

# FIELD system プロバイダ

Version 1.0.0

## ユーザーズ ガイド

July 10, 2020

備考:

### 【改版履歴】

バージョン	日付	内容
1.0.0	2020-07-10	初版

### 【動作確認機器】

機種	注意事項
FIELD system Box	

## 目次

1. はじめに .....	4
2. アプリケーション開発のための環境セットアップ .....	6
2.1. FIELD system Box とクライアント PC との接続 .....	6
2.2. インストール.....	6
3. コマンドリファレンス .....	7
3.1. メソッド・プロパティ .....	7
3.1.1. CaoWorkspace クラス .....	7
3.1.1.1. CaoWorkspace::AddController メソッド .....	7
3.1.2. CaoController クラス .....	8
3.1.2.1. CaoController::ExtensionNames プロパティ.....	8
3.1.2.2. CaoController::AddExtension メソッド .....	8
3.1.2.3. CaoController::VariableNames プロパティ .....	9
3.1.2.4. CaoController::AddVariable メソッド .....	9
3.1.3. CaoExtension クラス .....	10
3.1.3.1. CapExtension::get_ID プロパティ.....	10
3.1.3.2. CaoExtension::VariableNames プロパティ .....	10
3.1.3.3. CaoExtension::AddVariable メソッド .....	10
3.1.4. CaoVariable クラス .....	11
3.1.5. CaoVariable::get_Value プロパティ .....	11
3.2. 変数一覧 .....	12
3.2.1. CaoController クラス .....	12
3.2.1.1. プロバイダ情報関連変数 .....	12
3.2.1.1.1. @MAKER_NAME.....	12
3.2.1.1.2. @VERSION.....	12
3.2.2. CaoExtension クラス .....	13
3.3. エラーコード.....	26
4. サンプルプログラム .....	27

## 1. はじめに

本書は、FIELD system<sup>1</sup>の FIELD コンバータ<sup>2</sup>で収集したデータを ORiN から参照する機能を提供する、ORiN Gateway と通信を行う CAO プロバイダのユーザーズガイドです。以降本書で扱う CAO プロバイダを FIELD system プロバイダと呼びます。

ORiN Gateway は FIELD system の専用ハードウェアである FIELD system BOX 内で動作する FIELD アプリケーションの 1 つです。FIELD system BOX 内でサーバとして動作し、FIELD system プロバイダを通じた要求に応じて FIELD system API(REST) を実行し、最新値を取得します。取得したデータは CAO オブジェクトに変換して FIELD system プロバイダに返します。

図 1-1 に主要関連要素の構成を示します。

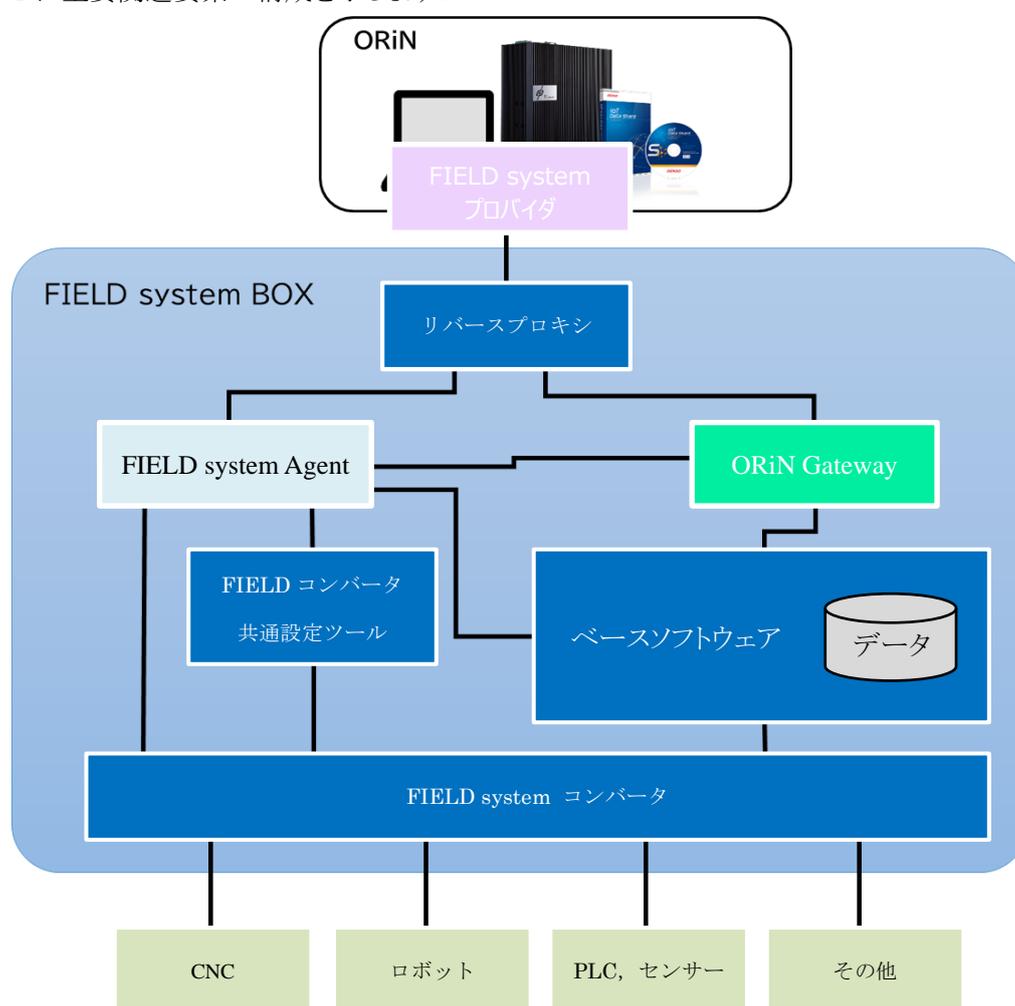


図 1-1 構成図

<sup>1</sup> FIELD system はファナック社の製造業向けオープンプラットフォームです。

<sup>2</sup> FIELD コンバータは FIELD system の構成要素の 1 つであり、生産機器からデータを収集する機能を持ちます。生産機器ごとに複数の FIELD コンバータが提供されています。

