

# Dummy Robot プロバイダ

Version 1.2.0

## ユーザーズ ガイド

April 14, 2021

【備考】

**【改版履歴】**

バージョン	日付	内容
1.0.0	2015-09-01	初版.
1.0.1	2016-12-26	概要のインストーラー関連の記述を削除.
1.1.0	2018-09-13	CaoController クラスのシステム変数追加. {@ERROR_CODE,@BUSY_STATUS,@NORMAL_STATUS,@CURRENT_DATETIME,@RANDOM} CaoTask クラスのシステム変数追加. {@START,@STOP}
1.1.1	2020-12-01	TASK 変数追加. {@ELAPSED_TIME,@STATUS} @START 変数変更. {@START=n(0~N-1)} OneCycle 動作対応.
1.2.0	2021-04-14	複数起動対応.

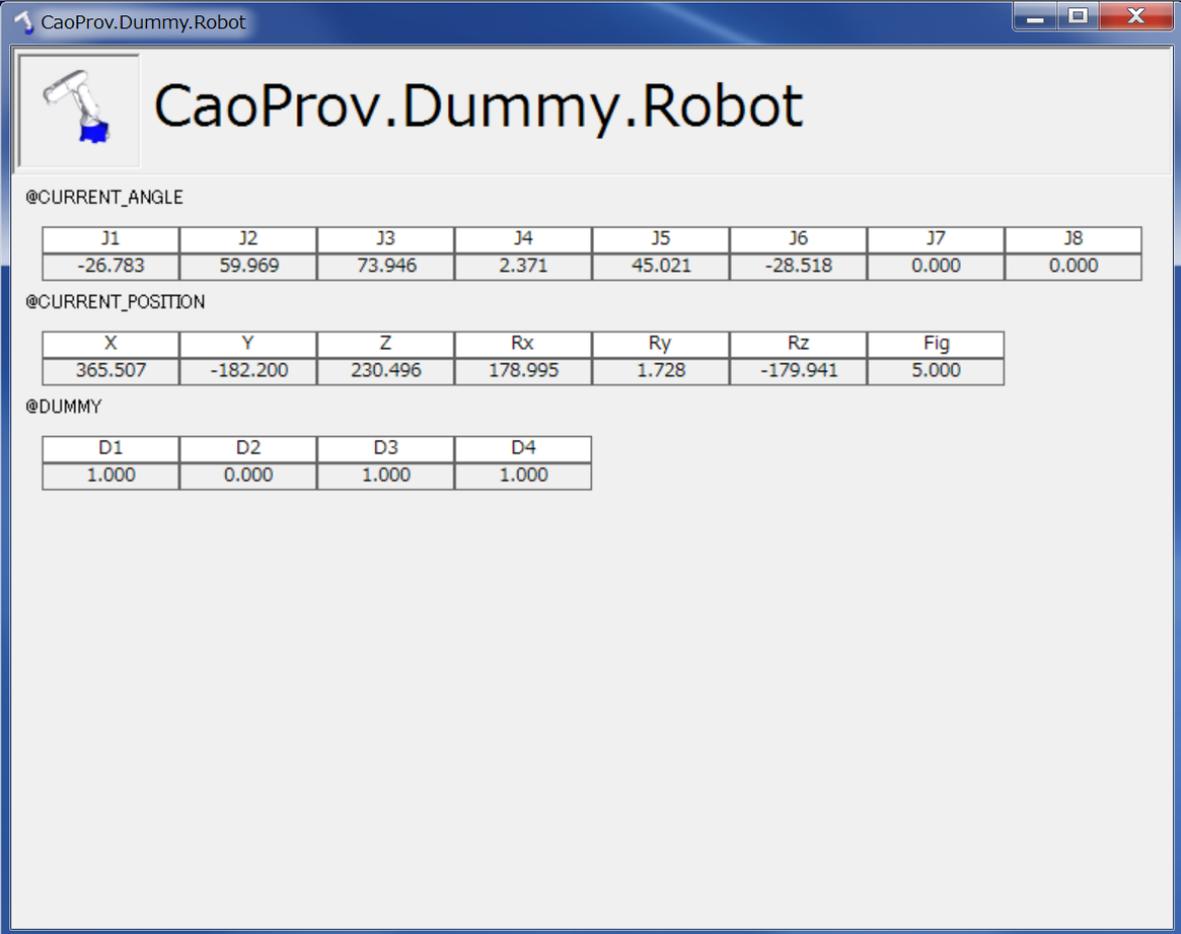
## 目次

1. はじめに.....	4
2. プロバイダの概要 .....	5
2.1. 概要.....	5
2.2. メソッド・プロパティ .....	6
2.2.1. CaoWorkspace::AddController メソッド.....	6
2.2.2. CaoController::AddRobot メソッド.....	7
2.2.3. CaoController::AddTask メソッド.....	7
2.2.4. CaoController::AddVariable メソッド .....	7
2.2.5. CaoController::get_TaskNames プロパティ .....	7
2.2.6. CaoController::get_VariableNames プロパティ.....	7
2.2.7. CaoRobot::AddVariable メソッド .....	8
2.2.8. CaoRobot::get_VariableNames プロパティ.....	8
2.2.9. CaoTask::AddVariable メソッド .....	8
2.2.10. CaoTask::get_VariableNames プロパティ.....	8
2.2.11. CaoTask::Start メソッド .....	8
2.2.12. CaoTask::Stop メソッド .....	8
2.2.13. CaoVariable::get_Value プロパティ .....	9
2.3. 変数一覧 .....	10
2.3.1. コントローラクラス.....	10
2.3.2. タスククラス.....	10
2.4. ini ファイルの設定 .....	10
2.4.1. サンプルファイル.....	11

## 1. はじめに

本書は Dummy Robot プロバイダのユーザーズガイドです。

Dummy Robot プロバイダでは、実際のロボットに接続せず、ダミーのロボットに接続して、ロボット各値の取得をシミュレートすることができます。



The screenshot shows the following data tables:

@CURRENT_ANGLE							
J1	J2	J3	J4	J5	J6	J7	J8
-26.783	59.969	73.946	2.371	45.021	-28.518	0.000	0.000

@CURRENT_POSITION						
X	Y	Z	Rx	Ry	Rz	Fig
365.507	-182.200	230.496	178.995	1.728	-179.941	5.000

@DUMMY			
D1	D2	D3	D4
1.000	0.000	1.000	1.000

図 1 Robot 用ダミープロバイダの画面

## 2. プロバイダの概要

### 2.1. 概要

Dummy Robot プロバイダは、ダミーのロボットに接続し、データの取得、ダミーロボットのデータを確認する機能を提供する CAO プロバイダです。

そのファイル形式は DLL(Dynamic Link Library)であり、CAO エンジンから使用時に動的にロードされます。

