FL-net provider

JEMA FL-net communication

Version 1.0.1

User’s guide

October 6, 2020

[Remarks]

[Revision history]

Version	Date	Description
1.0.1.0	2017-01-07	First edition.
	2017-07-06	Manual was corrected.
	2020-10-6	Addition of license addition items.

[Supported models]

Model	Version	Note

【Attention】

Additional license for " FL-net Provider " is required to use this provider.

Contents

1. Introduction

This is a user ‘s guide of provider that enables to communicate with a variety of industrial equipment by using JEMA FL-net.

For information about FL-net, please refer to the Web site of the Japan Electrical Manufacturers' Association (JEMA).

This provider shows that the function of FL-net communication and method which is mounted on it.

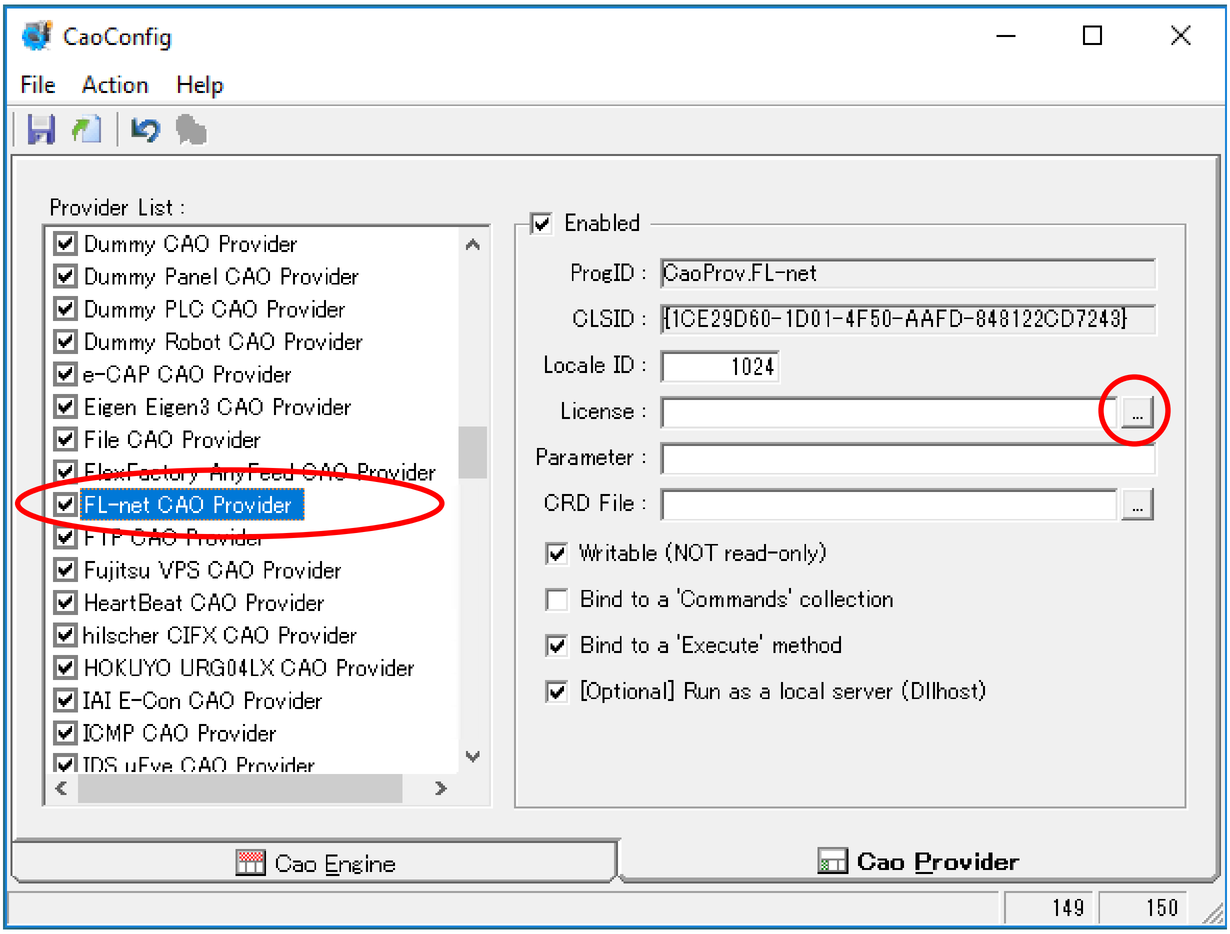
1.1. Installing license

To use OpenCV Provider, you need to install ORiN2 SDK, and also need to input “FL-net Provider” license information. If you would like to install it for evaluation, please use the following license.

FLE2-N2B6-VGX5-CWKL (valid for 3 months)

How to add the license is as follows.

