

Dummy Panel provider

Version 1.1.0

User's guide

September 13, 2018

Remarks:

[Revision History]

Version	Date	Content
1.0.0	2015-09-01	First edition.
1.1.0	2018-09-13	Added system variable of CaoController class. {@ERROR_CODE, @BUSY_STATUS, @NORMAL_STATUS, @CURRENT_DATETIME, @RANDOM}
	2020-04-17	Deleted the installer-related description from Overview.

Contents

1. Introduction	4
2. Overview of provider	5
2.1. Overview	5
2.2. Method and Properties	6
2.2.1. CaoWorkspace::AddController method	6
2.2.2. CaoController::AddVariable method	6
2.2.3. CaoController::get_VariableNames property	7
2.2.4. CaoVariable::get_Value property	7
2.2.5. CaoVariable::put_Value property	7
2.2.6. CaoVariable::get_Attribute property	7
2.2.7. CaoMessage:: get_Source property	8
2.2.8. CaoMessage::get_Number property	8
2.2.9. CaoMessage::get_Value property	8
2.3. Variable list	9
2.3.1. Controller class	9
2.4. Setting the Ini file	10
2.4.1. Sample file	10

1. Introduction

This is a user's guide of Dummy Panel provider.

Dummy Panel provider allows to simulate the control panel's value acquisition and setup by using a virtual control panel without connecting a real device.

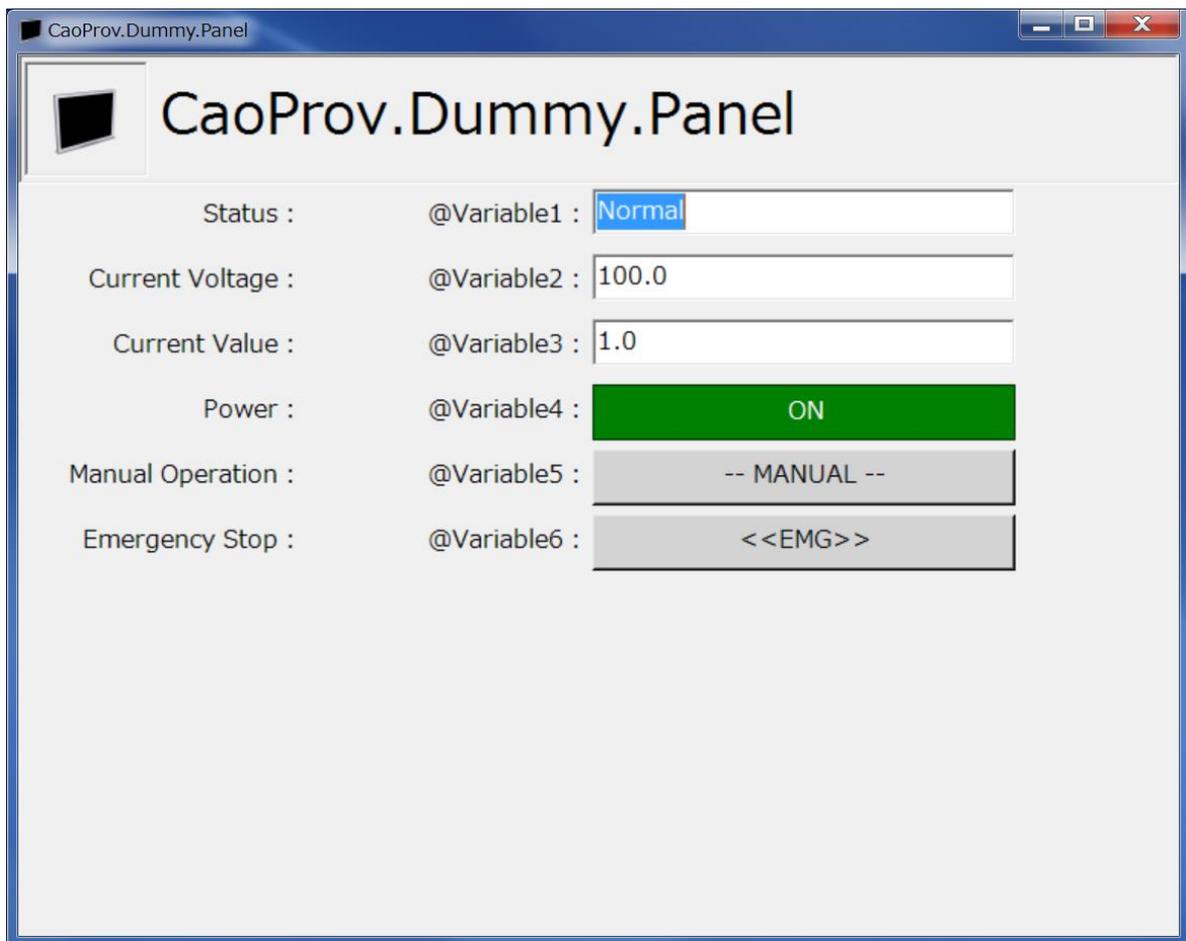


Figure 1 Dummy provider screen for Panel

2. Overview of provider

2.1. Overview

Dummy Panel provider is a CAO provider that allows to obtain and set data and to check the result by using a virtual control panel.

The file type is DLL (Dynamic Link Library) and is dynamically loaded by CAO engine when it is used.

Table 2-1 Dummy Panel provider

File name	CaoProvDummyPanel.dll
ProgID	CaoProv.Dummy.Panel

2.2. Method and Properties

2.2.1. CaoWorkspace::AddController method

Dummy Panel provider establishes a connection to the virtual control panel by referring to the parameters that have been passed at the AddController method execution.

