red 3: Yellow 4: Blue 5: Orange 6: User-defined 7: White
6	VT_BOOL	Name: Segment 4 blink/dominant FALSE: No Blinking TRUE: Blinking
7	VT_UI1	Name: Segment 4 color 0: OFF 1: Green 2: Red 3: Yellow 4: Blue 5: Orange 6: User-defined 7: White
8	VT_BOOL	Name: Segment 5 blink/dominant FALSE: No Blinking TRUE: Blinking
9	VT_UI1	Name: Segment 5 color 0: OFF 1: Green 2: Red 3: Yellow 4: Blue 5: Orange 6: User-defined 7: White

Type	Description	
10	VT_UI1	Name: Operating mode 0: Segment Mode 1: Level mode 2: Runlight mode
11	VT_UI1	Name: Number of segments/Level type
12	VT_UI1	Name: Blinking mode/Level value low byte
13	VT_UI1	Name: Blinking frequency/Level value high byte/Running speed
14	VT_UI1	Name: Reserved

Examples of use

Describes the example of lighting BALLUFF control lamp "BNI0082" connected to Port 2 in orange in the running mode.

```
'Connection
Call Connect

Addition of the 'Extension
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
                    "Port=1, IODD= Balluff-BNI_IOL-802-102-Z036-20150730-IODD1.1.xml ")

'Add Variable
Dim var As CaoVariable
Set var = extension.AddVariable("@ProcessDataOut")

'Get Value
Dim processDataOutValues As Variant
processDataOutValues = var.Value

'Set operating mode to running mode
processDataOutValues(10) = 4
'Set the background color of segment 1 to orange
processDataOutValues(1) = 5

Write the 'value to the device
var.Value = processDataOutValues
```

3.3.2.3. IO-Link Devices Settings

3.3.2.3.1. <IODD FileVariables>

Obtain/configure IO-Link device configuration information. You can specify the variables that are defined in SpecialistRoleMenuSet of IODD file.

Data Type

Type	Description

VT_VARIANT	The @ProcessDataOut datatype depends on the specified IODD files. For the types of the specified IODD files "IODevice.ProfileBody.ProcessDataCollection.ProcessDataOut" See.

Sample IODD Files

The following figure defines IODD file variables in IODD file "Balluff-BNI_IOL-802-102-Z036-20150730-IODD1. 1.xml".