表 2-1 Dummy Robot プロバイダ

ファイル名	CaoProvDummyRobot.dll
ProgID	CaoProv.Dummy.Robot



### 2.2.2. CaoController::AddRobot メソッド

AddRobot メソッドを呼び出すと CaoRobot オブジェクトが取得できます。CaoController クラスの AddRobot メソッドの引数は、ロボット名(BSTR 型)を指定します。ここで指定する"ロボット名"は任意の文字列で特に何の制限もありません。例えば、AddRobot ("Robot1")を指定することができます。

**書式** AddRobot( <bstrName:BSTR > [,<bstrOption:BSTR>] )  
<bstrName> : [in] ロボット名  
<bstrOption> [in] オプション文字列 (未使用)

### 2.2.3. CaoController::AddTask メソッド

CaoController クラスの AddTask メソッドの引数は、タスク名(BSTR 型)を指定します。ここで指定する"タスク名"は CaoController::get\_TaskNames プロパティにより取得可能なタスク名を指定してください。

**書式** AddTask( <bstrName:BSTR > [,<bstrOption:BSTR>] )  
<bstrName> : [in] タスク名  
<bstrOption> [in] オプション文字列 (未使用)

### 2.2.4. CaoController::AddVariable メソッド

CaoController クラスの AddVariable メソッドの引数は、変数名を指定します。ここで指定する"変数名"は CaoController::get\_VariableNames プロパティにより取得可能な変数名を指定してください。

**書式** AddVariable( <bstrName:BSTR > [,<bstrOption:BSTR>] )  
<bstrName> : [in] システム変数名  
<bstrOption> [in] オプション文字列 (未使用)

### 2.2.5. CaoController::get\_TaskNames プロパティ

CaoController::AddTask メソッドで指定できるタスク名の一覧を取得します。

**書式** TaskNames( [<bstrOption:BSTR>] )  
<bstrOption> [in] オプション文字列 (未使用)

### 2.2.6. CaoController::get\_VariableNames プロパティ

CaoController::AddVariable メソッドで指定できる変数名の一覧を取得します。

**書式** VariableNames( [<bstrOption:BSTR>] )  
<bstrOption> [in] オプション文字列 (未使用)

### 2.2.7. CaoRobot::AddVariable メソッド

CaoRobot クラスの CaoRobot::AddVariable メソッドの引数は、変数名を指定します。ここで指定する"変数名"は 2.2.8CaoRobot::get\_VariableNames プロパティにより取得可能な変数名を指定してください。

変数の定義方法は「2.4.ini ファイルの設定」を参照して下さい。



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

<bstrName> : [in] システム変数名  
(ini ファイルにて”@<任意のデータ区分>”の形式で定義されている  
セクション名)

<bstrOption> [in] オプション文字列 (未使用)

### 2.2.8. CaoRobot::get\_VariableNames プロパティ

CaoRobot::AddVariable メソッドで指定できる変数名の一覧を取得します。ここで指定できる”変数名”は ini ファイル内で定義した変数名を指定してください。