FIELD system では controller クラスを起点としたデータモデルで機器構成を表現します。図 1-2 にロボットを例に FIELD system のデータモデルと ORiN のデータモデルの対応を示します。ORiN では FIELD system の各データモデルクラスを CaoExtension(拡張ボード)、クラスの属性を CaoVariable(変数)として扱います。FIELD system では controller\_robot\_axis クラスなど 1 つの機器構成の中に複数持つことができるクラスがあり、FIELD system が採番するインスタンス ID で区別されています。ORiN では CaoExtension 生成時のオプションにインスタンス ID を指定することにより区別します。

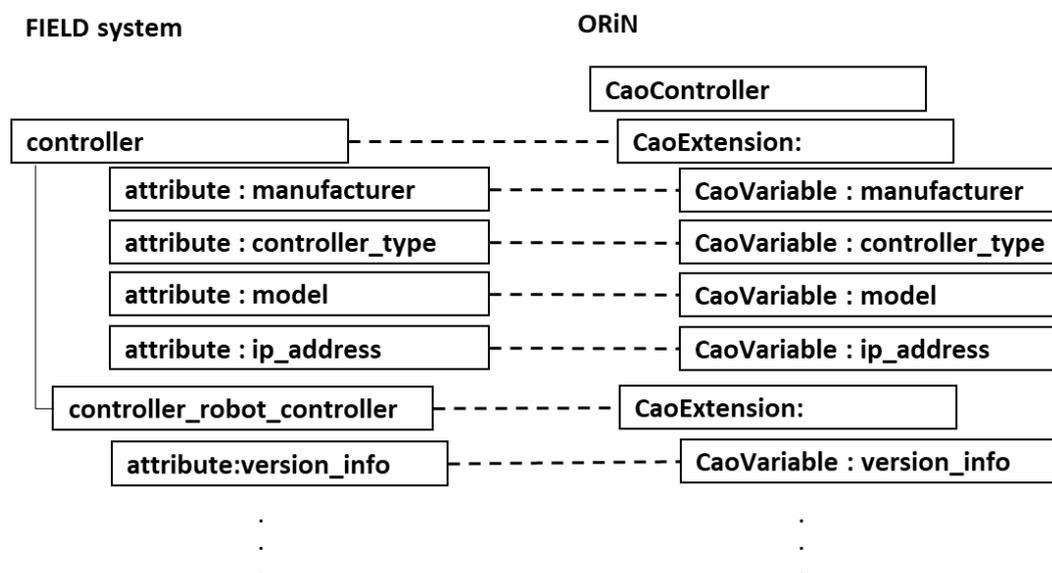


図 1-2 データモデルの対応

## 2. アプリケーション開発のための環境セットアップ

### 2.1. FIELD system Box とクライアント PC との接続

FIELD system BOX とクライアント PC は Ethernet で通信します。図 2-1 のようにハブを介して接続してください。

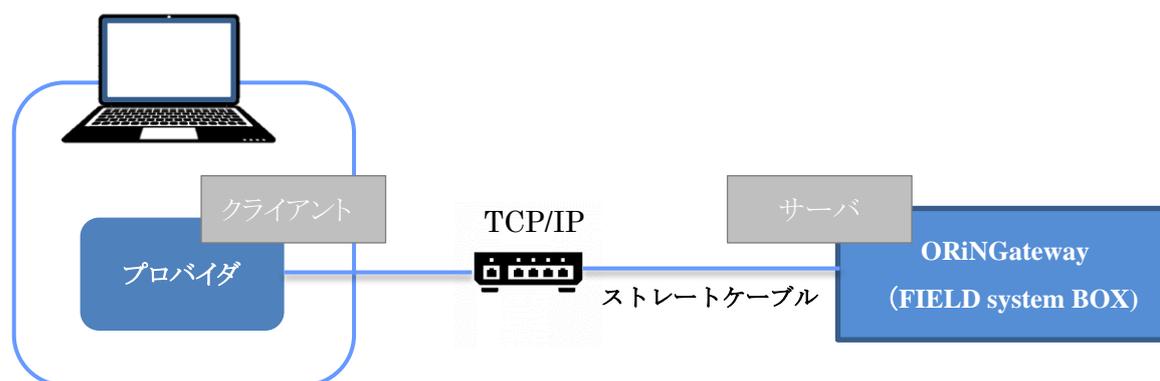


図 2-1 接続図

### 2.2. インストール

FIELD system プロバイダのファイル形式は DLL(Dynamic Link Library)であり、CAO エンジンから使用時に動的ロードされます。使用するにあたっては ORiN2SDK をインストールするか、表 2-1 を参照して手作業でレジストリ登録を行う必要があります。

表 2-1 FIELD system プロバイダ

ファイル名	CaoProvFIELDsystem.dll
ProgID	CaoProv.FANUC.FIELDsystem
レジストリ登録	regsvr32 CaoProvFIELDsystem.dll
レジストリ登録の抹消	regsvr32 /u CaoProvFIELDsystem.dll

## 3. コマンドリファレンス

### 3.1. メソッド・プロパティ

#### 3.1.1. CaoWorkspace クラス

##### 3.1.1.1. CaoWorkspace::AddController メソッド

FIELD system プロバイダは AddController メソッド実行時に渡されたパラメータを参照し、該当する FIELD system BOX と接続を行います。以下に、AddController メソッドの仕様を示します。

**書式** AddController ( <bstrCtrlName:VT\_BSTR>,<bstrProvName:VT\_BSTR>,  
<bstrPcName:VT\_BSTR>,[<bstrOption:VT\_BSTR>] )