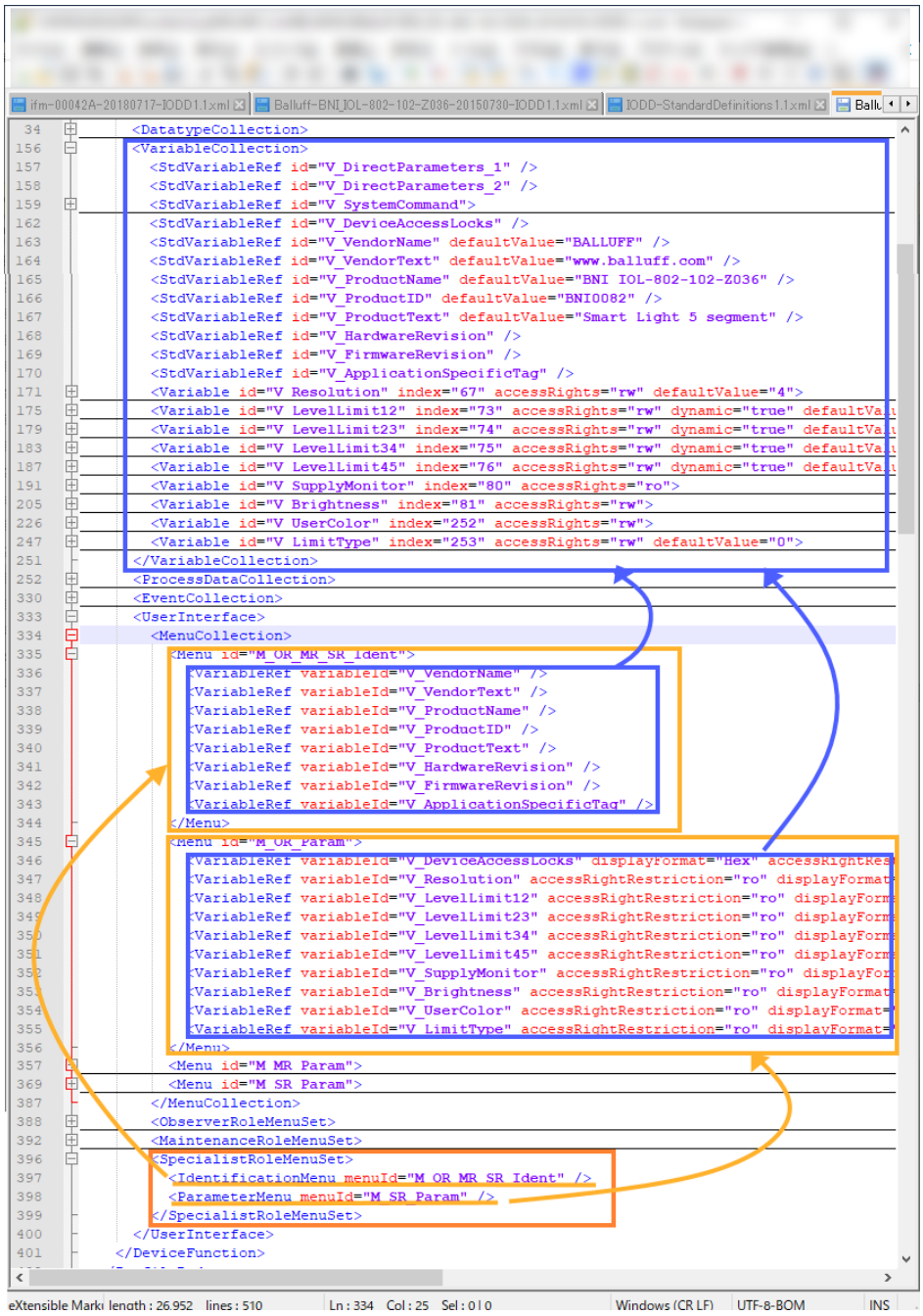


Fig. 3-3 Actual IODD variables definitions

Arrows indicate the referencing route for how to identify IODD files variables from SpecialistRoleMenuSet. Providers can add Variable or RecordItem in VariableCollection referenced by VariableRef and RecordItemRef in Menu corresponding to menuId defined in SpecialistRoleMenuSet as IODD file variables.

How to determine variable names

Variable names for IODD filevariables are created using text strings in ExternalTextCollection.PrimaryLanguage that are referenced by Name elements in their respective items according to the following rules:

If the target of VariableRef is not a Record type or if false is specified for subindexAccessSupport attribute of

Record type, the name of the variable is Variable. If the item is a RecordItemRef or VariableRef is referenced by a Record type and no false is specified in subindexAccessSupported attributes, the variable name is Variable name.RecordItem name.

If two or more IODD file variables have the same name, the name of the duplicate variable will be indexed ("_[index]" is appended) (see the example below).

- Example of Duplicate Variable Names

For example, if a Variable/RecordItem with different IDs results in the same variable name, the variable names are indexed in the order in which the duplicate Variable elements are defined, as follows: If a Variable/RecordItem with the same identifier is referenced in more than one Menu, the first one defined is used.

Variable ID in IODD	IODD Defined Names	Variable Names in Providers
Dummy_1	Dummy	Dummy
Dummy_2	Dummy	Dummy_2
Dummy_3	Dummy	Dummy_3

Examples of Actual IODD Fill Variables

The following table lists IODD file variables when IODD file "Balluff-BNI_IOL-802-102-Z036-20150730-IODD1. 1.xml" is loaded. The data types for each variable are also described.

