

MTConnect providers

User's Guide

Version 1.0.2

June 27, 2023

NOTE:

© 2018 DENSO WAVE INCORPORATED

The copyright of this manual belongs to DENSO WAVE INCORPORATED.

The company name or the product name that has been described is a trademark or a registered trademark of each company.

The content on this user's manual may be changed without notice.

[Revision History]

Version	Date	Description
1.0.0	2020-11-04	First edition
1.0.1	2022-07-14	Add @Current_RawData variable, @Probe_RawData variable, user-specified variable
1.0.2	2023-06-27	Fixed a problem where temporary files kept accumulating under certain conditions.

[Operation check Agent]

Agent URL	Version	Notes
https://smstestbed.nist.gov/vds/	Ver1.3.1	Smart Manufacturing Systems (SMS) Test Bed. Data may vary from development time.

No part of this user's manual may be reproduced in any form without permission.

- The content of this user's manual are subject to be changed without notice.
- The contents of this manual have been prepared in a thorough manner. However, please contact us if you notice any questions, mistakes, or omissions.
- Note that we cannot be held responsible for the effects of the operation regardless of the above sections.

Contents

1. Introduction	7
1.1. Provider's data access	8
2. Setting Up Your Environment for Application Development	9
2.1. Setting up a PC development environment	9
2.1.1. Automated installation of providers	9
3. Command Reference	10
3.1. Method/Property List	10
3.2. Method properties	10
3.2.1. CaoWorkspace classes.....	10
3.2.1.1. AddController method.....	10
3.2.2. CaoController classes.....	11
3.2.2.1. GetExtensionNames method	11
3.2.2.2. Extensions Properties	13
3.2.2.3. GetVariableNames method.....	13
3.2.2.4. Variables Properties	13
3.2.2.5. AddExtension method	13
3.2.2.6. AddVariable method.....	16
3.2.2.7. Execute method	17
3.2.3. CaoExtension classes	17
3.2.3.1. GetVariableNames Properties.....	17
3.2.3.2. Variables Properties	17
3.2.3.3. AddVariable method.....	17
3.2.4. CaoVariable classes	18
3.2.4.1. Value Properties.....	18
3.3. Command list	18
3.3.1.1. GetDeviceNames.....	18
3.3.1.2. GetTypeNames	19
3.3.1.3. GetDataItemIDs.....	19
3.4. Variable list.....	20
3.4.1. CaoController class-variable	20
3.4.1.1. @LoadingTime.....	20
3.4.1.2. @UpdateDateTime	21
3.4.2. CaoExtension class-variable.....	21

3.4.2.1. Dynamic variable	22
3.4.2.2. Static variable	24
3.4.2.3. User-specified variable	28
4. PROGRAMMING BY PROVIDERS	29
4.1. Sample Programming to Retrieve POSITIN and ANGLE	29
4.1.1. Sample program	29
4.1.1.1. Connection	31
4.1.1.2. Retrieves POSITION and ANGLE data.	31
4.1.1.3. Disconnect.....	33
5. Provider error code	34

1. Introduction

This user's guide is for providers accessing data collected by adapters through MTConnect's Agent. Fig. 1-1 shows the system configuration diagram assumed by the provider. From now on, the provider is simply referred to as the provider.