<bstrCtrlName> : [in] コントローラ名  
 <bstrProvName> : [in] プロバイダ名. 固定値 ="CaoProv.FANUC.FIELDsystem"  
 <bstrPcName> : [in] プロバイダの実行マシン名 (未使用)  
 <bstrOption> : [in] オプション文字列

以下にオプション文字列に指定するリストを示します。

**表 3-1 CaoWorkspace::AddController のオプション文字列**

オプション	意味
Server=<IPアドレス>[:<ポート番号>]	必須. 接続先 FIELD system BOX の IP アドレスとポート番号を指定します. ポート番号を省略した場合, 40000 番に設定されます.
Controller=<controller <sup>3</sup> のインスタンス ID>	必須. FIELD system の接続先 controller クラスのインスタンス ID を指定します.
ConnTimeout[=<タイムアウト時間>]	接続時のタイムアウト時間をミリ秒で指定します. 設定値が 1000 ミリ秒以下の場合, 1000 ミリ秒に設定されます. (デフォルト:5000)
Timeout[=<タイムアウト時間>]	コマンド送受信時のタイムアウト時間をミリ秒で指定します. (デフォルト:500)

<sup>3</sup> controller クラスは FIELD コンバータが扱うデータモデルの起点に構成されるクラスです.

**使用例**

```
CaoEngine caoEng;  
CaoWorkspaces caoWss;  
CaoWorkspace caoWs;  
CaoControllers caoCtrls;  
CaoController caoCtrl;  
  
caoEng = new CaoEngine();  
caoWss = caoEng.Workspaces;  
caoWs = caoWss.Item(0);  
caoCtrls = caoWs.Controllers;  
  
// 接続  
caoCtrl = caoWs.AddController("Sample", "CaoProv.FANUC.FIELDsystem", null,  
                             "Server=192.168.0.10:40000,  
                             Controller=controller00001,ConnTimeout=5000,Timeout=5000");
```

### 3.1.2. CaoController クラス

#### 3.1.2.1. CaoController::ExtensionNames プロパティ

AddExtension メソッドで指定できる拡張ボード名の一覧を取得します。FIELD system プロバイダでは、FIELD system の controller 配下に紐づくデータモデルのインスタンス ID 一覧を表します。controller 配下に紐づくデータモデルは FIELD コンバータによって異なります。詳細は FIELD system の取扱説明書をご確認ください。

**使用例**

```
// 拡張ボード名一覧取得  
string[] names = caoCtrl.GetExtensionNames(null);
```

#### 3.1.2.2. CaoController::AddExtension メソッド

FIELD system プロバイダで FIELD system のデータを参照するには、FIELD system のデータモデルに対応する CaoExtension オブジェクトを追加する必要があります。

拡張ボード名には任意の文字列か CaoController::GetExtensionNames で取得したインスタンス ID を使用できます。任意の文字列を使用する場合は、オプションにインスタンス ID を指定してください。以下に、AddExtension メソッドの仕様を示します。

**書式**

```
AddExtension (<bstrExtensionName:VT_BSTR>,  
              [<bstrOption:VT_BSTR>])
```

<bstrCtrlName> : [in] 拡張ボード名

<bstrOption> : [in] オプション文字列

以下にオプション文字列に指定するオプションを示します。

表 3-2 CaoController::AddExtension のオプション文字列

オプション	意味
InstanceId=[<FIELD system のデータモデルのインスタンス ID>]	データ取得対象のデータモデルのインスタンス ID を指定します。拡張ボード名に任意の文字列を指定した場合、インスタンス ID は必ず指定してください。 インスタンス ID は【データモデル名】+5桁の数字です。 例: controller_robot_controller00001

#### 使用例

```
// 拡張ボードの追加
// 任意の拡張ボード名を使用する場合、InstanceIdオプションを指定する
CCaoExtension caoExt = caoCtrl.AddExtension("Extension", "InstanceId=controller00001");

// CaoController::GetExtensionNamesで取得した名称を使用
var names = caoCtrl.GetExtensionNames(null);
CCaoExtension caoExt2 = caoCtrl.AddExtension(names[0], null);
```

### 3.1.2.3. CaoController::VariableNames プロパティ

表 3-3 に示すシステム変数名の一覧を取得します。

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

CaoController に変数オブジェクトを追加します。変数名は、表 3-3 に示す変数名のみ使用できます。

#### 書式

AddVariable (<bstrVariableName:VT\_BSTR>, [<bstrOption:VT\_BSTR>])

<bstrVariableName> : [in] 変数名  
<bstrOption> : [in] オプション文字列

#### 使用例

```
// バージョン取得用の変数を追加
CCaoVariable version = caoCtrl.AddVariable("@VERSION", null);
```

### 3.1.3. CaoExtension クラス

#### 3.1.3.1. CaoExtension::get\_ID プロパティ

CaoExtension の ID を取得します。FIELD system プロバイダでは、CaoExtension が対応する FIELD system のデータモデルのインスタンス ID を表します。

##### 使用例

---

```
// 拡張ボードの追加
CCaoExtension caoExt = caoCtrl.AddExtension("Extension", "InstanceId=controller00001");
// 拡張ボードのID取得
var id = caoExt.ID;
```

---

#### 3.1.3.2. CaoExtension::VariableNames プロパティ

変数名の一覧を取得します。FIELD system プロバイダでは、CaoExtension が対応する FIELD system のデータモデルのアトリビュート名の一覧を表します。

##### 使用例

---

```
// 拡張ボードの追加
CCaoExtension caoExt = caoCtrl.AddExtension("Extension", "InstanceId=controller00001");
// 変数名一覧取得
var attrNames = caoExt.GetVariableNames(null);
```

---

#### 3.1.3.3. CaoExtension::AddVariable メソッド

CaoExtension オブジェクトに変数オブジェクトを追加します。変数名には VariableNames プロパティで取得できる変数名のみ使用できます。

##### 書式

AddVariable (<bstrVariableName:VT\_BSTR>, [<bstrOption:VT\_BSTR>])