Provider variable name	Data Type	
Vendor Name	VT_BSTR	
Vendor Text	VT_BSTR	
Product Name	VT_BSTR	
Product ID	VT_BSTR	
Product Text	VT_BSTR	
Hardware Version	VT_BSTR	
Firmware Version	VT_BSTR	
Application Specific Tag	VT_BSTR	
Device Access Locks	VT_ARRAY | VT_VARIANT	
	0	VT_BOOL: Parameter (write) Access Lock False: Unlock True: Lock
	1	VT_BOOL: Data Storage Lock False: Unlock True: Lock

Provider variable name	Data Type	
	2	VT_BOOL: Local Parameterization Lock False: Unlock True: Lock
	3	VT_BOOL: Local User Interface Lock False: Unlock True: Lock
Standard Command	VT_UI1	
Resolution of level indicator	VT_UI1 0: 8 bit 1: 10 bit 2: 12 bit 3: 14 bit 4: 16 bit	
Level mode, limit segment 1 and 2	VT_UI2	
Level mode, limit segment 2 and 3	VT_UI2	
Level mode, limit segment 3 and 4	VT_UI2	
Level mode, limit segment 4 and 5	VT_UI2	
Supply monitoring.Supply Voltage	VT_BOOL False: OK True: Low Us	
Supply monitoring.Supply Voltage LED	VT_BOOL False: OK True: Low LED Voltage	
Brightness.Red channel brightness	VT_UI1	
Brightness.Green channel brightness	VT_UI	
Brightness.Blue channel brightness	VT_UI1	
User defined color.Red channel	VT_UI1	
User defined color.Green channel	VT_UI1	
User defined color.Blue channel	VT_UI1	
Limit register type	VT_UI1 0: Percent 1: Absolute	

3.3.2.4. IO-Link Devices

3.3.2.4.1. @Status

Gets the status of connected IO-Link devices.

Data Type

Type Description	
VT_I4	Status of IO-Link devices 0x00: Not connected 0x02: be connected

Examples of use

```
'Connection
Call Connect

Addition of the 'Extension
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
                                        "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")

'Add Variable
Dim var As CaoVariable
Set var = extension.AddVariable("@Status")

' Value retrieval
Dim value As Long
value = var.Value
```

3.3.2.4.2. @VendorId

Get the manufacturer id of the connected IO-Link devices.

Data Type

Type Description	
VT_I4	Manufacturer IDs of IO-Link devices

Examples of use

```
'Connection
Call Connect

Addition of the 'Extension
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
                                        "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")

'Add Variable
Dim var As CaoVariable
Set var = extension.AddVariable("@VendorId")
```

```
' Value retrieval
Dim value As Long
value = var.Value
```

3.3.2.4.3. @DeviceId

Get the device id of the connected IO-Link device.

Data Type

Type Description	
VT_I4	Device identifier of IO-Link device

Examples of use

```
'Connection
Call Connect

Addition of the 'Extension
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
                                        "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")

'Add Variable
Dim var As CaoVariable
Set var = extension.AddVariable("@DeviceId")

' Value retrieval
Dim value As Long
value = var.Value
```

3.3.2.4.4. @ProductName

The product name of the connected IO-Link device is obtained.

Data Type

Type Description	
VT_BSTR	Product name of IO-Link devices

Examples of use

```
'Connection
Call Connect

Addition of the 'Extension
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
                                        "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")

'Add Variable
Dim var As CaoVariable
Set var = extension.AddVariable("@ProductName")
```

```
' Value retrieval
Dim value As String
value = var.Value
```

3.3.2.4.5. @Serial

Get the serial number of the connected IO-Link devices.

Data Type

Type Description	
VT_BSTR	Serial number of IO-Link devices

Examples of use

```
'Connection
Call Connect

Addition of the 'Extension
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
                                        "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")

'Add Variable
Dim var As CaoVariable
Set var = extension.AddVariable("@Serial")

' Value retrieval
Dim value As String
value = var.Value
```

3.3.2.4.6. @ApplicationTag

Gets/sets the application tags of connected IO-Link devices.