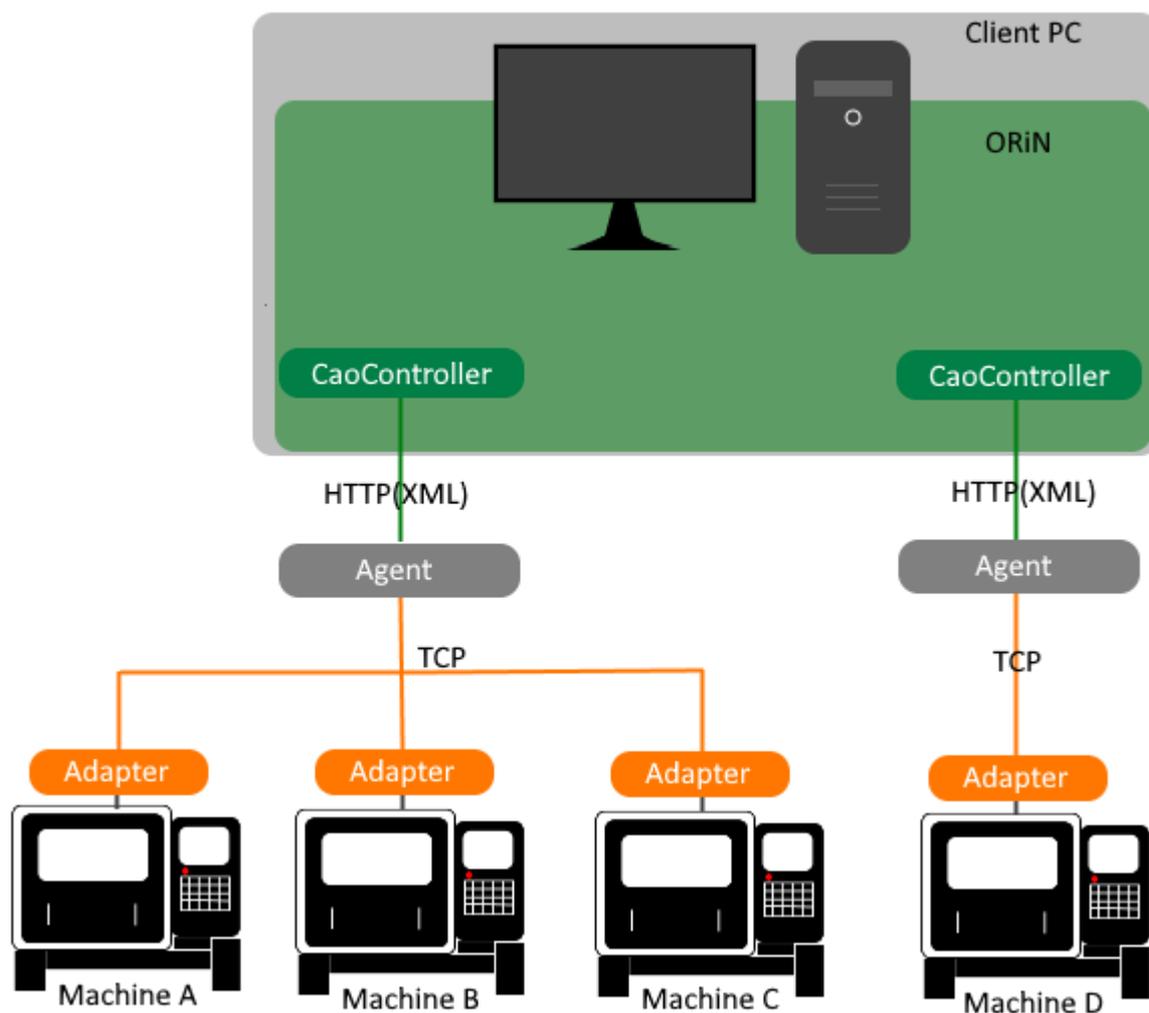


Fig. 1-1 Configuration Diagram

Providers connect with MTConnect's Agent through a CaoController object. Agent contains data from at least one Adapter in HTTP.

Fig. 1-2 shows the correspondence between providers' objects and MTConnect elements.
 (※An example. It does not represent everything.)

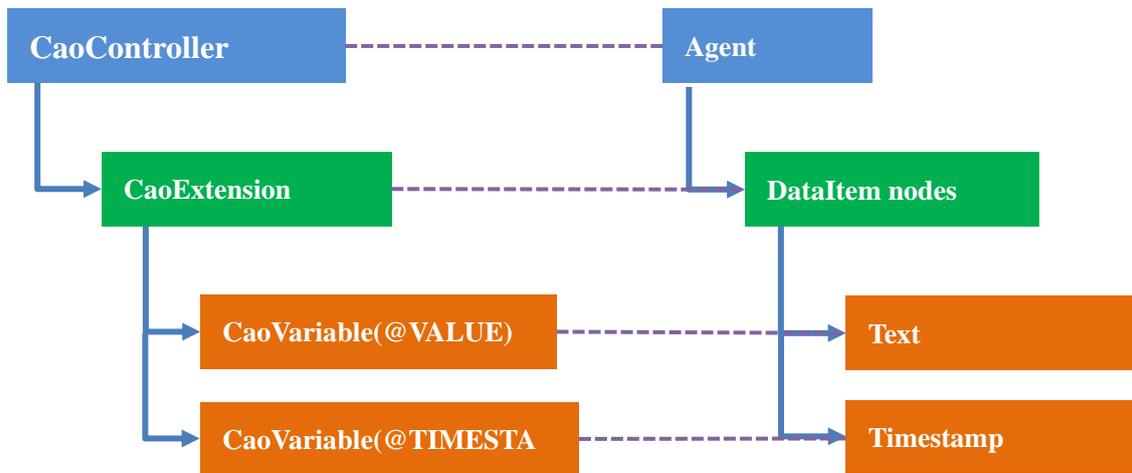


Fig. 1-2 Correspondence between providers' objects and MTConnect elements

CaoExtension corresponds to Agent's DataItem node. You can create CaoExtension with the id attribute of DataItem node. CaoExtension lower-layer CaoVariable correspond to the attributes of the nodes in current XML that have the same DataItemId attributes as Name properties of CaoExtension, and to Text.