<bstrVariableName> : [in] 変数名  
<bstrOption> : [in] オプション文字列

##### 使用例

---

```
// 拡張ボードの追加
CCaoExtension caoExt = caoCtrl.AddExtension("Extension", "InstanceId=controller00001");
// 変数名一覧取得
var attrNames = caoExt.GetVariableNames(null);
// 変数を追加
CCaoVariable val = caoCtrl.AddVariable(attrNames[0], null);
```

---

### 3.1.4. CaoVariable クラス

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

CaoController オブジェクトのシステム変数の場合はプロバイダ情報, CaoExtension オブジェクトの変数の場合は FIELD system のアトリビュートの最新値を取得します。

#### 使用例

---

```
// 拡張ボードの追加
CCaoExtension caoExt = caoCtrl.AddExtension("Extension", "InstanceId=controller00001");
// 変数名一覧取得
var attrNames = caoExt.GetVariableNames(null);
// 変数を追加
CCaoVariable val = caoCtrl.AddVariable(attrNames[0], null);
// 最新値取得
var value = val.Value;
```

---

## 3.2. 変数一覧

### 3.2.1. CaoController クラス

表 3-3 CaoController クラス システム変数一覧

変数名	説明	属性		ページ
		get	put	
@MAKER_NAME	メーカー名を取得します.	○	-	12
@VERSION	プロバイダバージョンを取得します.	○	-	12

#### 3.2.1.1. プロバイダ情報関連変数

FIELD system プロバイダが保持している情報を取得する変数です.

##### 3.2.1.1.1. @MAKER\_NAME

メーカー名を取得します.

データ型	内容
VT_BSTR	FANUC

##### 3.2.1.1.2. @VERSION

プロバイダバージョンを取得します.

データ型	内容
VT_BSTR	*.*.*

## 3.2.2. CaoExtension クラス

CaoExtension オブジェクトが対応するFIELD system のデータモデルによって使用できる変数が異なります。以下に、FIELD system 第3版 Application SDK 時点のデータモデルと変数の一覧を示します（変数を持たないデータモデルは除外しています）。

表 3-4 CaoExtension クラス 変数一覧

変数名	説明	データ型	属性	
			get	put
controller	コントローラの情報を表すクラス			
name	コントローラの任意名称	VT_BSTR	○	-
manufacturer	コントローラの製造メーカー	VT_BSTR	○	-
controller_type	コントローラの種別	VT_BSTR	○	-
model	機種名	VT_BSTR	○	-
ip_address	IP アドレス (コンバータの接続先)	VT_BSTR	○	-
controller_cnc	CNC 特有の情報を表すクラス			
software_name	CNC ソフトウェア名	VT_BSTR	○	-
software_version	CNC ソフトウェアのバージョン	VT_BSTR	○	-
total_paths	CNC の系統数	VT_I4	○	-
total_axes	軸数	VT_I4	○	-
total_servos	CNC のサーボ軸数	VT_I4	○	-
total_spindles	CNC の主軸数	VT_I4	○	-
plc_software_name	PLC(PMC)ソフトウェア名	VT_BSTR	○	-
plc_software_version	PLC(PMC)ソフトウェアのバージョン	VT_BSTR	○	-
total_plc_paths	PLC(PMC)の系統数	VT_I4	○	-
boot_software_name	BOOT ソフトウェア名	VT_BSTR	○	-
boot_software_version	BOOT ソフトウェアのバージョン	VT_BSTR	○	-
hardware_info	ハードウェア情報	VT_ARRAY  VT_VARIANT	○	-
	種類	0   VT_BSTR		
	ID	1   VT_BSTR		

変数名	説明	データ型	属性
-----	----	------	----

			get	put
controller_cnc_axis		CNC の軸情報を表すクラス		
axis_name	軸名称	VT_BSTR	○	-
axis_type	軸タイプ(サーボ軸(直線 linear/ 回転 rotation) 主軸 spindle)	VT_BSTR	○	-
axis_number	絶対軸(主軸)番号	VT_I4	○	-
path_number	軸(主軸)の所属する系統番号	VT_I4	○	-
path_axis_number	系統内軸番号	VT_I4	○	-
controller_cnc_path		CNC の系統の情報を表すクラス		
name	系統名称	VT_BSTR	○	-
path_number	自系統番号	VT_I4	○	-
total_axes	系統内軸数	VT_I4	○	-
path_type	制御タイプ	VT_BSTR	○	-
controller_laser		レーザ制御装置の情報を表すクラス		
software_name	ソフトウェア名	VT_BSTR	○	-
software_version	ソフトウェアのバージョン	VT_BSTR	○	-
total_axes	軸数	VT_I4	○	-
plc_software_name	PLC(PMC)ソフトウェア名	VT_BSTR	○	-
plc_software_version	PLC(PMC)ソフトウェアのバージョン	VT_BSTR	○	-
total_plc_paths	PLC(PMC)の系統数	VT_I4	○	-
boot_software_name	BOOT ソフトウェア名	VT_BSTR	○	-
boot_software_version	BOOT ソフトウェアのバージョン	VT_BSTR	○	-
hardware_info	ハードウェア情報	VT_ARRAY  VT_VARIANT	○	-
	種類	0   VT_BSTR		
	ID	1   VT_BSTR		
controller_laser_oscillator		レーザ発振器の情報を表すクラス		
name	レーザ発振器の名称	VT_BSTR	○	-
path_number	レーザ発振器の所属する系統番号	VT_I4	○	-
oscillator_number	レーザ発振器番号	VT_I4	○	-
変数名	説明	データ型	属性	

