FIELD system provider

Version 1.0.0

User's Guide

July 10, 2020

Remark:

[Revision History]

Version	Date	Description
1.0.0	2020-07-10	First edition

[Operation check device]

Model	Notes
FIELD system Box	

Contents

1. Introduction

This manual is a user's guide for CAO providers communicating with ORiNGateway that provides the ability to view data collected by FIELD system's FIELD converters from ORiN. The CAO-providers covered in this manual are referred to as FIELD system provider.[1][2]

ORiNGateway is a FIELD system dedicated hardware. It acts as a server within FIELD system BOX and runs FIELD applications (FIELD system API(REST in response to requests through FIELD system provider) to retrieve the most recent data. The retrieved data is converted to CAO-objects and returned to FIELD system provider.

Fig. 1-1 shows Configuration Diagram.

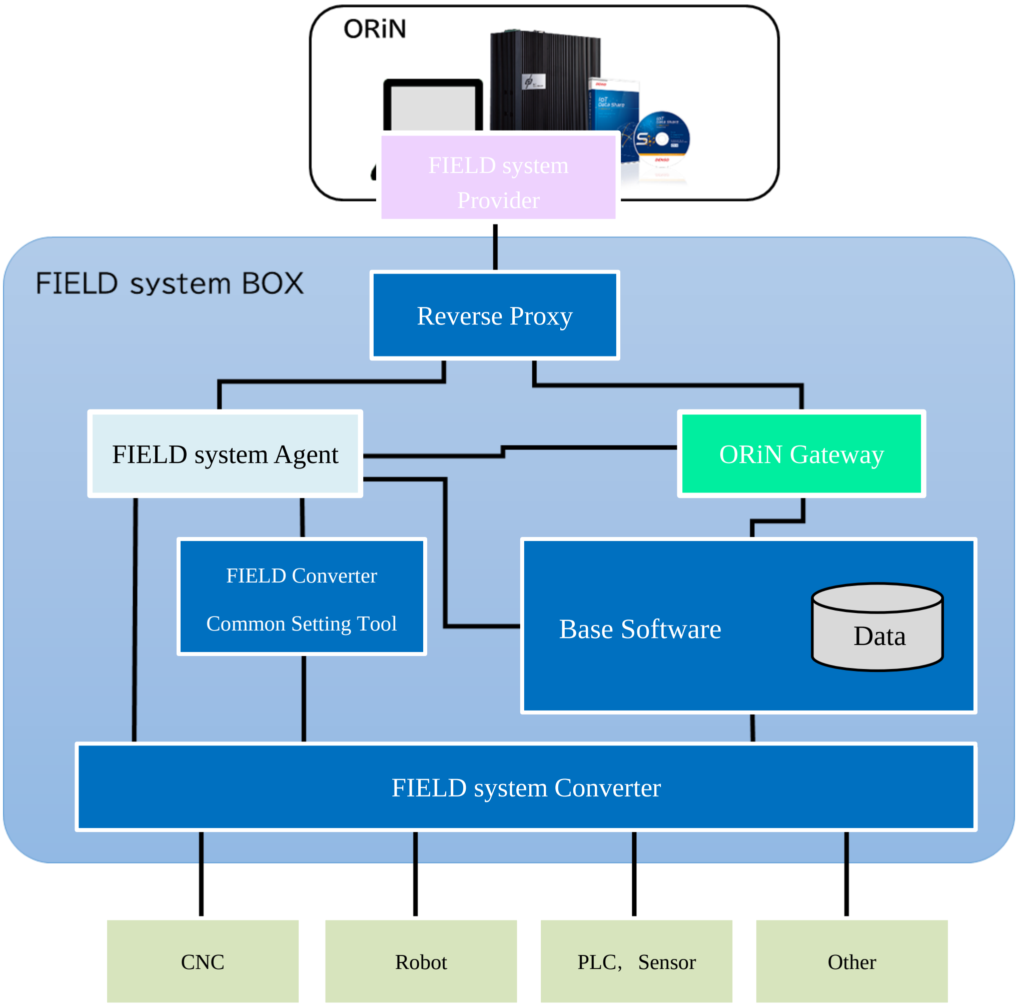

Fig. 1-1 Configuration Diagram

[1] FIELD system is an open-platform for Fanuc manufacturers.

[2] FIELD converters are a component of FIELD system and are capable of collecting data from manufacturing equipment. Several FIELD converters are provided for each manufacturing equipment.

In FIELD system, the device configuration is represented by the data model originating from controller class. Fig. 1-2 shows the correspondence between the data model of FIELD system and the data model of ORiN using the robot as an example. ORiN treats each data model class of FIELD system as a CaoExtension (extension board) and the attributes of the class as a CaoVariable (variable). FIELD system has a class which can have more than one in one device configuration such as controller_robot_axis class. It is distinguished by the instance ID which FIELD system numbers. In ORiN, it is distinguished by specifying the instance ID as an option at the time of CaoExtension generation.