1. Run the CaoConfig tool from the [Start] menu, and select the [Cao Provider] tab.
2. Select the [FL-net CAO Provider] item on the provider list.
3. Click the [...] button of the license input box.
4. Click the [Add] button in the "ORiN2 License Manager" window.
5. Input a license key, and click the [OK] button.
6. Click the [Close] button to exit.



Provider List :

☑ Dummy CAO Provider
☑ Dummy Panel CAO Provider
☑ Dummy PLC CAO Provider
☑ Dummy Robot CAO Provider
☑ e-CAP CAO Provider
☑ Eigen Eigen3 CAO Provider
☑ File CAO Provider
☑ FlexFactory AnyFeed CAO Provider
☑ FL-net CAO Provider
☑ FTP CAO Provider
☑ Fujitsu VPS CAO Provider
☑ HeartBeat CAO Provider
☑ hilscher CIFX CAO Provider
☑ HOKUYO URG04LX CAO Provider
☑ IAI E-Con CAO Provider
☑ ICMP CAO Provider
☑ IDS uEye CAO Provider

☑ Enabled

ProgID : CaoProv.FL-net

CLSID : {1CE29D60-1D01-4F50-AAFD-848122CD7243}

Locale ID : 1024

License : ...

Parameter :

CRD File : ...

☑ Writable (NOT read-only)

□ Bind to a 'Commands' collection

☑ Bind to a 'Execute' method

☑ [Optional] Run as a local server (Dllhost)

Figure 1-1 Installing ’ FL-net Provider ’ license

2. Outline of provider

2.1. Outline

FL-net provider is a CAO provider which enables operation according to ORiN by using CAO interface which wraps the communication to other company's FL-net-enabled devices. Its file format is DLL (Dynamic Link Library) and it will be loaded from CAO engine automatically at the time of operating.

Table 2-1 FL-net provider

File name	CaoProvFL-net.dll
ProgID	CaoProv.FL-net
Registration [1]	regsvr32 CaoProvFL-net.dll
Deregistration	regsvr32 /u CaoProvFL-net.dll

2.2. Method and Properties

2.2.1. CaoWorkspace::AddController method

FL-net provider performs FL-net communication processing when creating Controller object.

Syntax AddController(<bstrCtrlName:BSTR>,<bstrProvName:BSTR>,
<bstrPcName:BSTR > [,<bstrOption:BSTR>])

bstrCtrlName : [in] Controller name
bstrProvName : [in] Provider name. Fixed to =” CaoProv.FL-net”.
bstrPcName : [in] Computer name where provider runs.
bstrOption : [in] Option character strings

The following table shows a list of option character strings

Table 2-2 Option character strings of CaoWorkspace::AddController

Option	Description
NodeName[=<Node Name>]	Node name (within 10characters) Default: nil
MyIP[=<IP Adress>]	IP Address of EtherNet board to be used Default: IP Address of EtherNet board *Least significant digit of IP Address is self node No. e.g. 192.168.250.5 ← self node No. is 5

[1] Provider Registration is able to execute by regsvr32.exe, or RegCOM.exe ([Start]→[ORiN2]→[Tools])

	Note: If there are many EtherNet boards exist in a system, the system chooses a board automatically. Thus, it does NOT always guarantee to select EtherNet board which is you will use, so explicitly specifying the EtherNet board is highly recommended.
CommonAddr1[=<Common 1 Address>]	Common memory area 1 Address (0 to 511) Unit[WORD] Default: 0
CommonSize1[=<Common 1 Size>]	Common memory area 1 Size (0 to 512) Unit [WORD] Default: 0
CommonAddr2[=<Common 2 Address>]	Common memory area 2 Address (0 to 8191) Unit [WORD] Default: 0
CommonSize2[=<Common 2 Size>]	Common memory s area 2 Size (0 to 8192) Unit [WORD] Default: 0
TokenWdgTime[=<Token Watchdog Time>]	Token watchdog time (1 to 255) by 1 ms Default: 50
MinFrmInterval[=<Minimum Frame Interval>]	Minimum frame interval (0 to 50) by 100μs Default: 0

2.2.2. CaoController:: Execute method

For available command names and details, refer to 2.3.2.

Syntax Execute(< bstrCommand:BSTR > [,<vntParam:VARIANT>[,< pVal:VARIANT>]])

bstrCommand	: [in] Command name
vntParam	: [in] Parameter
pVal	: [out] Acquired data

2.2.3. CaoController::AddVariable method

This method creates a variable object. You can use variables written in 2.4.1 only for variable names. If other variable names are specified, an error occurs.

Syntax AddVariable(<bstrName:BSTR > [,<bstrOption:BSTR>])

bstrName	: [in] a unique name

bstrOption : [in] Option character strings

The following table shows available “Option character string ”.

