

Dummy CNC provider

Version 1.0.0

User's guide

September 13, 2018

Remarks:

[Revision History]

Version	Date	Content
1.0.0	2018-09-13	First edition.
	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::AddRobot method	6
2.2.3. CaoController::AddTask method	7
2.2.4. CaoController::AddVariable method	7
2.2.5. CaoController::get_TaskNames property	7
2.2.6. CaoController::get_VariableNames property	7
2.2.7. CaoRobot::AddVariable method	7
2.2.8. CaoRobot::get_VariableNames property	8
2.2.9. CaoTask::AddVariable method	9
2.2.10. CaoTask::get_VariableNames property	9
2.2.11. CaoTask::Start method	9
2.2.12. CaoTask::Stop method	9
2.2.13. CaoVariable::get_Value property	9
2.3. Variable list	10
2.3.1. Controller class	10
2.3.2. Task class	10
2.4. Setting of Ini file	11
2.4.1. Sample file	12

1. Introduction

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

Dummy CNC provider allows to simulate the CNC values acquisition by using a virtual CNC without connecting to a real CNC.

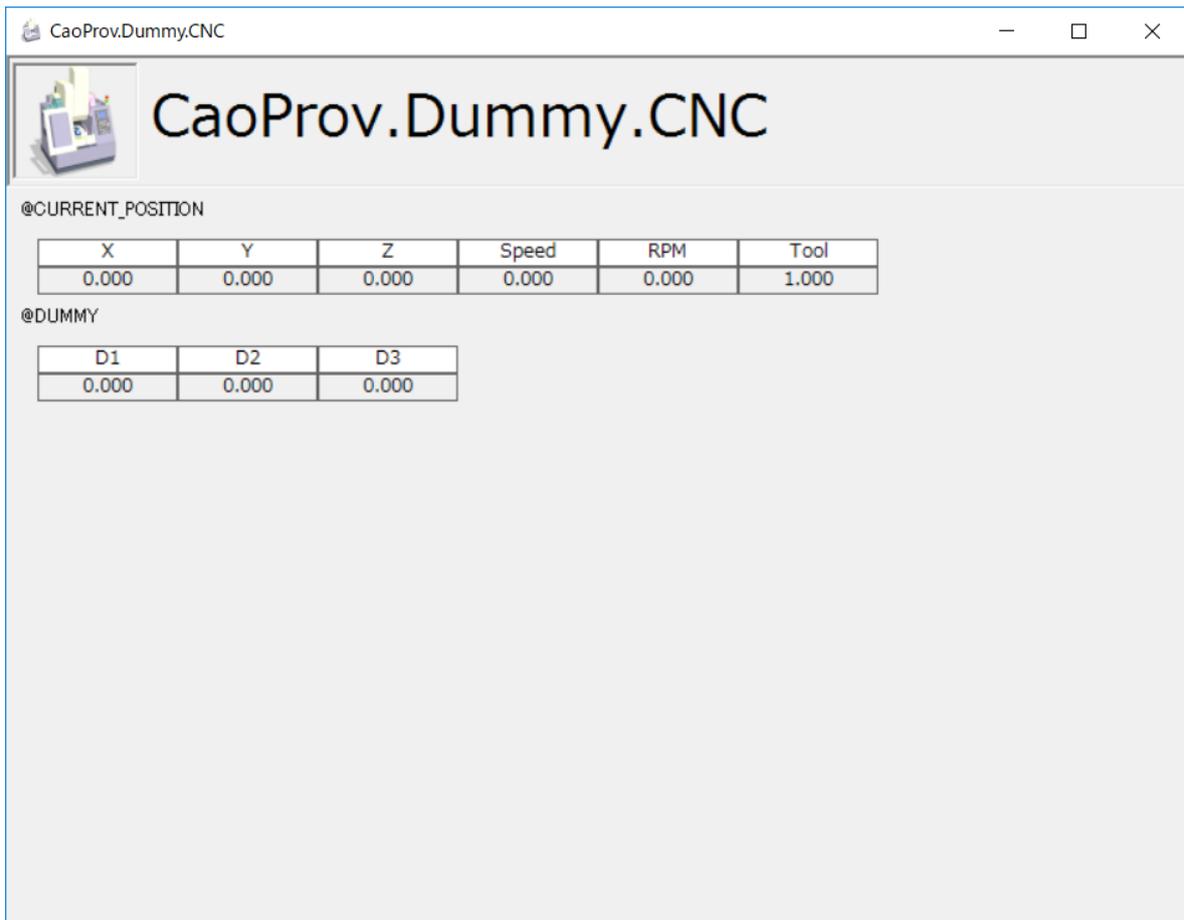


Figure 1 Dummy provider screen for CNC

2. Overview of provider

2.1. Overview

Dummy CNC provider is a CAO provider that allows to obtain data and to check result by connecting to a dummy CNC.