			get	put
controller_robot_controller		ロボットコントローラの情報を表すクラス		
version_info	ソフトウェアバージョンのオブジェクト	VT_ARRAY  VT_VARIANT	○	-
	ソフトウェアの種類	0   VT_BSTR		
	バージョン	1   VT_BSTR		
total_groups	グループ数の合計 (コントローラの制御単位の合計)	VT_I4	○	-
application_name	アプリケーション名	VT_BSTR	○	-
controller_robot_group		ロボットコントローラの制御系統を表すクラス		
total_axes	軸数	VT_I4	○	-
group_name	グループ名	VT_BSTR	○	-
group_number	グループ番号	VT_I4	○	-
mechanical_name	機種	VT_BSTR	○	-
total_robot_axes	ロボット軸数	VT_I4	○	-
total_external_axes	付加軸数	VT_I4	○	-
controller_robot_axis		ロボットコントローラで制御される軸を表すクラス		
max_amp_current	最大アンプ電流	VT_R8	○	-
axis_name	軸名	VT_BSTR	○	-
axis_number	軸番号	VT_I4	○	-
axis_type	軸タイプ (直線 linear/回転 rotation)	VT_BSTR	○	-
controller_injection		射出成形機のコントローラの情報を表すクラス		
name	射出成形機の機械名	VT_BSTR	○	-
controller_wirecut		ワイヤカット放電加工機の情報を表すクラス		
name	ワイヤカット放電加工機の機械名	VT_BSTR	○	-
controller_sensor_individual		個別のセンサの情報を表すクラス		
name	センサ名称	VT_BSTR	○	-
manufacturer	製造メーカー	VT_BSTR	○	-
model	型式 型番	VT_BSTR	○	-
description	任意のコメント	VT_BSTR	○	-
kind	センサ種別	VT_BSTR	○	-
installation	設置部位	VT_BSTR	○	-
usage	用途	VT_BSTR	○	-
変数名	説明	データ型	属性	

			get	put		
status	コントローラ間で共通の状態を表すクラス					
controller_status_number	コントローラ状態番号	VT_I4	○	-		
program_name	メインプログラム名(第1系統)	VT_BSTR	○	-		
act_program_name	実行中プログラム名(第1系統)	VT_BSTR	○	-		
alarm_flag	アラームフラグ	VT_I4	○	-		
alarm_info	アラーム情報	VT_ARRAY  VT_VARIANT	○	-		
		アラーム種別			0	VT_BSTR
		アラーム番号			1	VT_BSTR
		アラームメッセージ			2	VT_BSTR
lot_number	ロット番号	VT_BSTR	○	-		
product_amount	生産量	VT_I4	○	-		
good_product_amount	良品生産量	VT_I4	○	-		
ng_product_amount	不良品生産量	VT_I4	○	-		
act_cycle_time	サイクルタイム(実績)	VT_R8	○	-		
total_power_on_time	積算電源 ON 時間(コントローラ全体)	VT_R8	○	-		
total_operation_time	積算稼働時間(コントローラ全体)	VT_R8	○	-		
act_power_consumption	実消費電力(コントローラ全体)	VT_I4	○	-		
product_amount_per_cycle	1 サイクルあたりの生産量	VT_I4	○	-		
status_cnc	CNC 共通の状態を表すクラス					
fan_speed	ファンの回転数	VT_I4	○	-		
fan_status	ファンの交換情報	VT_BSTR	○	-		
plc_status	PLC(PMC)のステータス	VT_BSTR	○	-		
status_cnc_path	CNC の系統の状態を表すクラス					
mode	動作モード	VT_BSTR	○	-		
status	動作状態	VT_BSTR	○	-		
program_name	メインプログラム名	VT_BSTR	○	-		
act_program_name	実行中プログラム名	VT_BSTR	○	-		
program_comment	メインプログラムのコメント	VT_BSTR	○	-		
act_program_comment	実行中プログラムのコメント	VT_BSTR	○	-		
sequence_number	シーケンス番号	VT_BSTR	○	-		
変数名	説明	データ型	属性			

			get	put
act_rotation_speed	回転数(実測値)	VT_R8	○	-
feedrate	送り速度(実測値)	VT_R8	○	-
act_f_code	指令されている F コード	VT_BSTR	○	-
act_m_code	指令されている M コード	VT_BSTR	○	-
act_s_code	指令されている S コード	VT_BSTR	○	-
act_t_code	指令されている T コード	VT_BSTR	○	-
act_b_code	指令されている B コード	VT_BSTR	○	-
product_amount	実行中のプログラムで生産した 部品の総数	VT_R8	○	-
total_product_amount	これまでに生産した部品の総数	VT_R8	○	-
operation_time	プログラムの運転時間	VT_R8	○	-
cutting_time	プログラムの切削時間	VT_R8	○	-
dryrun_signal	ドライラン信号状態	VT_I4	○	-
machine_lock_signal	マシンロック(全軸)信号状態	VT_I4	○	-
work_coordinate_number	ワーク座標系選択番号	VT_I4	○	-
cut_override	切削送りオーバーライド	VT_I4	○	-
rapid_override	早送りオーバーライド	VT_I4	○	-
status_cnc_axis	CNC で操作しているの軸の状態を表すクラス			
axis_load	軸の負荷	VT_I4	○	-
rotation_override	回転速度オーバーライド	VT_I4	○	-
temperature	モータ温度	VT_I4	○	-
absolute_position	絶対座標	VT_R8	○	-
relative_position	相対座標	VT_R8	○	-
machine_position	機械座標	VT_R8	○	-
rotation_speed	回転速度	VT_R8	○	-
insulation_resistance_data	絶縁抵抗値	VT_R8	○	-
leakage_detection_number	絶縁抵抗劣化状態	VT_I4	○	-
status_laser_oscillator	レーザ発振器の状態を表すクラス			
power_peek	レーザ出力(ピーク)指令値	VT_I4	○	-
power_bottom	レーザ出力(ボトム)指令値	VT_I4	○	-
frequency	レーザ出力パルス周波数指令値	VT_I4	○	-
duty	レーザ出力デューティ指令値	VT_I4	○	-
変数名	説明	データ型	属性	