Table 2-3 Option character strings of CaoController::AddVariable

Option	Description
Type[=<Variable type>]	Variable type (Default: 0) 0: Acquire/Configure data in self node common area 1: Acquire data in specified node common area 2: Acquire information of specified administration node
Node [=<Node No.]	Node number (Default: self node number*) Area: 1 to 254 Note: at the time of “Type=0”, it will be ignored.
Area[=<Common area>]	Specify common area 1 or 2 (Default: 1) Note: at the time of “Type=2”, it will be ignored.
Offset[=<Off set>]	Offset point (Default: 0) from the head of node common area Note: at the time of “Type=2”, it will be ignored.
Size[=<Data length]	Length of reading/writing data from Offset point (WORD unit Default: 1) Note: at the time of “Type=2”, it will be ignored.
Signed[=True / False]	Specify whether the data, which is acquired or configured, is signed or unsigned. True: Data type with sign False: Data type without sign (Default)

* For self node No., refer to My IP option in Table 2-2.

Table 2-4 List of functions of each “Type” Option

Variant type	Data type	Description	Attribute		Option				
			get	put	Node	Area	Offset	Size	Signed
“Type=0”	[When“Signed=False”] VT_ARRAY |VT_UI2 [When “Signed=True”] VT_ARRAY |VT_I2	Acquire/ Configure WORD-type data as per Size from Offset address within common area in the specified self node Area.	✓	✓	-	✓	✓	✓	✓

“Type=1”	[When “Signed=False”] VT_ARRAY |VT_UI2 [When “Signed=True”] VT_ARRAY |VT_I2 or VT_EMPTY	[When the specified node, which is self node or other node, is participating] Acquire WORD-type data as per Size from Offset address within common area in the specified self node Area. [When the specified node is other node and is not participating] VT_EMPTY	✓	-	✓	✓	✓	✓	✓
“Type=2”	[When “Signed=False”] VT_ARRAY |VT_UI2 [When “Signed=True”] VT_ARRAY |VT_I2	Acquire information of specified administration node. VT_ARRAY |VT_UI2 Array[0]: Common area 1 Address (0 to 511) Array[1]: Common area 1Size (0 to 512) Array[2]: Common area 2 Address (0 to 8191) Array[3]: Common area 2 Size (0 to 8192) Array[4]: Token watchdog time (1 to 255) Array[5]: Allowable minimum frame interval (0 to 50) Array[6]: Upper Layer Status (ULS) 0x2000: WARNING 0x4000: ALARM 0x8000: RUN/STOP Array[7]: Error code of ULS Array[8]: FA link state (LKS) 0x0001: Status of participation 0x0002:Communication invalidity detection 0x0010: Upper layer actuating signal error 0x0020: Common memory data validity notice 0x0040: Common memory setting completion 0x0080: Address overlap detection	✓	-	✓	-	-	-	✓

2.2.1. CaoController::GetVariableNames property

Obtain the system variable list in Table 2-6.

2.2.2. CaoVariable::get_Value property

Obtain information corresponding to variables. For implementation status and data to be set, refer to 2.4.1.

2.2.3. CaoVariable::put_Value property

Set information corresponding to variables. For implementation status and data to be set, refer to 2.4.1.

2.3. Command list

2.3.1. CaoController class

Table 2-5 CaoController::Execute Command list

Command name	Function	Page
ProviderCancel	Set the status of cancel	P.10
ProviderClear	Release the status of cancel	P.10
ClearLog	Clear Log information	P.10

2.3.2. Details of CaoController::Execute command

ProviderCancel

Syntax *object*.ProviderCancel()

Argument nil

Return value nil

Description Set a provider to a Cancel state.

Operational status of self node upper layer status (@ULS) becomes “STOP”.

While the status of cancel maintaining, obtaining/setting user variable is prohibited.

To release the cancel status, execute a “ProviderClear” command.

ProviderClear

Syntax *object*.ProviderClear()

Argument nil

Return value nil

Description Release a cancel status of a provider.

Operational status of self node upper layer status (@ULS) becomes other conditions except “STOP”.

ClearLog

Syntax `object.ClearLog()`

Argument nil

Return value nil

Description Clear Log information.