Data Type

Type Description	
VT_BSTR	Application Tags for IO-Link Devices

Examples of use

```
'Connection
Call Connect

Addition of the 'Extension
Dim extension As CaoExtension
Set extension = controller.AddExtension("sensor", _
                                        "Port=1, IODD=ifm-00042A-20180717-IODD1.1.xml")

'Add Variable
Dim var As CaoVariable
Set var = extension.AddVariable("@ApplicationTag")
```

```
' Value retrieval
Dim value As String
value = var.Value
```

4. Programming by the provider

The providers communicate with IO-Link master each time they retrieve and set variables as described above. There is no communication check when CaoController/CaoExtension is created.

4.1. Sample Programming to Control Tower Light

Here is a sample program for lighting BALLUFF control lamp "BNI0082" connected to Port 2 in the running mode in orange. The requirements for the sample program are described below.

Table 4-1 Sample program requirements

Requirements	Description
Host	192.168.0.1
	Port number is 1.
	IO-Link devices use Vendor: Balluff, Device: BNI0082
Processing content	Rewrites the Process Data Out value.

The following sections provide specific codes.

4.1.1. Sample program

An overview of the sample program is shown below.

Sample **ControlBNI0082.vb**

```
' Object
Dim engine As CaoEngine
Dim workspace As CaoWorkspace
Dim controller As CaoController
Dim extension As CaoExtension


Private Sub Main
    Dim processDataOut As CaoVariable

    ' Initialization
    Call Initialize

    Set processDataOut = extension.AddVariable("@ProcessDataOut")

   'Get Value
   Dim processDataOutValues As Variant
   processDataOutValues = var.Value
  'Set operating mode to running mode
  processDataOutValues(10) = 4
  'Set the background color of segment 1 to orange
  processDataOutValues(1) = 5
   Write the 'value to the device
   var.Value = processDataOutValues
```

```
    Call extension.Variables.Remove(processDataOut.Index)
    Set processDataOut = Nothing

    'Termination processing
    Call Finalize
End Sub

' Initialization method
Private Sub Initialize()
    ' Creation of CaoEngine objects
    Set engine = New CaoEngine
    ' Creation of CaoWorkspace objects
    Set workspace = engine.AddWorkspace("NewWrks", "")
    ' Creation of CaoController objects
    Set controller = workspace.AddController("Master1", _
                                             "CaoProv.ifm.IO-Link", _
                                             "", _
                                             "Server=192.168.0.1")
    ' Creation of CaoExtension objects
    Set extension = controller.AddExtension("BNI0082", _
                    "Port=1, IODD= Balluff-BNI_IOL-802-102-Z036-20150730-IODD1.1.xml "))

End Sub

' Termination method
Private Sub Finalize()
    ' remove CaoExtension from CaoController
    Call controller.Extensions.Remove(extension.Index)
    Set extension = Nothing

    ' remove CaoController from CaoWorkspace
    Call workspace.Controllers.Remove(controller.Index)
    Set controller = Nothing

    ' remove CaoWorkspace from CaoEngine
    Call engine.Workspaces.Remove(workspace.Index)
    Set workspace = Nothing

    ' Clear CaoEngine
    Set engine = Nothing
End Sub
```

5. Provider error code

This provider has the following unique error codes masked by the 0x8011****.

For common ORiN2 errors, refer to the error codes section in ORiN2 Programming Guide.

Table 5-1 Unique error code

Error Number	Description
0x80110001	A mandatory option was not specified. Options specified as required in the User's Guide cannot be omitted.
0x80110002	The specified option is out of value range. Specify a value within the value range shown in the user's guide.
0x80110003	The data type of the write data is invalid, and cannot be converted. Write data using the data type shown in the user's guide or a data type that can be converted in memory.
0x80110004	The data type of the received data is different from the assumption. Please contact us if this error occurs.
0x80110005	A CaoExtension with the same port number already exists. The same port number cannot be specified for different Extension. Delete one of them.
0x80111000	The specified IODD file or IODD-StandardDefinitions1. 1.xml does not exist. Review the path specification. If IODD-StandardDefinitions1. 1.xml file does not exist, place it under Bin¥XML¥.
0x80112000	The specified IODD files have different versions. IODD version supports only 1.1. If there are multiple versions, specify the version 1.1 IODD files.
0x801130**[2]	An unexpected error occurred while parsing ProcessDataIn data for IODD files. Check if your IODD file is an approved file.
0x801140**[2]	An unexpected error occurred while analyzing ProcessDataOut data of the IDD file. Check if your IODD file is an approved file.
0x801150**[2]	An unexpected error occurred while parsing Variable data for IODD files. Check if your IODD file is an approved file.
0x801160**[2]	An unexpected error occurred while analyzing Text data of the IDD file. Check if your IODD file is an approved file.
0x801170**[2]	An unexpected error occurred while parsing VariableCollection data for IODD files. Check if your IODD file is an approved file.
0x801180**[2]	An unexpected error occurred while analyzing DataType data of the IDD file. Check if your IODD file is an approved file.