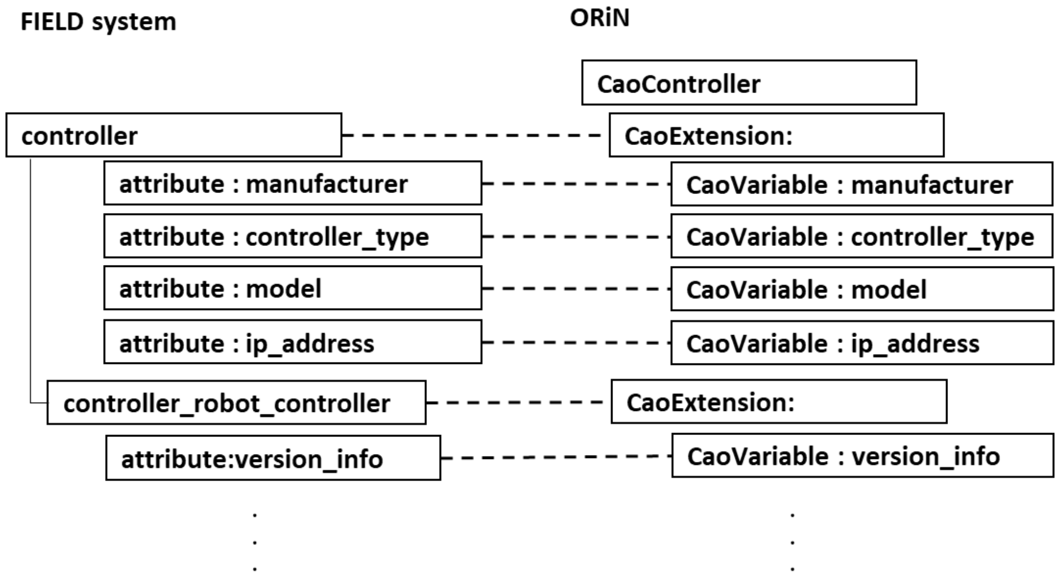

Fig. 1-2 Correspondence between data models

2. Setting Up Your Environment for Application Development

2.1. Connecting FIELD system Box to Client-PC

FIELD system BOX and the client-PC communicate on Ethernet. Connect via a hub as shown in Fig. 2-1.

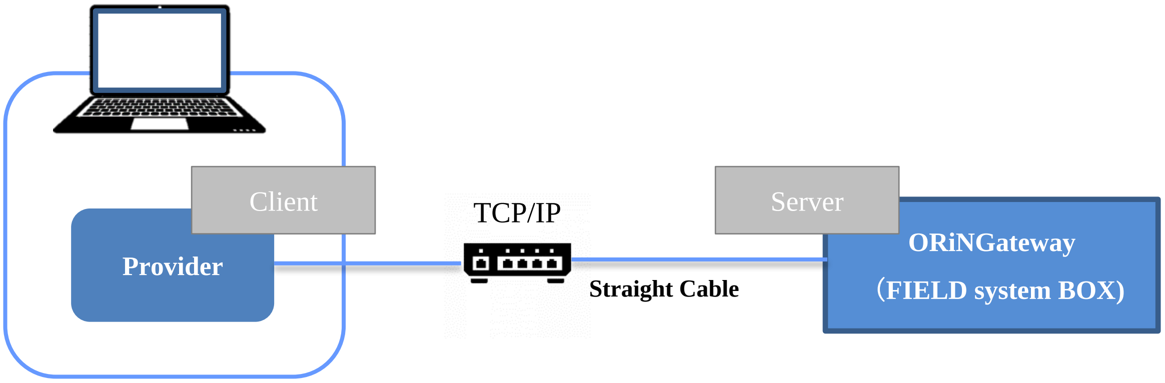

Fig. 2-1 Connection Diagram

2.2. Operating system installation

The file format for FIELD system provider is DLLs (Dynamic Link Library), which are dynamically loaded when used by the CAO-engine. To use it, you must install ORiN2SDK or manually register the registry by referring to Table 2-1.

Table 2-1 FIELD system provider

File name	CaoProvFIELDsystem.dll
ProgID	CaoProv.FANUC. FIELDsystem
Registry registration	Regsvr32 CaoProvFIELDsystem.dll
Delete Registry Registration	Regsvr32 /u CaoProvFIELDsystem.dll

3. Command Reference

3.1. Method properties

3.1.1. CaoWorkspace classes

3.1.1.1. CaoWorkspace::AddController method

FIELD system provider refer to the parameters passed when AddController method is executed and connect to the appropriate FIELD system BOX. The following are the specifics of AddController method:

FORMAT AddController (<bstrCtrlName:VT_BSTR>,<bstrProvName:VT_BSTR>,
<bstrPcName:VT_BSTR>, [<bstrOption:VT_BSTR>])

<bstrCtrlName>	:	[in] Controller name
<bstrProvName>	:	[in] Provider name. Fixed value="CaoProv.FANUC.FIELDsystem"
<bstrPcName>	:	[in] Provider's running machine name (unused)
<bstrOption>	:	[in] Option string

The following is a list of option strings:

Table 3-1 CaoWorkspace::AddController Option-String

Option	Meaning
Server = <connection parameter>	Required. IP address of the destination FIELD system BOX. The port number is fixed at 5007 when connecting.
In Controller[=<FIELD system Host controller >][3]	Specifies name attribute of the destination controller. If it is omitted, the controller name specified by the first argument of AddController is used to connect.
ConnTimeout[=<timeout>]	Specifies the timeout in milliseconds for the connection. If the set value is 1000 milliseconds or less, it is set to 1000 milliseconds. (Default: 5000)
Timeout[=<timeout>]	Specifies the timeout period in milliseconds when sending or receiving commands. (Default: 500)

[3] Controller class is the class that is the origin of the data model handled by FIELD converter.