2.4. Variable list

2.4.1. Controller class

Table 2-6 Controller class system variable list

Variable name	Data type	Description	Attribute		Option				
			get	put	Node	Area	Offset	Size	Signed
@ENTRY	VT_BOOL	Status of participation of self node TRUE:Participation FALSE:Leaving	✓	-	-	-	-	-	-
@VENDOR_NAME	VT_BSTR	Self node vendor name	✓	-	-	-	-	-	-
@DEVICE_MODEL	VT_BSTR	Self node vendor model name	✓	-	-	-	-	-	-
@NODE_NAME	VT_BSTR	Self node name	✓	-	-	-	-	-	-
@UPPER_STATUS	[When “Signed=False”] VT_UI2 [When “Signed=True”] VT_I2	Operational state of Self node upper layer state 0x2000:WARNING 0x4000:ALARM 0x8000:RUN/STOP	✓	-	-	-	-	-	✓
@UPPER_ERROR_CODE	[When “Signed=False”] VT_UI2 [When “Signed=True”] VT_I2	ULS error code	✓	-	-	-	-	-	✓

@LINK_STATUS	[When “Signed=False”] VT_UI2 [When “Signed=True”] VT_I2	Self node FA link state 0x0001: State of participation 0x0002: Communication invalidity detection 0x0010: Upper layer actuating signal error 0x0020: Common memory data validity notice 0x0040: Common memory setting completion 0x0080: Address overlap detection	✓	-	-	-	-	-	✓
@NODE_LIST	[When “Signed=False”] VT_ARRAY | VT_UI1 [When “Signed=True”] VT_ARRAY | VT_I1	node No. participation	✓	-	-	-	-	-	✓
@OWN_STATUS	[When “Signed=False”] VT_UI2 [When “Signed=True”] VT_I2	Self node status 0x0008: Token watchdog time error 0x0010: Common memory area error 0x0020: Frame waiting 0x0040: Node No. duplicated 0x0080: Self node setting completed	✓	-	-	-	-	-	✓
@IP_ADDR	VT_BSTR	IP Address connecting	✓	-	-	-	-	-	-
@NET_MANAGE_PARAM	[When “Signed=False”] VT_ARRAY |VT_UI2 [When “Signed=True”] VT_ARRAY |VT_I2	Network administration information Array [0]: Token holding node No. (1 to 254) Array[1]: Minimum frame interval by 100µs Array[2]: Allowable Refresh Cycle Time (0 to 65535) by 1 ms Array[3]: Current value of Refresh Cycle Measurement Time (0 to 65535) by 1 ms Array[4]: Maximum value of Refresh Cycle Measurement Time (0 to 65535) by 1 ms Array[5]: Minimum value of Refresh Cycle Measurement Time (0 to 65535) by 1 ms	✓	-	-	-	-	-	✓

@PROFILE	VT_ARRAY|VT_BSTR	Profile information Array[0]: Profile common specification version Array[1]: Identifying system parameters Characters Array[2]: Modifying system parameters Numbers Array[3]: Changing system parameters Date Array[4]: Device type Array[5]: Manufacture name Array[6]: Product name	✓	-	-	-	-	-	-
@LOG_DATA	[When "Signed=False"] VT_ARRAY |VT_UI4 [When "Signed=True"] VT_ARRAY |VT_I4	Log information (128 [DWORD]) For details, see Table 2-7.	✓	-	-	-	-	-	✓

Table 2-7 Log information and location

ARRAY Offset [DWORD]	Name	Description
0	The total number of transmission at the socket part	Transmission and reception related items
1	The total number of errors of transmission at the socket part	Transmission and reception related items
2 to 5	-	Unused
6	The total number of reception at the socket part	Transmission and reception related items
7	The total number of reception errors at the socket part	Transmission and reception related items
8 to 18	-	Unused
19	Number of the events of receiving cyclic frame	Frame type related data
20 to 23	-	Unused
24	Number of errors in receiving cyclic frame	Cyclic transmission related data
25	-	Unused

26	Number of cyclic CBN errors	Cyclic transmission related data
27	Number of cyclic TBN errors	Cyclic transmission related data
28	Number of cyclic BSIZE errors	Cyclic transmission related data
29 to 35	-	Unused
36	Number of resending messages	Message transmission related data
37	Number of times message transmission resending was over	Message transmission related data
38 to 41	-	Unused
42	Number of errors in receiving message frame	Message transmission related data
43	Number of message sequence number errors	Message transmission related data
44	Number of recognizing resent message sequence numbers	Message transmission related data
45 to 47	-	Unused
48	Number of ACK errors	ACK related data
49	Number of ACK version of sequence number errors	ACK related data
50	Number of ACK sequence number errors	ACK related data
51 to 59	-	Unused
60	Number of detection of duplicated token	Token related data
61	Number of the events of discarding token	Token related data
62	Number of the events of reissuing token	Token related data
63 to 65	-	Unused
66	Number of the events of token retention timeout	Token related data
67	Number of the events of token watchdog timeout	Token related data
68 to 72	-	Unused
73	Number of frame waiting state	Operation state of node, date on joining and disengagement
74	Number of joining	Operation state of node, date on joining and disengagement
75	Number of self node disconnection	Operation state of node, date on joining and disengagement
76	Number of self node disconnection by skipped token	Operation state of node, date on joining and disengagement
77	Number of recognizing disconnection of other nodes	Operation state of node, date on joining and disengagement
78 to 127	-	Unused