			get	put
sequence	レーザー発振器の起動状態	VT_BSTR	○	-
beam_on	ビームオン状態	VT_BSTR	○	-
output_monitor	レーザー実出力強度	VT_I4	○	-
water_flow	冷却水流量	VT_R8	○	-
run_on_time	レーザー発振器の積算電源ON時間	VT_R8	○	-
operation_time	レーザー発振器指令時間	VT_R8	○	-
cabinet_temperature	庫内温度	VT_R8	○	-
cabinet_humidity	庫内湿度	VT_R8	○	-
cabinet_water_temperature	冷却水温度	VT_R8	○	-
dewpoint	露点	VT_R8	○	-
discharge_tube_pressure	放電管内圧力	VT_R8	○	-
power_consumption	レーザー発振器の消費電力	VT_R8	○	-
total_power_consumption	レーザー発振器の積算消費電力	VT_R8	○	-
gas_supply_pressure	レーザーガス供給圧	VT_R8	○	-
calibration_coefficient	パワー補正係数	VT_I4	○	-
laser_cavity_unit_output	レーザー共振器ユニット出力値	VT_I4	○	-
fast_fiber_break_detector	FFBD(高速ファイバ断線検知)	VT_I4	○	-
cutting_index	加工指数	VT_I4	○	-
exhaust_pump	排気ポンプ稼働時間	VT_R8	○	-
power_supply	電源	VT_ARRAY  VT_VARIANT	○	-
	駆動電源番号	0   VT_I4		
	駆動電源の情報	1   VT_BSTR		
	チャンネル	2   VT_I4		
	電圧値	3   VT_R8		
	電流値	4   VT_R8		
	電力値	5   VT_R8		
	ヒートシンク温度	6   VT_R8		
status_plc	PLC の状態を表すクラス			
status	稼働状態	VT_BSTR	○	-

変数名	説明	データ型	属性
-----	----	------	----

			get	put
status_sensor_individual		個別のセンサの状態を表すクラス		
status	センサの動作状態	VT_BSTR	○	-
data	取得した生値と単位	VT_ARRAY  VT_VARIANT	○	-
	センサの生値	0   VT_BSTR		
	センサの単位(コモンコード)	1   VT_BSTR		
status_sensor_individual_bulk_info		センサのバルクデータの状態を表すクラス		
type	種類	VT_BSTR	○	-
start_address	アドレス	VT_BSTR	○	-
length	データ数	VT_I4	○	-
data_type	データの型	VT_BSTR	○	-
status_robot_controller		ロボットコントローラの状態を表すクラス		
fault	アラームのフラグ	VT_I4	○	-
running	稼働中のフラグ	VT_I4	○	-
paused	停止中のフラグ	VT_I4	○	-
servo_ready	サーボレディのフラグ	VT_I4	○	-
tp_enable	TP スイッチのフラグ	VT_I4	○	-
battery_alarm	バッテリーアラームのフラグ	VT_I4	○	-
estop_operator_panel	非常停止パネルのフラグ	VT_I4	○	-
estop_teach_pendant	非常停止 TP のフラグ	VT_I4	○	-
estop_external	外部非常停止のフラグ	VT_I4	○	-
fence_open	フェンスオープンフラグ	VT_I4	○	-
enabling_device	デッドマンスイッチのフラグ	VT_I4	○	-
status_robot_controller_variable		ロボットコントローラの変数の状態を表すクラス		
type	種類	VT_BSTR	○	-
name	名前	VT_BSTR	○	-
value	値	VT_BSTR	○	-
unit	単位	VT_BSTR	○	-
status_robot_controller_task		ロボットコントローラで動作しているタスクの状態を表すクラス		
task	ロボットのタスク	VT_ARRAY  VT_VARIANT	○	-
	タスク名	0   VT_BSTR		

変数名	説明		データ型		属性	
					get	put
		プログラム名	1	VT_BSTR		
		行番号	2	VT_I4		
		状態	3	VT_BSTR		
status_robot_group	ロボットコントローラの制御系統の情報を表すクラス					
is_running	対象グループ稼働フラグ		VT_I4		○	-
is_servo_ready	対象グループサーボレディフラグ		VT_I4		○	-
user_frame_id	ユーザ座標系番号		VT_I4		○	-
tool_frame_id	ツール座標系番号		VT_I4		○	-
joint_position	各軸座標のオブジェクト		VT_ARRAY   VT_VARIANT		○	-
	各軸座標値	0	VT_ARRAY   VT_VARIANT			
	単位	1	VT_ARRAY   VT_VARIANT			
cartesian_position	直交座標のオブジェクト		VT_ARRAY   VT_VARIANT		○	-
	タイプ	0	VT_BSTR			
	直行座標値	1	VT_ARRAY   VT_VARIANT			
	単位	2	VT_ARRAY   VT_VARIANT			
	形態	3	VT_BSTR			
status_robot_axis	ロボットコントローラで制御される軸の状態を表すクラス					
speed	速度		VT_R8		○	-
acceleration	加速度		VT_R8		○	-
torque	トルク		VT_R8		○	-
q_current	Q 相電流		VT_R8		○	-
dis_torque	外乱値		VT_R8		○	-
axis_position	軸位置		VT_R8		○	-
current	電流		VT_R8		○	-
ovc_value	ovc 値		VT_R8		○	-
pulse	パルス		VT_R8		○	-