Usage example(C#)

```
CaoEngine caoEng;
CaoWorkspaces caoWss;
CaoWorkspace caoWs;
CaoControllers caoCtrls;
CaoController caoCtrl;

caoEng = new CaoEngine();
caoWss = caoEng.Workspaces;
caoWs = caoWss.Item(0);
caoCtrls = caoWs.Controllers;

// Connection
caoCtrl = caoWs.AddController("Sample", "CaoProv.FANUC.FIELDsystem", null,
                              "Server=192.168.0.10:40000,
                              Controller=controller00001,ConnTimeout=5000, Timout=5000");
```

3.1.2. CaoController classes

3.1.2.1. CaoController::ExtensionNames Properties

Gets a list of extended board names that can be specified by AddExtension method. For FIELD system provider, the instance IDs of the data models that are linked under controller of FIELD system are listed. The data models associated with controller depend on FIELD converters. For more information, refer to FIELD system's instruction manual.

Usage example(C#)

```
// Get extended board name list
 string[] names = caoCtrl.GetExtensionNames(null);
 _
```

3.1.2.2. CaoController::AddExtension method

To view the data in FIELD system with FIELD system provider, you must add a CaoExtension object that corresponds to the data model of FIELD system.

The extension board name can be any string or an instance ID obtained by CaoController::GetExtensionNames. If you use any string, specify the instance ID in the option. The following are the specifics of AddExtension method:

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

<bstrCtrlName>	:	[in] Expansion Board Name
<bstrOption>	:	[in] Option string

The following options are specified in the option string:

Table 3-2 Option strings for CaoController::AddExtension

Option	Meaning
InstanceId=[<FIELD system data model instance ID>]	Specifies the instance ID of the data model to be acquired. If you specify an arbitrary character string for the extension board name, be sure to specify the instance ID. The instance ID is [data model name] + a 5-digit number. e.g. controller_robot_controller 00001

Usage example(C#)

```
// Adding an expansion board
// Specify InstanceId option when using any expansion board name.
CCaoExtension caoExt = caoCtrl.AddExtension("Extension", "InstanceId=controller00001");

// Use the name obtained by CaoController::GetExtensionNames
var names = caoCtrl.GetExtensionNames(null);
CCaoExtension caoExt2 = caoCtrl.AddExtension(names[0], null);
```

3.1.2.3. CaoController::VariableNames Properties

Get system variable names shown in Table 3-3.

3.1.2.4. CaoController::AddVariable method

Adds a variable object to CaoController. Only the variable names shown in Table 3-3 can be used as variable names.

FORMAT AddVariable (<bstrVariableName:VT_BSTR>, [<bstrOption:VT_BSTR>])

<bstrVariableName>	:	[in] Variable name
<bstrOption>	:	[in] Option string

Usage example(C#)

```
// Add variable for getting provider version
CCaoVariable version = caoCtrl.AddVariable("@VERSION", null);
```

3.1.3. CaoExtension classes

3.1.3.1. CapExtension::get_ID Property

Get CaoExtension identity. For FIELD system provider, CaoExtension represents the instance identity of the data model for the corresponding FIELD system.

Usage example(C#)

```
// Adding an expansion board
CCaoExtension caoExt = caoCtrl.AddExtension("Extension", "InstanceId=controller00001");
// Get ID of expansion board
var id = caoExt.ID;
```

3.1.3.2. CaoExtension::VariableNames Properties

Get a list of variable names. For FIELD system provider, lists the attribute names of the data models of CaoExtension corresponding FIELD system.

Usage example(C#)

```
// Adding an expansion board
CCaoExtension caoExt = caoCtrl.AddExtension("Extension", "InstanceId=controller00001");
// Get list of variable names
var attrNames = caoExt.GetVariableNames(null);
```

3.1.3.3. CaoExtension::AddVariable method

Adds a variable object to CaoExtension object. Variable names can only be variable names that can be retrieved with VariableNames property.

FORMAT AddVariable (<bstrVariableName:VT_BSTR>, [<bstrOption:VT_BSTR>])

<bstrVariableName>	:	[in] Variable name
<bstrOption>	:	[in] Option string

Usage example(C#)

```
// Adding an expansion board
CCaoExtension caoExt = caoCtrl.AddExtension("Extension", "InstanceId=controller00001");
// Get list of variable names
var attrNames = caoExt.GetVariableNames(null);
// Add Variable
CCaoVariable val = caoCtrl.AddVariable(attrNames[0], null);
```

3.1.4. CaoVariable classes

3.1.5. CaoVariable::get_Value Property

Gets the provider information for the system variable of CaoController object, or the most recent value of FIELD system attribute for the variable of CaoExtension object.

Usage example(C#)

```
// Adding an expansion board
CCaoExtension caoExt = caoCtrl.AddExtension("Extension", "InstanceId=controller00001");
// Get list of variable names
var attrNames = caoExt.GetVariableNames(null);
// Add Variable
CCaoVariable val = caoCtrl.AddVariable(attrNames[0], null);
// Acquisition of the latest value
var value = val.Value;
```

3.2. Variable list

3.2.1. CaoController classes

Table 3-3 CaoController Class System Variable List

Variable name	Description	Attribute		Page
		Get	Put	
@MAKER_NAME	Obtain the manufacturer's name.	○	-	12
@VERSION	Get the provider version.	○	-	12

3.2.1.1. Provider information-related variables

Gets the info held by FIELD system provider.

3.2.1.1.1. @MAKER_NAME

Obtain the manufacturer's name.

Data Type	Description
VT_BSTR	FANUC

3.2.1.1.2. @VERSION

Get the provider version.