Table 2-8 Controller class User variable list

Variable name	Data type	Description	Attribute		Option				
			get	put	Node	Area	Offset	Size	Signed
any	Depend on setting value of “Type” option. See Table 2-4.	See Table 2-4	See Table 2-4						

2.5. Error code

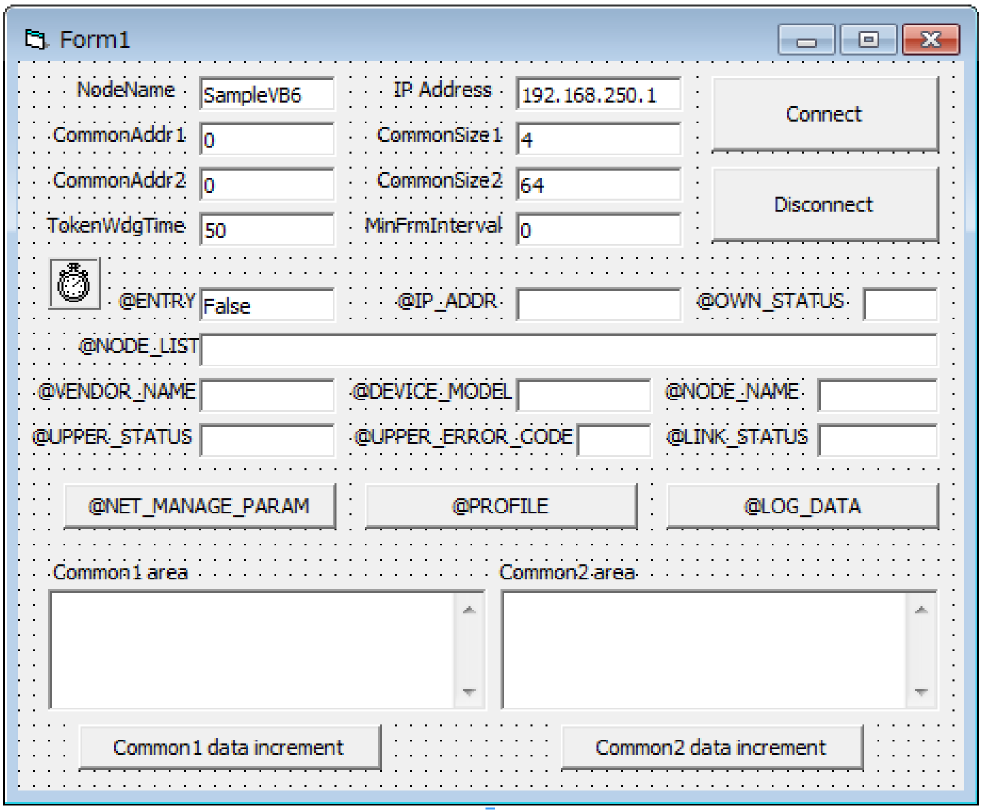
FL-net provider defines the original error codes as shown below.

Table 2-9 Original error code list

Error name	Error No.	Description
E_CAOP_FA_INIT	0x80100001	Initialization error: Fail to save the resource.
E_CAOP_FA_INIT_NETWORK	0x80100002	Initialization error: Fail to connect to the network
E_CAOP_FA_INIT_C1	0x80100003	Initialization error: Common memory area 1 setting error
E_CAOP_FA_INIT_C2	0x80100004	Initialization error: Common memory area 2 setting error
E_CAOP_FA_INIT_NODE_NUM	0x80100005	Initialization error: Node number setting error
E_CAOP_FA_INIT_ERR_TW	0x80100006	Initialization error: Token watchdog time setting error
E_CAOP_FA_INIT_ALREADY	0x80100007	Initialization error: Multiple connection error Note: Multiple nodes are not allowed to connect e.g., While connecting to node number 1, node number 2 is connected.
E_CAOP_FA_CFG	0x80100010	Execution error: Self node configuration unestablished
E_CAOP_FA_ADDRESS	0x80100011	Execution error: Address error
E_CAOP_FA_SIZE	0x80100012	Execution error: Size error
E_CAOP_FA_NODE_NOT_PARTICIPATE	0x80100013	Execution error: An unparticipated node exists

For ORiN2 common errors, refer to the error section in ORiN2programming guide.