変数名	説明	データ型	属性	
			get	put
status_injection	射出成形機のコントローラの状態を表すクラス			
injection_controller_status_number	射出成形機コントローラの状態	VT_BSTR	○	-
status_wirecut	ワイヤカット放電加工機のコントローラの状態を表すクラス			
controller_status_number	コントローラ状態番号	VT_I4	○	-
program_name	メインプログラム名(第1系統)	VT_BSTR	○	-
act_program_name	実行中プログラム名(第1系統)	VT_BSTR	○	-
alarm_flag	アラームフラグ	VT_I4	○	-
alarm_info	アラーム情報	VT_ARRAY  VT_VARIANT	○	-
	アラーム種別	0   VT_BSTR		
	アラーム番号	1   VT_BSTR		
	アラームメッセージ	2   VT_BSTR		
total_power_on_time	積算電源 ON 時間(コントローラ全体)	VT_R8	○	-
total_operation_time	積算稼働時間(コントローラ全体)	VT_R8	○	-
act_power_consumption	積算消費電力量(コントローラ全体)	VT_I4	○	-
mode	動作モード	VT_BSTR	○	-
sequence_number	シーケンス番号	VT_BSTR	○	-
feedrate	送り速度(実測値)	VT_R8	○	-
absolute_position	絶対座標	VT_R8	○	-
relative_position	相対座標	VT_R8	○	-
machine_position	機械座標	VT_R8	○	-
total_length	現在のプログラムでの全加工距離	VT_R8	○	-
act_length	現在のプログラムでの加工完了距離	VT_R8	○	-
wire_break_flag	ワイヤ断線中フラグ	VT_I4	○	-
act_cutting_times	現在の加工回数	VT_I4	○	-
awf_flag	自動結線実行中フラグ	VT_I4	○	-
cut_wire_flag	ワイヤ切断実行中フラグ	VT_I4	○	-
consume_warning_flag	消耗品警告フラグ	VT_I4	○	-
peri_maintenance_warning_flag	定期保守警告フラグ	VT_I4	○	-

変数名	説明	データ型	属性	
			get	put
quality_maintenance_file_info	品質情報ファイルの情報を表すクラス			
name	ファイル名	VT_BSTR	○	-
type	ファイルの種類	VT_BSTR	○	-
quality_maintenance_robot_controller	ロボットコントローラの品質情報を表すクラス			
total_power_on_time	積算電源 ON 時間	VT_R8	○	-
total_servo_on_time	積算サーボオン時間	VT_R8	○	-
total_running_time	積算稼働時間	VT_R8	○	-
total_waiting_time	積算待機時間	VT_R8	○	-
quality_maintenance_injection	射出成形機の成形サイクル毎のコントローラの状態を表すクラス			
status	機械状態	VT_BSTR	○	-
cycle_time	型閉じ開始から型閉じ開始までの時間. 1 成形毎の時間	VT_R8	○	-
injection_time	射出開始から終了までの時間	VT_R8	○	-
extruder_time	計量開始から終了までの時間	VT_R8	○	-
minimum_cushion	射出・保圧中のスクリュ最前進位置	VT_R8	○	-
extruder_position	計量完了時のスクリュ位置	VT_R8	○	-
peak_injection_pressure	射出・保圧中の最大射出圧	VT_R8	○	-
shot_number	サイクル数	VT_I4	○	-
good_shot	成形機で良品と判断したサイクル数	VT_I4	○	-
bad_shot	成形機で不良品と判断したサイクル数	VT_I4	○	-
remain_shot	成形機に設定されている生産予定数に到達するまでのサイクル数	VT_I4	○	-
alarm_code	アラームコード	VT_BSTR	○	-
vp_transfer_position	射出工程から保圧工程に切替えた時のスクリュ位置	VT_R8	○	-
temp_mold1	1 サイクル中のある時点での金型 1 の温度	VT_R8	○	-

変数名	説明	データ型	属性	
			get	put
temp_mold2	1サイクル中のある時点での金型2の温度	VT_R8	○	-
temp_nozzle1	1サイクル中のある時点でのノズル1の温度	VT_R8	○	-
temp_nozzle2	1サイクル中のある時点でのノズル2の温度	VT_R8	○	-
temp_barrel1	1サイクル中のある時点でのバレル1の温度	VT_R8	○	-
temp_barrel2	1サイクル中のある時点でのバレル2温度	VT_R8	○	-
temp_barrel3	1サイクル中のある時点でのバレル3の温度	VT_R8	○	-
temp_barrel4	1サイクル中のある時点でのバレル4温度	VT_R8	○	-
temp_nozzle_adapter	1サイクル中のある時点でのノズルアダプタ温度の温度	VT_R8	○	-
temp_hopper	1サイクル中のある時点でのホッパー下の温度	VT_R8	○	-
extruder_start_position	計量開始時のスクリュ位置	VT_R8	○	-
extruder_torque1	計量トルク 1:計量行程中の任意区間における平均トルク	VT_R8	○	-
extruder_torque2	計量トルク 2:計量行程中の任意区間における平均トルク	VT_R8	○	-
hispeed_monitor1	高速モニタ 1:射出・保圧工程中の任意の点での射出圧力	VT_R8	○	-
extruder_torque3	計量トルク 3:計量行程中の任意区間における平均トルク	VT_R8	○	-
extruder_torque4	計量トルク 4:計量行程中の任意区間における平均トルク	VT_R8	○	-
hispeed_monitor2	高速モニタ 2:射出・保圧工程中の任意の点での射出圧力	VT_R8	○	-
extruder_torque5	計量トルク 5:計量行程中の任意区間における平均トルク	VT_R8	○	-