[2] ** contains specific codes indicating what errors occurred. Refer to the attached table for details codes.

Table 5-2 Detailed error codes during IODD file data analysis

Error Number	Description
0x01	Required attribute does not exist.
0x02	Required element does not exist.
0x03	Attribute value out of range.
0x04	Element value is out of range.
0x05	The specified text does not exist.
0x06	The specified ProcessData does not exist.
0x07	The specified Variable does not exist.
0x08	The element with the specified Subindex does not exist.
0x09	The specified Datatype does not exist.
0x0A	An unsupported Datatype is specified.

It also returns an error code from IO-Link Master, masked with "0x8010****", and a Http error code masked with "0x8012****".

Appendix A. Correspondence between IODD data types and VARIANT data types

This is where you define the corresponding tables for VARIANT data types of IODD.

IODD Datatype	VARIANT Datatype
BooleaT	VT_BOOL
UIntegerT	$2 \leq bitlength \leq 8$: VT_UI1 $8 < bitlength \leq 16$: VT_UI2 $16 < bitlength \leq 32$: VT_UI4 $32 < bitlength \leq 64$: VT_UI8
IntegerT	$2 \leq bitlength \leq 8$: VT_I1 $8 < bitlength \leq 16$: VT_I2 $16 < bitlength \leq 32$: VT_I4 $32 < bitlength \leq 64$: VT_I8
Float32T	VT_R4
StringT	VT_BSTR
OctetStringT	VT_UI1 | VT_ARRAY
TimeT	VT_DATE
TimeSpanT	VT_I8

IODD Datatype	VARIANT Datatype
RecordT	VT_VARIANT | VT_ARRAY
ArrayT	Array of defined data types

Appendix B. Communication Protocol Command Correspondence Table

The providers use POST method of Rest API to communicate with IO-Link master. In the adr fields of the following tables for Get/Put of variables

Variable name	Values of JSON adr fields	
	Get	Put
CaoController variables		
@ProductCode	Deviceinfo/productcode/getdata	-
@Vendor	Deviceinfo/vendor/getdata	-
@DeviceFamily	Deviceinfo/devicefamily/getdata	-
@HWRevision	Deviceinfo/hwrevision/getdata	-
@SerialNumber	Deviceinfo/serialnumber/getdata	-
@SWRevision	Deviceinfo/swrevision/getdata	-
@BootLoaderRevision	Deviceinfo/bootloaderrevision/getdata	-
@ExtensionRevisions	Deviceinfo/extensionrevisions/getdata	-
@FieldbusType	Deviceinfo/fieldbustype/getdata	-
@IOLinkMasterName	Devicetag/applicationtag/getdata	Devicetag/applicationtag/setdata
@IOLinkMasterVersion	Firmware/version/getdata	-
@IOLinkMasterSWType	Firmware/type/getdata	-
@ContainerMaxSize	Firmware/container/maxsize/getdata	-
@ContainerChunkSize	Firmware/container/chunksize/getdata	-
@ContainerSize	Firmware/container/size/getdata	-
@Temperature	Processdatamaster/temperature/getdata	-
@Voltage	Processdatamaster/voltage/getdata	-
@Current	Processdatamaster/current/getdata	-
@SuperVisionStatus	Processdatamaster/supervisionstatus/	-