Data Type	Description
VT_BSTR	*.*.*

3.2.2. CaoExtension classes

The variables that CaoExtension can use depend on the data model of the corresponding FIELD system. The following is a list of data models and variables as of the third Application SDK of FIELD system (excluding data models that do not have variables):

Table 3-4 CaoExtension class-variable list

<table>
<tr><th colspan="2" rowspan="2">Variable name</th><th colspan="2" rowspan="2">Description</th><th colspan="2" rowspan="2">Data Type</th><th colspan="2">Attribute</th></tr>
<tr><th>Get</th><th>Put</th></tr>
<tr><td colspan="2">Controller</td><td colspan="6">Class that represents controller information</td></tr>
<tr><td></td><td>Name</td><td colspan="2">Any name of the controller</td><td colspan="2">VT_BSTR</td><td>○</td><td>-</td></tr>
<tr><td></td><td>Manufacturer</td><td colspan="2">Manufacturer of the controller</td><td colspan="2">VT_BSTR</td><td>○</td><td>-</td></tr>
<tr><td></td><td>Controller_type</td><td colspan="2">Controller type</td><td colspan="2">VT_BSTR</td><td>○</td><td>-</td></tr>
<tr><td></td><td>Model</td><td colspan="2">Model name</td><td colspan="2">VT_BSTR</td><td>○</td><td>-</td></tr>
<tr><td></td><td>Ip_address</td><td colspan="2">IP address
(Converter connection destination)</td><td colspan="2">VT_BSTR</td><td>○</td><td>-</td></tr>
<tr><td colspan="2">Controller_cnc</td><td colspan="6">Class that represents CNC-specific information</td></tr>
<tr><td></td><td>Software_name</td><td colspan="2">CNC software name</td><td colspan="2">VT_BSTR</td><td>○</td><td>-</td></tr>
<tr><td></td><td>Software_version</td><td colspan="2">CNC software version</td><td colspan="2">VT_BSTR</td><td>○</td><td>-</td></tr>
<tr><td></td><td>Total_paths</td><td colspan="2">Number of CNC systems</td><td colspan="2">VT_I4</td><td>○</td><td>-</td></tr>
<tr><td></td><td>Total_axes</td><td colspan="2">Number of axes</td><td colspan="2">VT_I4</td><td>○</td><td>-</td></tr>
<tr><td></td><td>Total_servos</td><td colspan="2">Number of CNC servo axes</td><td colspan="2">VT_I4</td><td>○</td><td>-</td></tr>
<tr><td></td><td>Total_spindles</td><td colspan="2">Number of CNC spindles</td><td colspan="2">VT_I4</td><td>○</td><td>-</td></tr>
<tr><td></td><td>Plc_software_name</td><td colspan="2">PLC(PMC) Software name</td><td colspan="2">VT_BSTR</td><td>○</td><td>-</td></tr>
<tr><td></td><td>Plc_software_version</td><td colspan="2">PLC(PMC) Software Version</td><td colspan="2">VT_BSTR</td><td>○</td><td>-</td></tr>
<tr><td></td><td>Total_plc_paths</td><td colspan="2">Number of PLC (PMC) systems</td><td colspan="2">VT_I4</td><td>○</td><td>-</td></tr>
<tr><td></td><td>Boot_software_name</td><td colspan="2">BOOT software name</td><td colspan="2">VT_BSTR</td><td>○</td><td>-</td></tr>
<tr><td></td><td>Boot_software_version</td><td colspan="2">BOOT software version</td><td colspan="2">VT_BSTR</td><td>○</td><td>-</td></tr>
<tr><td rowspan="3"></td><td rowspan="3">Hardware_info</td><td colspan="2">Hardware information</td><td colspan="2">VT_ARRAY|
VT_VARIANT</td><td rowspan="3">○</td><td rowspan="3">-</td></tr>
<tr><td rowspan="2"></td><td>Type</td><td>0</td><td>VT_BSTR</td></tr>
<tr><td>ID</td><td>1</td><td>VT_BSTR</td></tr>
</table>

Controller_cnc_axis		Class representing the CNC axis information			
	Axis_name	Axis name	VT_BSTR	○	-
	Axis_type	Axis (servo-axis (linear linear/ rotational rotation) Spindle spindle)	VT_BSTR	○	-
	Axis_number	Absolute axis (spindle) number	VT_I4	○	-
	Path_number	Line number to which axis (main axis) belongs	VT_I4	○	-
	Path_axis_number	Axis number in the system	VT_I4	○	-
Controller_cnc_path		Class representing CNC grid information			
	Name	Line name	VT_BSTR	○	-
	Path_number	Own line number	VT_I4	○	-
	Total_axes	Number of axes in the system	VT_I4	○	-
	Path_type	Control Type	VT_BSTR	○	-
Controller_laser		Class representing the information of the laser controller			
	Software_name	Software name	VT_BSTR	○	-
	Software_version	Software version	VT_BSTR	○	-
	Total_axes	Number of axes	VT_I4	○	-
	Plc_software_name	PLC(PMC) Software name	VT_BSTR	○	-
	Plc_software_version	PLC(PMC) Software Version	VT_BSTR	○	-
	Total_plc_paths	Number of PLC (PMC) systems	VT_I4	○	-
	Boot_software_name	BOOT software name	VT_BSTR	○	-
	Boot_software_version	BOOT software version	VT_BSTR	○	-
	Hardware_info	Hardware information	VT_ARRAY| VT_VARIANT	○	-
		Type	0	VT_BSTR	
		ID	1	VT_BSTR	
Controller_laser_oscillator		Class representing the information of the laser oscillator			
	Name	Name of the laser oscillator	VT_BSTR	○	-
	Path_number	Strain number to which the laser oscillator belongs	VT_I4	○	-
	Oscillator_number	Laser oscillator number	VT_I4	○	-