3. Sample program



NodeName: SampleVB6
IP Address: 192.168.250.1
CommonAddr1: 0
CommonSize1: 4
CommonAddr2: 0
CommonSize2: 64
TokenWdgTime: 50
MinFrmInterval: 0
@ENTRY: False
@IP_ADDR:
@OWN_STATUS:
@NODE_LIST:
@VENDOR_NAME:
@DEVICE_MODEL:
@NODE_NAME:
@UPPER_STATUS:
@UPPER_ERROR_CODE:
@LINK_STATUS:

List 3-1 **Form1.frm**

```
Option Explicit

Private caoEng As CaoEngine
Private caoCtrls As CaoControllers
Private caoCntl As CaoController
Private caoVarEntry As CaoVariable

Private caoVarVendorName As CaoVariable
Private caoVarDeviceModel As CaoVariable
Private caoVarNodeName As CaoVariable
Private caoVarUpperStatus As CaoVariable
Private caoVarUpperErrorCode As CaoVariable
Private caoVarLinkStatus As CaoVariable
Private caoVarOwnStatus As CaoVariable
Private caoVarIPAddr As CaoVariable
Private caoVarNetManageParam As CaoVariable
Private caoVarProfile As CaoVariable
Private caoVarLogData As CaoVariable

Private caoVarNodeList As CaoVariable
Private caoVarCommon1 As CaoVariable
Private caoVarCommon2 As CaoVariable

Private Sub Form_Load()

    Set caoEng = New CaoEngine
    Set caoCtrls = caoEng.Workspaces(0).Controllers
    Timer1.Interval = 100
```

```
End Sub

Private Sub Form_Unload(Cancel As Integer)

    cmdDisconnect_Click

End Sub

' Connect
Private Sub cmdConnect_Click()

    Dim strOption As String

    ' Generate Option character strings of CaoWorkspace::AddController
    strOption = "NodeName=" & txtNodeName.Text & ",MyIP=" & txtIPAddress.Text & _
        ",CommonAddr1=" & txtCommonAddr1.Text & ",CommonSize1=" & txtCommonSize1.Text & _
        ",CommonAddr2=" & txtCommonAddr2.Text & ",CommonSize2=" & txtCommonSize2.Text & _
        ",TokenWdgTime=" & txtTokenWdgTime.Text & ",MinFrmInterval=" & txtMinFrmInterval.Text

    ' Execute CaoWorkspace::AddController
    Set caoCntl = caoCtrls.Add("", "CaoProv.FL-net", "", strOption)

    ' CaoController::AddVariable
    '   - System variable -
    Set caoVarEntry = caoCntl.AddVariable("@ENTRY", "")
    Set caoVarVendorName = caoCntl.AddVariable("@VENDOR_NAME", "")
    Set caoVarDeviceModel = caoCntl.AddVariable("@DEVICE_MODEL", "")
    Set caoVarNodeName = caoCntl.AddVariable("@NODE_NAME", "")
    Set caoVarUpperStatus = caoCntl.AddVariable("@UPPER_STATUS", "Signed=True")
    Set caoVarUpperErrorCode = caoCntl.AddVariable("@UPPER_ERROR_CODE", "Signed=True")
    Set caoVarLinkStatus = caoCntl.AddVariable("@LINK_STATUS", "Signed=True")
    Set caoVarNodeList = caoCntl.AddVariable("@NODE_LIST", "")
    Set caoVarOwnStatus = caoCntl.AddVariable("@OWN_STATUS", "Signed=True")
    Set caoVarIPAddr = caoCntl.AddVariable("@IP_ADDR", "")
    Set caoVarNetManageParam = caoCntl.AddVariable("@NET_MANAGE_PARAM", "Signed=True")
    Set caoVarProfile = caoCntl.AddVariable("@PROFILE", "")
    Set caoVarLogData = caoCntl.AddVariable("@LOG_DATA", "Signed=True")

    '   - Self node Common1 Area user variable for access -
    strOption = "Signed=True,Type=0,Area=1,Offset=0,Size=" & txtCommonSize1.Text
    Set caoVarCommon1 = caoCntl.AddVariable("Common1", strOption)

    '   - Self node Common2 Area user variable for access -
    strOption = "Signed=True,Type=0,Area=2,Offset=0,Size=" & txtCommonSize2.Text
    Set caoVarCommon2 = caoCntl.AddVariable("Common2", strOption)

    Timer1.Enabled = True

    txtNodeName.Enabled = False
    txtIPAddress.Enabled = False
    txtCommonAddr1.Enabled = False
    txtCommonSize1.Enabled = False
    txtCommonAddr2.Enabled = False
    txtCommonSize2.Enabled = False
    txtTokenWdgTime.Enabled = False
    txtMinFrmInterval.Enabled = False

    cmdDisconnect.Enabled = True
    cmdConnect.Enabled = False

End Sub

' Disconnect
Private Sub cmdDisconnect_Click()
```

```
        Timer1.Enabled = False

        DoEvents

        If Not caoVarCommon2 Is Nothing Then
            caoCntl.Variables.Remove caoVarCommon2.Index
            Set caoVarCommon2 = Nothing
        End If

        If Not caoVarCommon1 Is Nothing Then
            caoCntl.Variables.Remove caoVarCommon1.Index
            Set caoVarCommon1 = Nothing
        End If

        If Not caoVarLogData Is Nothing Then
            caoCntl.Variables.Remove caoVarLogData.Index
            Set caoVarLogData = Nothing
        End If

        If Not caoVarProfile Is Nothing Then