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

Table2-1 Dummy CNC provider

File name	CaoProvDummyCNC.dll
ProgID	CaoProv.Dummy.CNC

2.2.3. CaoController::AddTask method

For an argument of AddTask method on CaoController class, specify a task name (BSTR type). "Task name" specified here should be a task name that can be obtained by CaoController::get_TaskNames property.

Format AddTask(<bstrName:BSTR > [,<bstrOption:BSTR>])
 <bstrName> : [in] Task name
 <bstrOption> [in] Option character string (not used)

2.2.4. CaoController::AddVariable method

For an argument of AddVariable method on CaoController class, specify a variable name (BSTR type). "Variable name" specified here should be a task name that can be obtained by CaoController::get_VariableNames property.

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

2.2.5. CaoController::get_TaskNames property

Obtain the list of task names that can be specified by CaoController::AddTask method.

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

2.2.6. CaoController::get_VariableNames property

Obtain the list of variable names that can be specified by CaoController::AddVariable method.

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

2.2.7. CaoRobot::AddVariable method

For an argument of AddVariable method on CaoRobot class, specifies a variable name. "Variable name" specified here should be a variable name that can be obtained by CaoRobot::get_VariableNames property. For details about how to define a variable, refer to "2.4 Setting of Ini file".

Format AddVariable(<bstrName:BSTR > [,<bstrOption:BSTR>])
 <bstrName> : [in] Variable name (Section name that is defined by the form

<bstrOption> of”@<arbitrary data classification>” in the ini file)
[in] Option character string (not used)

2.2.8. CaoRobot::get_VariableNames property

Obtain a variable name list that can be specified by CaoRobot::AddVariable method. “Variable name” specified here should be a variable name that has been defined in the ini file.

For the details about how to define a variable , refer to “2.4 Setting of Ini file”.

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

2.2.9. CaoTask::AddVariable method

For an argument of AddVariable method on CaoTask class, specify a variable name (BSTR type).

See Table 2-3 for a list of implemented variables.

2.2.10. CaoTask::get_VariableNames property

Obtain a variable name list that can be specified by CaoTask::AddVariable method.

2.2.11. CaoTask::Start method

Run a program that corresponds to an object.

The following shows the argument specifications of Start.

Format	Start(<lMode:long > [,<bstrOption:BSTR>])
<lMode>	: [in] Mode (an arbitrary long value)
<bstrOption>	[in] Option character string (not used)

2.2.12. CaoTask::Stop method

Stop a program that corresponds to an object.

The following shows the argument specifications of Stop.

Format	Stop(<lMode:long > [,<bstrOption:BSTR>])
<lMode>	: [in] Mode (an arbitrary long value)
<bstrOption>	[in] Option character string (not used)

2.2.13. CaoVariable::get_Value property

Obtain a variable value corresponding to an object. The variable's value obtained here is a variable value (Double type array) that has been specified at the execution of AddVariable on CaoRobot class.

2.3. Variable list

2.3.1. Controller class

Table 2-2 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.	√	-

2.3.2. Task class

Table 2-3 Task class variable list

Variable identifier	Data type	Explanation	Attribute	
			get	put
@START	VT_BOOL	Task execution. When get (get) = True: Running, False: Stopping Setting (put) = Execution starts without depending on the value.	√	√
@STOP	VT_BOOL	Task stop. When get (get) = True: Stopped, False: Running When setting (put) = stops independent of value.	√	√

2.4. Setting of Ini file

You can set/change the behavior of the virtual CNC motion in the Data section of the ini file. The CNC motion data used here should be a CSV file.

To configure CNC motion data, create a new section with the format of ”@<arbitrary data classification>”, 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%CNC.ini"
```

```
[data]
```

```
datapath=<Path of a folder that contains CNC motion data>
```

```
interval=<Interval before displaying the next CNC data (ms)>
```

```
[@<arbitrary data classification>]
```

```
datafile=<CSV file name that stores CNC motion data>
```

```
columns=<Motion destination position name>[,< Motion destination position name >]
```

2.4.1. Sample file

By the setting of the following sample file, a CSV file in the “CNC.Data” folder under the operation environment will be the CNC motion data. In this setting, a CNC moves each motion destination position by 100ms.

Based on the "@CURRENT_POSITION" section, the CNC motion will refer data stored in "position.csv" file. The motion destination positions will be named in order from the left (X,Y,Z,Speed,RPM,Tool).

CNC.ini

```
[data]
```

```
datapath=¥CNC.Data
```

```
interval=100
```

```
[@CURRENT_POSITION]
```

```
datafile=position.csv
```

```
columns=X,Y,Z,Speed,RPM,Tool
```

position.csv

```
1156.91 ,498.56 , -298.00 ,100.00 ,3000 ,1
```

```
1150.08 ,498.56 , -298.00 ,100.00 ,3000 ,1
```

```
1142.71 ,498.56 , -298.00 ,100.00 ,3000 ,1
```

```
1134.83 ,498.56 , -298.00 ,100.00 ,3000 ,1
```

```
1126.41 ,498.56 , -298.00 ,100.00 ,3000 ,1
```

```
1117.47 ,498.56 , -298.00 ,100.00 ,3000 ,1
```