変数名	説明	データ型	属性	
			get	put
extruder_torque6	計量トルク 6:計量行程中の任意 区間における平均トルク	VT_R8	○	-
hispeed_monitor3	高速モニタ 3:射出・保圧工程中 の任意の点での射出圧力	VT_R8	○	-
temp_mold3	1サイクル中のある時点での金型 3 の温度	VT_R8	○	-
temp_mold4	1サイクル中のある時点での金型 4 の温度	VT_R8	○	-
hispeed_monitor4	高速モニタ 4:射出・保圧工程中 の任意の点での射出圧力	VT_R8	○	-
hispeed_monitor5	高速モニタ 5:射出・保圧工程中 の任意の点での射出圧力	VT_R8	○	-
consumption_power	消費電力	VT_R8	○	-
judge_flag_promoni	成形機に設定された上下限値を 超えた モニタ項目	VT_BSTR	○	-
judge_flag_hisp_5p	成形機に設定された上下限値を 超えた モニタ項目(高速モニタ 1 から 5)	VT_BSTR	○	-
judge_flag_ext_torq6p	成形機に設定された上下限値を 超えた モニタ項目(計量トルク 1~6)	VT_BSTR	○	-
injection_start_press	射出開始位置時の圧力	VT_R8	○	-
precise_extruder3_comp_quantity3	精密計量 3 の圧縮量	VT_R8	○	-
precise_extruder3_comp_press3	精密計量 3 の圧縮圧	VT_R8	○	-
precise_extruder3_v_p_correction	精密計量 3 の V-P 位置の補正位 置	VT_R8	○	-
backflow_peak	バックフローのピーク	VT_R8	○	-
backflow_average	バックフローの平均値	VT_R8	○	-
event_alarm	アラームイベントを表すクラス			
type	種別(アスキー文字列)	VT_BSTR	○	-
number	アラーム番号(アスキー文字列)	VT_BSTR	○	-
severity	重度(アスキー文字列)	VT_BSTR	○	-

変数名	説明	データ型	属性			
			get	put		
message	メッセージ(utf-8)	VT_BSTR	○	-		
comment	コメント(utf-8)	VT_BSTR	○	-		
date	アラーム発生日時	VT_BSTR	○	-		
event_variable_info	イベント発生時の変数の情報を表すクラス					
dataset_id	データセット ID	VT_BSTR	○	-		
dataset_name	データセット名	VT_BSTR	○	-		
variable_info	変数情報	VT_ARRAY  VT_VARIANT	○	-		
		変数の種類			0	VT_BSTR
		変数の型			1	VT_BSTR
		変数のアドレス			2	VT_BSTR
		変数名			3	VT_BSTR
		変数の単位(コモンコード)			4	VT_BSTR
tag	付帯情報	VT_ARRAY  VT_VARIANT	○	-		
		キー			0	VT_BSTR
		値			1	VT_BSTR
event_variable	イベント発生時の変数の値を表すクラス					
value	変数の値	VT_BSTR	○	-		
tag	付帯情報	VT_ARRAY  VT_VARIANT	○	-		
		キー			0	VT_BSTR
		値			1	VT_BSTR
archive_maintenance_compress_file_info	保守用バックアップファイルの情報を表すクラス					
name	ファイル名	VT_BSTR	○	-		
type	ファイル形式	VT_BSTR	○	-		

### 3.3. エラーコード

FIELD system プロバイダは、以下の固有エラーを返します。

エラー名	エラー番号	説明
JSON 解析エラー	0x801103E9	FIELD system API 実行結果の JSON 解析でエラーが発生した場合に返します。 エラー原因は以下が考えられます。 <ul style="list-style-type: none"><li>•API リクエストの送受信に失敗している</li><li>•FIELD コンバータによる測定値が 1 件も存在しない</li></ul> FIELD system の管理画面から ORiNGateway のログや FIELD コンバータのデータ収集状況を確認してください。
接続先 controller 指定エラー	0x801104B0	接続先に指定された FIELD system の controller が見つからない場合に返します。
インスタンス ID フォーマットエラー	0x801104B1	CaoExtension オブジェクト生成時に指定したインスタンス ID のフォーマットが正しくない場合に返します。

## 4. サンプルプログラム

以下に、本プロバイダを使用して FIELD system と通信を行い、測定値を取得するサンプルプログラム(C#)を示します。

### Sample

### Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using ORiN2.ManagedCAO;
using System.Threading;

namespace Sample
{
    class Program
    {
        static void Main(string[] args)
        {
            CCaoEngine caoEng = null;

            try
            {
                // Create Cao Objects.
                caoEng = new CCaoEngine();
                CCaoWorkspaces caoWss = caoEng.Workspaces;
                CCaoWorkspace caoWs = caoWss[0];
                CCaoControllers caoCtrls = caoWs.Controllers;

                var connOpt = "Server=192.168.56.101:40000,Controller=controller00015,
                               ConnTimeout=5000,Timeout=5000";

                // Connect to FIELD system
                CCaoController caoCtrl = caoWs.AddController("FIELDsystem_Sample",
                                                            "CaoProv.FANUC.FIELDsystem",
                                                            null,
                                                            connOpt);

                // Get CaoExtension names.
                var extNames = caoCtrl.GetExtensionNames(null);
                // Add CaoExtensions.
                foreach (var name in extNames)
                {
                    caoCtrl.AddExtension(name, null);
                }

                foreach (var ext in caoCtrl.Extensions)
                {
                    // Get variable names.
                    var names = ext.GetVariableNames(null);
                    // Add CaoVariables.
                    foreach (var name in names)
                    {
                        ext.AddVariable(name, null);
                    }
                }

                Thread.Sleep(1000);
            }
        }
    }
}
```

```
        Console.WriteLine("CaoExtension: " + ext.ID);
        Console.WriteLine();

        // Prints the value of each variable to the console
        foreach (var variable in ext.Variables)
        {
            try
            {
                var value = variable.Value;
                PrintValue(variable.Name, value);
            }
            catch (Exception)
            {
                Console.WriteLine(
                    string.Format("{0}: Failed to get value.", variable.Name));
            }
        }
    }
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
finally
{
    if (caoEng != null)
    {
        caoEng.Dispose();
        caoEng = null;
    }
}

Console.ReadKey();
}

/// <summary>
/// Display variable value
/// </summary>
/// <param name="name">variable name</param>
/// <param name="obj">value</param>
static void PrintValue(string name, object obj)
{
    if (obj == null)
    {
        Console.WriteLine(" null");
        return;
    }

    if (obj is Array)
    {
        var arr = (Array)obj;
        for (int i = 0; i < arr.Length; i++)
        {
            PrintValue(string.Format("{0} [{1}]", name, i), arr.GetValue(i));
        }
    }
    else
    {
        Console.WriteLine(string.Format("{0}: {1}", name, obj.ToString()));
    }
}
```

```
}  
  }  
}
```