            caoCntl.Variables.Remove caoVarProfile.Index
            Set caoVarProfile = Nothing
        End If

        If Not caoVarNetManageParam Is Nothing Then
            caoCntl.Variables.Remove caoVarNetManageParam.Index
            Set caoVarNetManageParam = Nothing
        End If

        If Not caoVarIPAddr Is Nothing Then
            caoCntl.Variables.Remove caoVarIPAddr.Index
            Set caoVarIPAddr = Nothing
        End If

        If Not caoVarOwnStatus Is Nothing Then
            caoCntl.Variables.Remove caoVarOwnStatus.Index
            Set caoVarOwnStatus = Nothing
        End If

        If Not caoVarNodeList Is Nothing Then
            caoCntl.Variables.Remove caoVarNodeList.Index
            Set caoVarNodeList = Nothing
        End If

        If Not caoVarLinkStatus Is Nothing Then
            caoCntl.Variables.Remove caoVarLinkStatus.Index
            Set caoVarLinkStatus = Nothing
        End If

        If Not caoVarUpperErrorCode Is Nothing Then
            caoCntl.Variables.Remove caoVarUpperErrorCode.Index
            Set caoVarUpperErrorCode = Nothing
        End If

        If Not caoVarUpperStatus Is Nothing Then
            caoCntl.Variables.Remove caoVarUpperStatus.Index
            Set caoVarUpperStatus = Nothing
        End If

        If Not caoVarNodeName Is Nothing Then
            caoCntl.Variables.Remove caoVarNodeName.Index
            Set caoVarNodeName = Nothing
        End If

        If Not caoVarDeviceModel Is Nothing Then
            caoCntl.Variables.Remove caoVarDeviceModel.Index
            Set caoVarDeviceModel = Nothing
        End If

        If Not caoVarVendorName Is Nothing Then
            caoCntl.Variables.Remove caoVarVendorName.Index
            Set caoVarVendorName = Nothing
        End If

        If Not caoVarEntry Is Nothing Then
            caoCntl.Variables.Remove caoVarEntry.Index
            Set caoVarEntry = Nothing
        End If

        If Not caoCntl Is Nothing Then
            caoCtrls.Remove caoCntl.Index
            Set caoCntl = Nothing
        End If

        txtEntry.Text = "False"
        txtNodeList.Text = ""
        txtCommon1.Text = ""
        txtCommon2.Text = ""

        txtNodeName.Enabled = True
        txtIPAddress.Enabled = True
        txtCommonAddr1.Enabled = True
        txtCommonSize1.Enabled = True
        txtCommonAddr2.Enabled = True
        txtCommonSize2.Enabled = True
        txtTokenWdgTime.Enabled = True
        txtMinFrmInterval.Enabled = True

        cmdConnect.Enabled = True
        cmdDisconnect.Enabled = False

    End Sub

    Private Sub Timer1_Timer()

        Dim i As Integer
        Dim sMsg As String
        Dim vntVal As Variant

        ' Obtain @ENTRY system variable
        If CBool(caoVarEntry.Value) Then
            txtEntry.Text = "True"
        Else
            txtEntry.Text = "False"
        End If

        ' Obtain @IP_ADDR system variable
        txtIPAddr.Text = caoVarIPAddr.Value

        ' Obtain @OWN_STATUS system variable
        txtOwnStatus.Text = "0x" & Hex(caoVarOwnStatus.Value)

        ' @NODE_LIST system variable
        vntVal = caoVarNodeList.Value
        If VarType(vntVal) = (vbArray Or vbByte) Then
            For i = 0 To UBound(vntVal)
                sMsg = sMsg & vntVal(i)
                If i < UBound(vntVal) Then
                    sMsg = sMsg & ","
                End If
            Next
            txtNodeList.Text = sMsg
        End If

        ' Obtain @VENDOR_NAME system variable
        txtVendorName.Text = caoVarVendorName.Value

        ' Obtain @DEVICE_MODEL system variable
        txtDeviceModel.Text = caoVarDeviceModel.Value

        ' Obtain @NODE_NAME system variable
        txtNodeName.Text = caoVarNodeName.Value

        ' Obtain @UPPER_STATUS system variable
        txtUpperStatus.Text = "0x" & Hex(caoVarUpperStatus.Value)

        ' Obtain @UPPER_ERROR_CODE system variable
        txtUpperErrorCode.Text = "0x" & Hex(caoVarUpperErrorCode.Value)

        ' Obtain @LINK_STATUS system variable
        txtLinkStatus.Text = "0x" & Hex(caoVarLinkStatus.Value)

        ' Obtain self node Common1 area data
        vntVal = Empty
        vntVal = caoVarCommon1.Value