	getdata	
@Version	-	-
CaoExtension variables		
@SendProcDataPort	Iolinkmaster/port[n]/senddatatosmob/getdata	Iolinkmaster/port[n]/senddatatosmob/setdata
@TransCycleTimePreset	Iolinkmaster/port[n]/mastercycletime_preset/getdata	-
@TransCycleTimeAct	Iolinkmaster/port[n]/mastercycletime_actual/getdata	-
@Mode	Iolinkmaster/port[n]/mode/getdata	Iolinkmaster/port[n]/mode/setdata
@ComSpeed	Iolinkmaster/port[n]/comspeed/getdata	Iolinkmaster/port[n]/comspeed/setdata
@ValidationDataStorage	Iolinkmaster/port[n]/validation_datastorage_mode/getdata	Iolinkmaster/port[n]/validation_datastorage_mode/setdata
@ValidationVendorId	Iolinkmaster/port[n]/validation_vendorid/getdata	Iolinkmaster/port[n]/validation_vendorid/setdata
@ValidationDeviceId	Iolinkmaster/port[n]/validation_deviceid/getdata	Iolinkmaster/port[n]/validation_deviceid/setdata
@DataStorageMaxSize	Iolinkmaster/port[n]/datastorage/maxsize/getdata	-
@DataStorageChunkSize	Iolinkmaster/port[n]/datastorage/chunksize/getdata	-
@DataStorageSize	Iolinkmaster/port[n]/datastorage/size/getdata	-
@Pin2Input	Iolinkmaster/port[n]/pin2in/getdata	-
@ProcessDataIn	Iolinkmaster/port[n]/iolinkdevice/pdin/getdata	-
@ProcessDataOut	Iolinkmaster/port[n]/iolinkdevice/pdout/getdata	Iolinkmaster/port[n]/iolinkdevice/pdout/setdata
<IODD FileVariables>	Iolinkmaster/port[n]/iolinkdevice/iolreadacyclic	Iolinkmaster/port[n]/iolinkdevice/iolwriteacyclic
@Status	Iolinkmaster/port[n]/iolinkdevice/status/getdata	-
@VendorId	Iolinkmaster/port[n]/iolinkdevice/vendorid/getdata	-
@DeviceId	Iolinkmaster/port[n]/iolinkdevice/dev	-

	iceid/getdata	
@ProductName	Iolinkmaster/port[n]/iolinkdevice/productname/getdata	-
@Serial	Iolinkmaster/port[n]/iolinkdevice/serial/getdata	-
@ApplicationTag	Iolinkmaster/port[n]/iolinkdevice/applicationtag/getdata	Iolinkmaster/port[n]/iolinkdevice/applicationtag/setdata

Appendix C. List of Errors from IO-Link Master

Error Number () decimal value in	Description
0x0190(400)	Invalid request.
0x0191(401)	Authentication error. The SSL settings, user name, and password name may be different.
0x0193((403)	Access denied. The SSL settings, user name, and password name may be different.
0x0194(404)	Address not found.
0x01F4(500)	Internal server error.
0x01F7(503)	The service is not running. You may not have a IO-Link device connected to the port, but you are accessing IO-Link device information.
0x0212(530)	Input data is abnormal.
0x0213(531)	IO-Link failure. IO-Link master and IO-Link devices are in abnormal status.
0x0214(532)	PLC connection error. IO-Link master is already connected to the fieldbus PLC.