変数の定義方法は「2.4.ini ファイルの設定」を参照して下さい。



VariableNames( [<bstrOption:BSTR>] )

<bstrOption> [in] オプション文字列 (未使用)

### 2.2.9. CaoTask::AddVariable メソッド

CaoTask クラスの AddVariable メソッドの引数は、システム変数名を指定します。

実装されているシステム変数の一覧は表 2-4 を参照して下さい。

### 2.2.10. CaoTask::get\_VariableNames プロパティ

AddVariable メソッドで指定できる変数名とシステム変数名の一覧を取得します。

### 2.2.11. CaoTask::Start メソッド

オブジェクトに対応しているプログラムを実行します。

以下に、Start の仕様を示します。



Start( <lMode:long > [,<bstrOption:BSTR>] )

<lMode> : [in] モード (任意の long 値)

<bstrOption> [in] オプション文字列 (未使用)

### 2.2.12. CaoTask::Stop メソッド

オブジェクトに対応しているプログラムを停止します。

以下に、Stop の引数仕様を示します。



Stop( <lMode:long > [,<bstrOption:BSTR>] )

<lMode> : [in] モード (任意の long 値)

<bstrOption> [in] オプション文字列 (未使用)

### 2.2.13. CaoVariable::get\_Value プロパティ

オブジェクトに対応している変数の値を取得します。ここで取得される変数の値は CaoRobot クラスの CaoRobot::AddVariable メソッド実行時に指定した変数の値(double 型配列)です。

変数の実装状況およびデータ型は「2.4.ini ファイルの設定」を参照して下さい。

## 2.3. 変数一覧

### 2.3.1. コントローラクラス

表 2-3 コントローラクラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@CURRENT_DATETIME	VT_DATE	現在時刻	○	-
@BUSY_STATUS	VT_BOOL	true=プログラム動作中, false=プログラム停止中	○	-
@NORMAL_STATUS	VT_BOOL	true=正常, false=異常(エラー発生中) (ダミー動作のため, 常時 true)	○	-
@ERROR_CODE	VT_I4	発生中のエラーの番号を 10 進数の値で取得します. エラーが発生していない時は, 0 を返します. (ダミー動作のため, 常時 0)	○	-
@RANDOM	VT_R8	0.0~1.0 のランダムな値を返します.	○	-

### 2.3.2. タスククラス

表 2-4 タスククラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@START	VT_BOOL	タスク実行. 取得(get)時=True: 実行中, False: 停止中 設定(put)時= n(0~N-1)書き込みで datafile の n 番 目の CSV で実行を開始します.	○	○
@STOP	VT_BOOL	タスク停止. 取得(get)時=True: 停止中, False: 実行中 設定(put)時= 値に依存せず停止します.	○	○
@ELAPSED_TIME	VT_R8	スタート後の経過時間(ms)	○	-
@STATUS	VT_I4	タスクの状態(0:停止, 1:実行中)	○	-

## 2.4. ini ファイルの設定

ダミーのロボットの動作の挙動を ini ファイルの data セクション内にて設定・変更することができます. ロボッ

トの動作データは CSV ファイルにて作成されたものが適用されます。

またロボットの動作データに対しての設定を”@<任意のデータ区分名>”の形式で新たにセクションを作成し、指定することができます。

設定が記載されている ini ファイルは以下のパス上に存在します。

"<ORiN2 SDK インストールフォルダ>¥CAO¥ProviderLib¥Dummy¥Bin¥Robot.ini"

[data]

datapath=<ロボットの動作データが存在しているフォルダへのパス>

interval=<次ロボットデータを表示するまでのインターバル(ms)>

tasklist=<タスク名>[,<タスク名>]

[@<任意のデータ区分名>]

datafile=<ロボット動作データが記載された CSV ファイル名>[,<CSV ファイル名>]

columns=<データ点名>[,<データ点名>]

loop=<実行回数>

#### 2.4.1. サンプルファイル

下記サンプルファイルの設定により、動作環境内の”Robot.Data”フォルダ内の CSV ファイルをロボットの動作データとし、また動作データ内のデータ点の移行を 100ms で行う設定となります。

また”@CURRENT\_ANGLE”セクションにより、ロボットの動作は”angle.csv”ファイル内に記載されているデータを参照し、表示する点名は左から”J1”, ”J2”, ”J3”, ”J4”, ”J5”, ”J6”, ”J7”, ”J8”となるように設定されています。

##### Robot.ini

[data]

datapath=¥Robot.Data

interval=100

[@CURRENT\_ANGLE]

datafile=angle.csv

columns=J1,J2,J3,J4,J5,J6,J7,J8

##### angle.csv

-26.783,38.125,82.602,1.732,58.197,-27.682,0,0

-26.704,38.125,82.602,1.729,58.196,-27.602,0,0

-26.443,38.124,82.603,1.721,58.192,-27.337,0,0