        If VarType(vntVal) = (vbArray Or vbInteger) Then
            sMsg = vbNullString
            For i = 0 To UBound(vntVal)
                sMsg = sMsg & vntVal(i)
                If i < UBound(vntVal) Then
                    sMsg = sMsg & ","
                End If
            Next
            txtCommon1.Text = sMsg
        End If

        ' Self node Common2 area data
        vntVal = Empty
        vntVal = caoVarCommon2.Value
        If VarType(vntVal) = (vbArray Or vbInteger) Then
            sMsg = vbNullString
            For i = 0 To UBound(vntVal)
                sMsg = sMsg & vntVal(i)
                If i < UBound(vntVal) Then
                    sMsg = sMsg & ","
                End If
            Next
            txtCommon2.Text = sMsg
        End If

    End Sub

    ' Common1 data increment
    Private Sub cmdCommon1DataIncrement_Click()

        Dim i As Integer
        Dim vntVal As Variant

        ' Upload self node Common1 area data
        vntVal = caoVarCommon1.Value
        If VarType(vntVal) = (vbArray Or vbInteger) Then
            For i = 0 To UBound(vntVal)
                vntVal(i) = vntVal(i) + 1
            Next
            caoVarCommon1.Value = vntVal
        End If

    End Sub
```

```
' Common2 data increment
Private Sub cmdCommon2DataIncrement_Click()

    Dim i As Integer
    Dim vntVal As Variant

    ' Upload self node Common2 area data
    vntVal = caoVarCommon2.Value
    If VarType(vntVal) = (vbArray Or vbInteger) Then
        For i = 0 To UBound(vntVal)
            vntVal(i) = vntVal(i) + 1
        Next
        caoVarCommon2.Value = vntVal
    End If

End Sub

' @NET_MANAGE_PARAM
Private Sub cmdNetManageParam_Click()

    Dim i As Integer
    Dim vntVal As Variant
    Dim sMsg As String

    ' Obtain @NET_MANAGE_PARAM
    vntVal = caoVarNetManageParam.Value
    If VarType(vntVal) = (vbArray Or vbInteger) Then
        For i = 0 To UBound(vntVal)
            sMsg = sMsg & "Array[" & i & "]: " & Str(vntVal(i)) & vbCrLf
        Next
        MsgBox sMsg, vbOKOnly, "@NET_MANAGE_PARAM"
    End If

End Sub

' @PROFILE
Private Sub cmdProfile_Click()

    Dim i As Integer
    Dim vntVal As Variant
    Dim sMsg As String

    ' Obtain @PROFILE
    vntVal = caoVarProfile.Value
    If VarType(vntVal) = (vbArray Or vbString) Then
        For i = 0 To UBound(vntVal)
            sMsg = sMsg & "Array[" & i & "]: " & vntVal(i) & vbCrLf
        Next
        MsgBox sMsg, vbOKOnly, "@PROFILE"
    End If

End Sub

' @LOG_DATA
Private Sub cmdLogData_Click()

    Dim i As Integer
    Dim j As Integer
    Dim vntVal As Variant
    Dim sMsg As String

    ' Obtain @LOG_DATA
    vntVal = caoVarLogData.Value
    If VarType(vntVal) = (vbArray Or vbLong) Then
        For i = 0 To UBound(vntVal) Step 8
```

```
                    sMsg = sMsg & "Array[" & i & "-" & (i + 7) & "]: "
                    For j = 0 To 7
                        If vntVal(i + j) < 0 Then
                            sMsg = sMsg & (CDec(vntVal(i + j)) + CDec(4294967296#))
                        Else
                            sMsg = sMsg & vntVal(i + j)
                        End If
                        If j < 7 Then
                            sMsg = sMsg & ", "
                        Else
                            sMsg = sMsg & vbCrLf
                        End If
                     Next
                Next
                MsgBox sMsg, vbOKOnly, "@LOG_DATA"
            End If

        End Sub
```

4. Restrictions

1. This product uses communication port from 55000 to 55004. When other applications use these ports, this provider will cause an error; so please stop these other applications.

2. Since this product uses communication port from 55000 to 55004, if Windows Firewall is valid, Firewall needs to be unblocked.

3. Processing such as display, printing and other conducts creates heavy load on CPU, and results in response decrement. Once response decrement of communication processing occurs, it could cause leaving from FL-net network and affect other nodes. If Network failure occurs many times, change the setting of network (e.g., Token watchdog time) to make it work correctly. However, if that change is not effective, reduce the heavy load, such as display, printing and other conduct on CPU.