Controller_robot_contorller		Class representing the robot controller information			
	Version_info	Software version of the object	VT_ARRAY|	○	-

Variable name	Description	Data Type	Attribute	
			Get	Put
		VT_VARIANT		
	Types of Software	0 VT_BSTR		
	Version	1 VT_BSTR		
Total_groups	Total number of groups (total control unit of the controller)	VT_I4	○	-
Application_name	Application name	VT_BSTR	○	-
Controller_robot_group	Class representing the control system of the robot controller			
Total_axes	Number of axes	VT_I4	○	-
Group_name	Group name	VT_BSTR	○	-
Group_number	Group number	VT_I4	○	-
Mechanical_name	Model	VT_BSTR	○	-
Total_robot_axes	Number of robot axes	VT_I4	○	-
Total_external_axes	Number of additional axes	VT_I4	○	-
Controller_robot_axis	Class representing an axis controlled by the robot controller			
Max_amp_current	Maximum Amplifier Current	VT_R8	○	-
Axis_name	Axis name	VT_BSTR	○	-
Axis_number	Axis number	VT_I4	○	-
Axis_type	Axis type (Linear linear/ Rotation rotation)	VT_BSTR	○	-
Controller_injection	Class representing the controller information of the injection machine			
Name	Machine name of the injection molding machine	VT_BSTR	○	-
Controller_wirecut	Class representing information for wire-cut EDM			
Name	Name of the wire-cut discharge machine	VT_BSTR	○	-
Controller_sensor_individual	Class representing the information of the individual sensor			
Name	Sensor name	VT_BSTR	○	-
Manufacturer	Manufacturer	VT_BSTR	○	-
Model	Model No.	VT_BSTR	○	-
Description	Any comment	VT_BSTR	○	-
Kind	Sensor type	VT_BSTR	○	-

Variable name		Description		Data Type	Attribute	
					Get	Put
	Installation	Installation site		VT_BSTR	○	-
	Usage	Typical Uses		VT_BSTR	○	-
Status		Class that represents a common state between controllers				
	Controller_status_number	Controller status number		VT_I4	○	-
	Program_name	Main program name (1st line)		VT_BSTR	○	-
	Act_program_name	Running program name (1st line)		VT_BSTR	○	-
	Alarm_flag	Alarm Flag		VT_I4	○	-
	Alarm_info	Alarm information		VT_ARRAY| VT_VARIANT	○	-
		Alarm type	0	VT_BSTR		
		Alarm number	1	VT_BSTR		
		Alarm Messages	2	VT_BSTR		
	Lot_number	Lot number		VT_BSTR	○	-
	Product_amount	Production volume		VT_I4	○	-
	Good_product_amount	Production volume of good products		VT_I4	○	-
	Ng_product_amount	Production volume of defective products		VT_I4	○	-
	Act_cycle_time	Cycle time (actual)		VT_R8	○	-
	Total_power_on_time	Accumulated power ON time (entire controller)		VT_R8	○	-
	Total_operation_time	Accumulated operating time (entire controller)		VT_R8	○	-
	Act_power_consumption	Actual power consumption (entire controller)		VT_I4	○	-
	Product_amount_per_cycle	Production volume per cycle		VT_I4	○	-
Status_cnc		Class representing a common CNC state				
	Fan_speed	Fan speed		VT_I4	○	-
	Fan_status	Fan Replacement Information		VT_BSTR	○	-
	Plc_status	PLC (PMC) Status		VT_BSTR	○	-
Status_cnc_path		Class that represents the state of the CNC system				
	Mode	Operation mode		VT_BSTR	○	-
	Status	Operating status		VT_BSTR	○	-

Variable name		Description	Data Type	Attribute	
				Get	Put
	Program_name	Main program name	VT_BSTR	○	-
	Act_program_name	Running program name	VT_BSTR	○	-
	Program_comment	Main program comment	VT_BSTR	○	-
	Act_program_comment	Comment of the running program	VT_BSTR	○	-
	Sequence_number	Sequence number	VT_BSTR	○	-
	Act_rotation_speed	Number of revolutions (actual value)	VT_R8	○	-
	Feedrate	Feedrate (Actual measurement)	VT_R8	○	-
	Act_f_code	F code specified	VT_BSTR	○	-
	Act_m_code	M code specified	VT_BSTR	○	-
	Act_s_code	Commanded S code	VT_BSTR	○	-
	Act_t_code	T code specified	VT_BSTR	○	-
	Act_b_code	B code specified	VT_BSTR	○	-
	Product_amount	Total number of components produced by a running program	VT_R8	○	-
	Total_product_amount	Total number of parts produced so far	VT_R8	○	-
	Operation_time	Program operating time	VT_R8	○	-
	Cutting_time	Program cutting time	VT_R8	○	-
	Dryrun_signal	Dry run signal status	VT_I4	○	-
	Machine_lock_signal	Machine lock (all axes) signal status	VT_I4	○	-
	Work_coordinate_number	Work coordinate system selection number	VT_I4	○	-
	Cut_override	Feedrate override	VT_I4	○	-
	Rapid_override	Rapid traverse override	VT_I4	○	-
Status_cnc_axis		Class that represents the state of the axis being operated by the CNC			
	Axis_load	Load on the shaft	VT_I4	○	-
	Rotation_override	Rotational speed override	VT_I4	○	-
	Temperature	Motor temperature	VT_I4	○	-
	Absolute_position	Absolute coordinate	VT_R8	○	-
	Relative_position	Relative coordinates	VT_R8	○	-