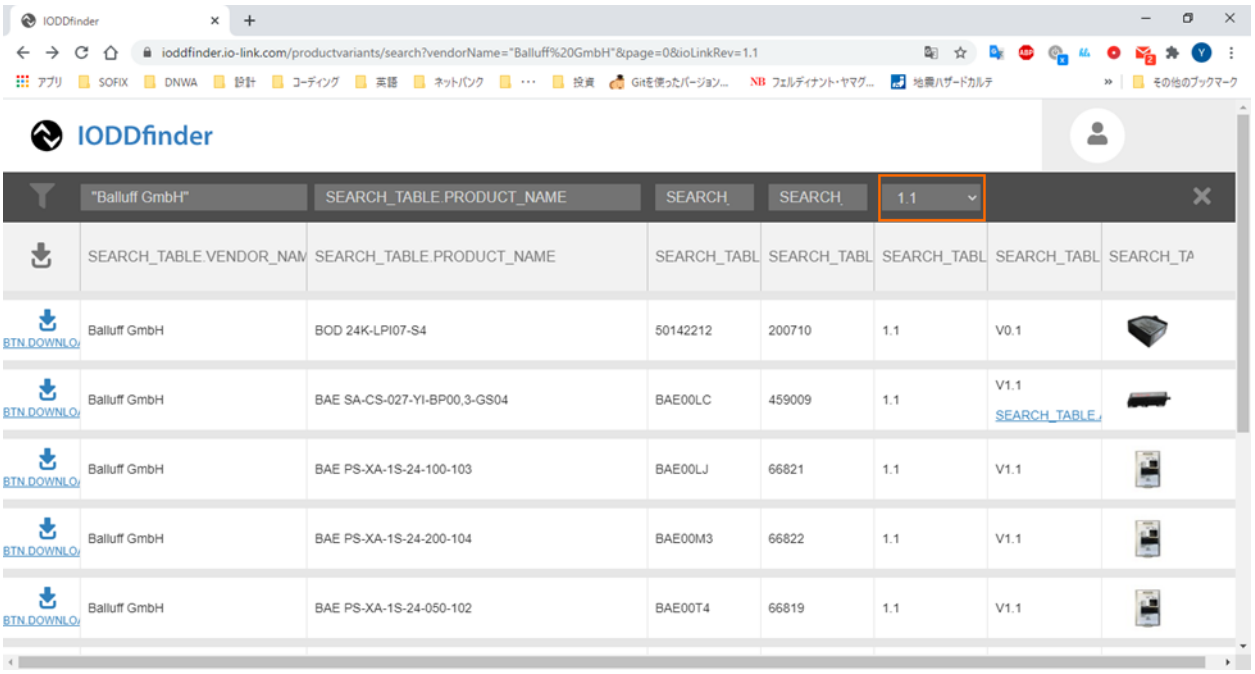
Appendix D. IODD file

How to Obtain IODD Files

To connect to a IO-Link device, a IODD(IO Device Description) file is required. Download the specified file from the following URL and use it.

https://ioddfinder.io-link.com

Note that the schema version supported by the provider is 1.1, so be careful when downloading. The following illustration shows an example of searching for IODD files from the above URLs.


IODDfinder
ioddfinder.io-link.com/productvariants/search?vendorName="Balluff%20GmbH"&page=0&ioLinkRev=1.1
IODDfinder
"Balluff GmbH"
SEARCH_TABLE.PRODUCT_NAME
SEARCH_
SEARCH_
1.1
SEARCH_TABLE.VENDOR_NAM
SEARCH_TABLE.PRODUCT_NAME
SEARCH_TABL
SEARCH_TABL
SEARCH_TABL
SEARCH_TABL
SEARCH_TA
Balluff GmbH
BOD 24K-LPI07-S4
50142212
200710
1.1
V0.1
Balluff GmbH
BAE SA-CS-027-YI-BP00,3-GS04
BAE00LC
459009
1.1
V1.1
SEARCH_TABLE
Balluff GmbH
BAE PS-XA-1S-24-100-103
BAE00LJ
66821
1.1
V1.1
Balluff GmbH
BAE PS-XA-1S-24-200-104
BAE00M3
66822
1.1
V1.1
Balluff GmbH
BAE PS-XA-1S-24-050-102
BAE00T4
66819
1.1
V1.1

How to distinguish IODD files

The root element of IODD file is <**IODevice**>, and the xsi:schemaLocation attribute of the <**IODevice**> element is described as "**http://www.io-link.com/IODD/2010/10 IODD1.1.xsd**". If an error occurs even though a IODD file is specified, check the contents of the specified file and check whether the file is normal.

Also, "IODD-StandardDefinitions.xml", which is included by default during provider installation, is not a IODD file, but a definition file referenced by all IODD files. Do not specify this as IODD option.