Format	<pre>AddController(<bstrCtrlName:BSTR>,<bstrProvName:BSTR>, <bstrPCName:BSTR> [,<bstrOption:BSTR>])</pre>
<bstrCtrlName>	<p>: [in] Controller name Specify a unique arbitrary string for each connection. * An error (0x80000205) occurs if the same name is specified by a different application or computer. If an empty string ("") is entered, CAO engine automatically assigns a unique controller name.</p>
<bstrProvName>	<p>: [in] Option character string Fixed value = "CaoProv. Dummy.Panel"</p>
<bstrPcName>	<p>[in] Computer name to execute provider For a local connection, enter an empty string (""). To establish a remote connection, specify a target computer name.</p>
<bstrOption>	<p>[in] Option character string (not used)</p>

2.2.2. CaoController::AddVariable method

AddVariable method on CaoController class is a method to access variables. In this provider, you can specify a variable that can be obtained by CaoController::get_VariableNames property as a variable name. For information on how to define variables, refer to "2.4Setting the Ini file".

Please refer to "2.3 Variable list" for other system variables.

Format	<pre>AddVariable(<bstrName:BSTR > [,<bstrOption:BSTR>])</pre>
< bstrName >	<p>: [in] Variable name</p>
<bstrOption>	<p>: [in] Option character string (not used)</p>

2.2.3. CaoController::get_VariableNames property

Get the variable name list implemented by this provider as an array.

Format VariableNames([<bstrOption:BSTR>])
<bstrOption> : [in] Option character string (not used)

2.2.4. CaoVariable::get_Value property

Get a variable value corresponding to an object. The variable value obtained here is the value of the control panel's item corresponding to the variable name that has been specified in CaoController::AddVariable method of CaoController class.

2.2.5. CaoVariable::put_Value property

Set a variable value corresponding to an object. The variable value set here is the value of the control panel's item corresponding to the variable name that has been specified at the execution of CaoController::AddVariable method on CaoController class.

2.2.6. CaoVariable::get_Attribute property

Get the ON/OFF state of a toggle button by Integer type data. This property is available only when the variable name that has been specified at the execution of CaoController::AddVariable method on CaoController class is controlled/displayed by a toggle button.

Table 2-2 Values obtained by CaoVariable::get_Attribute property

State	Value
Toggle button ON	1
Toggle button OFF	0

2.2.7. CaoMessage::get_Source property

From the items displayed in the virtual control panel, get an item name that has been changed its value.

2.2.8. CaoMessage::get_Number property

If the state of an item displayed in the virtual control panel has been changed by a toggle button operation, the state of the item after change can be obtained by this property by Integer type. The value (Integer type) obtained here is the same as the one that is obtained by get_Attribute on CaoVariable class. For details, refer to “Table 2-2 Values obtained by CaoVariable::get_Attribute property”.

2.2.9. CaoMessage::get_Value property

If the state of an item displayed in the virtual control panel has been changed by a toggle button operation, the state of the item after change can be obtained by this property by Integer type. The value (Integer type) obtained here is the same as the one that is obtained by get_Attribute on CaoVariable class. For details, refer to “Table 2-2 Values obtained by CaoVariable::get_Attribute property”.

2.3. Variable list

2.3.1. Controller class

Table 2-1 Controller class variable list

Variable identifier	Data type	Explanation	Attribute	
			get	put
@CURRENT_DATETIME	VT_DATE	Current time	√	-
@BUSY_STATUS	VT_BOOL	true = During program operation, false = Program halted	√	-
@NORMAL_STATUS	VT_BOOL	true = normal, false = abnormal (error occurring) (Due to dummy operation, always true)	√	-
@ERROR_CODE	VT_I4	Acquires the number of the error occurring as a decimal value. If no error occurred, 0 is returned. (Due to dummy operation, always 0)	√	-
@RANDOM	VT_R8	Returns a random value between 0.0 and 1.0.	√	-

2.4. Setting the Ini file

You can set/change items displayed in the virtual control panel from the ini file.

To set a new item, create a new section with a form of “@<Arbitrary displayed item name>” in the ini file, and then specify it.

The ini file that lists the setup is stored in the following path

```
"<ORiN2 SDK installation folder>\CAO\ProviderLib\Dummy\Bin\Panel.ini"
```

```
[@<Arbitrary displayed item name>]
```

```
Type=<"Button" or "Text">
```

```
Caption=<item name>
```

```
InitialValue=<default>
```

```
InitialAttribute=<initial status of Toggle button ON: 1,OFF:0(Available only when Type is "Button")>
```

```
OnBackColor=<Background color when a toggle button is pressed (Available only when Type is "Button")>
```

```
OffBackColor=<Background color when a toggle button is not pressed (Available only when Type is "Button")>
```

```
OnForeColor=< Text color when a toggle button is pressed (Available only when Type is "Button")>
```

```
OffForeColor=< Text color when a toggle button is not pressed (Available only when Type is "Button")>
```

2.4.1. Sample file

Panel.ini

```
[Variable1]
```

```
Type=Text
```

```
Caption=Current Voltage
```

```
InitialValue=100.0
```

```
[Variable2]
```

```
Type=Button
```

```
Caption=Manual Operation
```

```
InitialValue=-- MANUAL --
```

```
InitialAttribute=0
```

```
OnBackColor=Yellow
```

```
OnForeColor=White
```

```
OffBackColor=LightGray
```

```
OffForeColor=Black
```