Variable name		Description	Data Type	Attribute	
				Get	Put
	Machine_position	Machine coordinates	VT_R8	○	-
	Rotation_speed	Rotational speed	VT_R8	○	-
	Insulation_resistance_data	Insulation resistance	VT_R8	○	-
	Leakage_detection_number	Insulation resistance deteriorated state	VT_I4	○	-
Status_laser_oscillator		Class representing the state of the laser oscillator			
	Power_peek	Laser output (peak) command value	VT_I4	○	-
	Power_bottom	Laser output (bottom) command value	VT_I4	○	-
	Frequency	Laser output pulse frequency command value	VT_I4	○	-
	Duty	Laser output duty command value	VT_I4	○	-
	Sequence	Starting state of the laser oscillator	VT_BSTR	○	-
	Beam_on	Beam-on state	VT_BSTR	○	-
	Output_monitor	Laser actual output intensity	VT_I4	○	-
	Water_flow	Cooling water consumption	VT_R8	○	-
	Run_on_time	Accumulated power supply ON time of the laser oscillator	VT_R8	○	-
	Operation_time	Laser oscillator command time	VT_R8	○	-
	Cabinet_temperature	Temperature in warehouse	VT_R8	○	-
	Cabinet_humidity	Warehouse humidity	VT_R8	○	-
	Cabinet_water_temperature	Cooling water temperature	VT_R8	○	-
	Dewpoint	Dew Point	VT_R8	○	-
	Discharge_tube_pressure	Pressure in the discharge tube	VT_R8	○	-
	Power_consumption	Power consumption of the laser oscillator	VT_R8	○	-
	Total_power_consumption	Total Power Consumption of Laser Oscillators	VT_R8	○	-
	Gas_supply_pressure	Laser gas supply pressure	VT_R8	○	-
	Calibration_coefficient	PCF	VT_I4	○	-
	Laser_cavity_unit_output	Laser resonator unit output value	VT_I4	○	-
	Fast_fiber_break_detector	FFBD (High-speed fiber-break	VT_I4	○	-

Variable name	Description	Data Type	Attribute	
			Get	Put
	detection)			
Cutting_index	Processing index	VT_I4	○	-
Exhaust_pump	Exhaust pump operating time	VT_R8	○	-
Power_supply	Power supply	VT_ARRAY| VT_VARIANT	○	-
	Drive power supply number	0 VT_I4		
	Drive power supply information	1 VT_BSTR		
	Channel	2 VT_I4		
	Voltage value	3 VT_R8		
	Current value	4 VT_R8		
	Power value	5 VT_R8		
	Heat sink temperature	6 VT_R8		
Status_plc	Class that represents the state of the PLC			
Status	Operating status	VT_BSTR	○	-
Status_sensor_individual	Class representing the status of individual sensors			
Status	Status of sensor	VT_BSTR	○	-
Data	Acquired raw values and units	VT_ARRAY| VT_VARIANT	○	-
	Raw value of the sensor	0 VT_BSTR		
	Unit of sensor (common code)	1 VT_BSTR		
Status_sensor_individual_bulk_info	Class that represents the state of the bulk data of the sensor			
Type	Type	VT_BSTR	○	-
Start_address	Address	VT_BSTR	○	-
Length	Number of data	VT_I4	○	-
Data_type	Data Type	VT_BSTR	○	-
Status_robot_controller	Class that represents the state of the robot controller			
Fault	Flags for alarms	VT_I4	○	-
Running	Active flag	VT_I4	○	-
Paused	Stopping Flag	VT_I4	○	-
Servo_ready	Servo ready flag	VT_I4	○	-
Tp_enable	TP Switch Flags	VT_I4	○	-

Variable name		Description		Data Type	Attribute	
					Get	Put
	Battery_alarm	Battery Alarm Flags		VT_I4	○	-
	Estop_operator_panel	Emergency Stop Panel Flags		VT_I4	○	-
	Estop_teach_pendant	Flag of Emergency Stop TP		VT_I4	○	-
	Estop_external	External Emergency Stop Flag		VT_I4	○	-
	Fence_open	Fence open flag		VT_I4	○	-
	Enabling_device	Deadman switch flag		VT_I4	○	-
Status_robot_controller_variable		Class that represents the state of the robot controller's variables				
	Type	Type		VT_BSTR	○	-
	Name	NAME		VT_BSTR	○	-
	Value	Value		VT_BSTR	○	-
	Unit	Unit		VT_BSTR	○	-
Status_robot_controller_task		Class that represents the state of the task running on the robot controller				
	Task	Robot Tasks		VT_ARRAY| VT_VARIANT	○	-
		Title of Task	0	VT_BSTR		
		Program name	1	VT_BSTR		
		Line number	2	VT_I4		
		Status	3	VT_BSTR		
Status_robot_group		Class representing the control system information of the robot controller				
	Is_running	Target Group Activity Flag		VT_I4	○	-
	Is_servo_ready	Target Group Servo Ready Flag		VT_I4	○	-
	User_frame_id	User coordinate system number		VT_I4	○	-
	Tool_frame_id	Tool coordinate system number		VT_I4	○	-
	Joint_position	Object of each axis coordinate		VT_ARRAY| VT_VARIANT	○	-
		Each axis coordinate value	0	VT_ARRAY | VT_R8		
		Unit	1	VT_ARRAY| VT_BSTR		
	Cartesian_position	Cartesian object		VT_ARRAY|	○	-

Variable name	Description		Data Type	Attribute	
				Get	Put
			VT_VARIANT		
	Type	0	VT_BSTR		
	Cartesian coordinate value	1	VT_ARRAY | VT_R8		
	Unit	2	VT_ARRAY| VT_BSTR		
	Format	3	VT_BSTR		
Status_robot_axis	Class representing the status of an axis controlled by the robot controller				
Speed	Speed		VT_R8	○	-
Acceleration	Acceleration		VT_R8	○	-
Torque	Torque		VT_R8	○	-
q_current	Q phase current		VT_R8	○	-
Dis_torque	Disturbance value		VT_R8	○	-
Axis_position	Shaft position		VT_R8	○	-
Current	Current		VT_R8	○	-
Ovc_value	Ovc value		VT_R8	○	-
Pulse	Pulse		VT_R8	○	-
Status_injection	Class that represents the state of the controller in the injection machine				
Injection_controller_status_number	Condition of the injection molding machine controller		VT_BSTR	○	-
Status_wirecut	Class representing the status of the controller of the wire-cut EDM machine				
Controller_status_number	Controller status number		VT_I4	○	-
Program_name	Main program name (1st line)		VT_BSTR	○	-
Act_program_name	Running program name (1st line)		VT_BSTR	○	-
Alarm_flag	Alarm Flag		VT_I4	○	-
Alarm_info	Alarm information		VT_ARRAY| VT_VARIANT	○	-
	Alarm type	0	VT_BSTR		
	Alarm number	1	VT_BSTR		

Variable name	Description	Data Type	Attribute	
			Get	Put
	Alarm Messages	2 VT_BSTR		
Total_power_on_time	Accumulated power ON time (entire controller)	VT_R8	○	-
Total_operation_time	Accumulated operating time (entire controller)	VT_R8	○	-
Act_power_consumption	Total power consumption (entire controller)	VT_I4	○	-
Mode	Operation mode	VT_BSTR	○	-
Sequence_number	Sequence number	VT_BSTR	○	-
Feedrate	Feedrate (Actual measurement)	VT_R8	○	-
Absolute_position	Absolute coordinate	VT_R8	○	-
Relative_position	Relative coordinates	VT_R8	○	-
Machine_position	Machine coordinates	VT_R8	○	-
Total_length	Total machining distance in the current program	VT_R8	○	-
Act_length	Machining completion distance in current program	VT_R8	○	-
Wire_break_flag	Wire break flag	VT_I4	○	-
Act_cutting_times	Current number of machining operations	VT_I4	○	-
Awf_flag	Automatic connection in progress flag	VT_I4	○	-
Cut_wire_flag	Wire Disconnect in Progress Flag	VT_I4	○	-
Consume_warning_flag	Consumables Warning Flag	VT_I4	○	-
Peri_maintenance_warning_flag	Periodic Maintenance Warning Flag	VT_I4	○	-
Quality_maintenance_file_info	Class representing the information in the quality information file			
Name	File name	VT_BSTR	○	-
Type	File type	VT_BSTR	○	-
Quality_maintenance_robot_controller	Class representing the quality information of the robot controller			
Total_power_on_time	Cumulative power ON time	VT_R8	○	-
Total_servo_on_time	Accumulated servo ON time	VT_R8	○	-
Total_running_time	Total operation time	VT_R8	○	-

Variable name	Description	Data Type	Attribute	
			Get	Put
Total_waiting_time	Cumulative standby time	VT_R8	○	-
Quality_maintenance_injection	For each molding cycle of the injection molding machine Class that represents the state of the controller			
Status	Machine condition	VT_BSTR	○	-
Cycle_time	The time between the beginning of type close and the beginning of type close. Time per molding	VT_R8	○	-
Injection_time	Time from start to end of injection	VT_R8	○	-
Extruder_time	Time from weighing start to end	VT_R8	○	-
Minimum_cushion	Screw forward position during injection and holding pressure	VT_R8	○	-
Extruder_position	Screw position at the completion of weighing	VT_R8	○	-
Peak_injection_pressure	Maximum injection pressure during injection and holding pressure	VT_R8	○	-
Shot_number	Number of cycles	VT_I4	○	-
Good_shot	Number of cycles judged to be good by the machine	VT_I4	○	-
Bad_shot	Number of cycles that the machine has determined to be defective	VT_I4	○	-
Remain_shot	Number of cycles until the expected production number set in the machine is reached	VT_I4	○	-
Alarm_code	Alarm code	VT_BSTR	○	-
Vp_transfer_position	Screw position when switching from injection process to holding pressure process	VT_R8	○	-
Temp_mold1	Temperature of mold 1 at a point during one cycle	VT_R8	○	-
Temp_mold2	Temperature of mold 2 at some point during one cycle	VT_R8	○	-

Variable name	Description	Data Type	Attribute	
			Get	Put
Temp_nozzle1	Temperature of nozzle 1 at some point during one cycle	VT_R8	○	-
Temp_nozzle2	Temperature of nozzle 2 at some point during one cycle	VT_R8	○	-
Temp_barrel1	Temperature of barrel 1 at some point in a cycle	VT_R8	○	-
Temp_barrel2	Barrel 2 temperature at some point during one cycle	VT_R8	○	-
Temp_barrel3	Temperature of barrel 3 at some point during one cycle	VT_R8	○	-
Temp_barrel4	Barrel 4 temperature at some point in a cycle	VT_R8	○	-
Temp_nozzle_adapter	Temperature of the nozzle adapter temperature at some point during one cycle	VT_R8	○	-
Temp_hopper	Temperature under the hopper at some point during one cycle	VT_R8	○	-
Extruder_start_position	Screw position at the start of weighing	VT_R8	○	-
Extruder_torque1	Weighing torque 1: Average torque in any section during the weighing process	VT_R8	○	-
Extruder_torque2	Weighing torque 2: Average torque in any section during the weighing process	VT_R8	○	-
Hispeed_monitor1	High-speed monitor 1: Injection pressure at any point during injection and holding pressure process	VT_R8	○	-
Extruder_torque3	Weighing torque 3: Average torque in any section during the weighing process	VT_R8	○	-
Extruder_torque4	Weighing torque 4: Average torque in any section during the weighing	VT_R8	○	-

Variable name	Description	Data Type	Attribute	
			Get	Put
	process			
Hispeed_monitor2	High-speed monitor 2: Injection pressure at any point during the injection and holding pressure process	VT_R8	○	-
Extruder_torque5	Weighing torque 5: Average torque in any section during the weighing process	VT_R8	○	-
Extruder_torque6	Weighing torque 6: Average torque in any section during the weighing process	VT_R8	○	-
Hispeed_monitor3	High-speed monitor 3: Injection pressure at any point during the injection and holding pressure process	VT_R8	○	-
Temp_mold3	Temperature of the mold 3 at a certain point during one cycle	VT_R8	○	-
Temp_mold4	Temperature of the mold 4 at some point during one cycle	VT_R8	○	-
Hispeed_monitor4	High-speed monitor 4: Injection pressure at any point during the injection and holding pressure process	VT_R8	○	-
Hispeed_monitor5	High-speed monitor 5: injection pressure at any point during the injection and holding pressure process	VT_R8	○	-
Consumption_power	Power consumption	VT_R8	○	-
Judge_flag_promoni	The upper and lower limits set on the machine have been exceeded. Monitor items	VT_BSTR	○	-
Judge_flag_hisp_5p	The upper and lower limits set on the machine have been exceeded. Monitor items (high-speed	VT_BSTR	○	-

Variable name	Description	Data Type	Attribute	
			Get	Put
	monitor 1 to 5)			
Judge_flag_ext_torq6p	The upper and lower limits set on the machine have been exceeded. Monitor item (Weighing torque 1 to 6)	VT_BSTR	○	-
Injection_start_press	Pressure at the injection start position	VT_R8	○	-
Precise_extruder3_comp_quantity3	Compression amount of the precision weighing 3	VT_R8	○	-
Precise_extruder3_comp_press3	Compression pressure of the precision weighing 3	VT_R8	○	-
Precise_extruder3_v_p_correction	Correction position of the V-P position of the precision weighing 3	VT_R8	○	-
Backflow_peak	Peak backflow	VT_R8	○	-
Backflow_average	Average backflow	VT_R8	○	-
Event_alarm	Class that represents the alarm event			
Type	Type (ASK character string)	VT_BSTR	○	-
Number	Alarm number (ASK character string)	VT_BSTR	○	-
Severity	Severity (ASK character string)	VT_BSTR	○	-
Message	Messaging (utf-8)	VT_BSTR	○	-
Comment	Commenting (utf-8)	VT_BSTR	○	-
Date	Alarm generated date	VT_BSTR	○	-
Event_variable_info	Class representing the information about the variable at the time of the event			
Dataset_id	Data set ID	VT_BSTR	○	-
Dataset_name	Data Set Name	VT_BSTR	○	-
Variable_info	Variable information	VT_ARRAY| VT_VARIANT	○	-
	Variable Type	0 VT_BSTR		
	Variable Types	1 VT_BSTR		
	Variable address	2 VT_BSTR		

Variable name		Description		Data Type		Attribute	
						Get	Put
			Variable name	3	VT_BSTR		
			Unit of variable (common code)	4	VT_BSTR		
	Tag	Ancillary information		VT_ARRAY| VT_VARIANT		○	-
			Key	0	VT_BSTR		
			Value	1	VT_BSTR		
Event_variable		Class representing the value of the variable at the time of the event					
	Value	Value of the variable		VT_BSTR		○	-
	Tag	Ancillary information		VT_ARRAY| VT_VARIANT		○	-
			Key	0	VT_BSTR		
			Value	1	VT_BSTR		
Archive_maintenance_compress_file_info		Class representing the information of the maintenance backup file					
	Name	File name		VT_BSTR		○	-
	Type	File format		VT_BSTR		○	-

3.3. Error code

FIELD system provider return the following unique failures:

Error name	Error Number	Description
JSON analysis failure	0x801103E9	Returned when an error occurs during JSON analysis of FIELD system API executionresult. Possible causes of the error are as follows: 1. Sending and receiving API requests has failed. 2. There is no single FIELD converter-based measure. Check the status of ORiNGateway logging and FIELD converter data collection from FIELD system administration window.
Access point controller Specified error	0x801104B0	Returns if controller of FIELD system specified for the destination cannot be found.
Instance ID Format error	0x801104B1	Returns an incorrect formatting of the instance ID specified when CaoExtension object was created.

4. Sample program

The following is a sample program (C#) that uses this provider to communicate with FIELD system and retrieve measurements.

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,
                                ConnTimout=5000,Timout=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()));
            }
```

```
                }
        }
}
```