1.1. Provider's data access

When the provider successfully connects to Agent, it automatically launches a worker thread and continues to periodically retrieve current to Agent on the worker thread. Note that the provider's data is accessing the value obtained by the worker thread, so it does not necessarily represent the present value of Agent.

2. Setting Up Your Environment for Application Development

2.1. Setting up a PC development environment

2.1.1. Automated installation of providers

If you have ORiN2 SDKs installed, you are ready to use the providers.

However, you should set up your development environment separately and have a programming environment that supports Component Object Model.

3. Command Reference

3.1. Method/Property List

Table 3-1 List of methods and properties

Category	Methods/Properties ¹	Function	See Also
CaoWorkspace			
	AddController	M Connected to controller	P.10
CaoController			
	GetExtensionNames	M Obtaining a list of expansion board names that can be connected	P.11
	Extensions	P Retrieving an Expansion Board Collection Retained by the Controller	P.13
	GetVariableNames	M Get a list of variable names that can be connected	P.13
	Variables	P Retrieving Variable Collections Held by the Controller	P.13
	AddExtension	M Adding an Expansion Board Object	P.13
	AddVariable	M Adding Variable Objects	P.16
	Execute	M Execute Extended Commands	P.17
CaoExtenstion			
	GeVariableNames	M Get a list of variable names that can be connected	P.17
	Variables	P Retrieving Variable Collections Retained by a Task	P.17
	AddVariable	M Adding Variable Objects	P.17
CaoVariable			
	Value	P Get/set value	P.18

3.2. Method properties

3.2.1. CaoWorkspace classes

3.2.1.1. AddController method

Add controller objects to CaoWorkspace. Providers refer to the parameters passed when AddController method is executed and connect to the appropriate Agent. The following are the specifics of AddController method:

SYNOPSIS

AddController

```
(
    "<controller name>", // Controller name (optional)
```

¹ M: Indicates methods, P: properties, and E: events, respectively.

```

    "CaoProv.MTConnect",           // Provider name (fixed)
    "<machine name>",             // Provider execution machine name (optional)
    "<Option>"                     // Option character string (optional)
)

```

Option

The following is an optional specification for Option character string: Option character string is a comma (,) string consisting of the options listed below.

Option	Required	Description	Value Range
Server	○	Specify the name of the server to connect to. (Example). http://192.168.0.1	-
Timeout	-	Specify a timeout (ms) when acquiring a value. If this timeout is exceeded when accessing CaoVariable::Value, an error will occur. The default value is 500ms.	0 or more
Sampling	-	Specify the sampling period of the worker thread. The default value is 500ms.	0 or more

Usage example

```

// Engine
ORiN2.ManagedCAO.CCaoEngine engine = new ORiN2.ManagedCAO.CCaoEngine();
// Workspace
ORiN2.ManagedCAO.CCaoWorkspace workspace = engine.AddWorkspace("NewWrks", "");
// Controller
ORiN2.ManagedCAO.CCaoController controller= workspace.AddController("mtconnect",
    "CaoProv.MTConnect",
    "",
    "server=http://smstestbed.nist.gov/vds");

```

3.2.2. CaoController classes

3.2.2.1. GetExtensionNames method

Retrieves a listing of the extension board names that can be used in AddExtension method. The extension board names that can be used by providers are Agent the id attribute of the existing DataItem node.

SYNOPSIS

GetExtensionNames

(

```
"<Option>" // Option character string
)
```

Option

The following is an optional specification for Option character string: Option character string is a comma (,) string consisting of the options listed below.

Option	Required	Description
Device	-	Specifies name properties of Device. Specify Device options to get a list of DataItem node IDs only in the specified Device node. Case is case sensitive.
Type	-	Specifies type properties of DataItem. If Type option is specified, the IDs of DataItem nodes with limited data types are retrieved. Only relevant if Device option is specified. If Device option is omitted, the specification of Type option is ignored. Case is case sensitive.

Optional combinations have three patterns, each of which changes the target DataItem node.

- Both omitted.
Retrieves an identity listing of all DataItem nodes present in Agent.
- Specify the Device options.
Retrieves an identity listing for all DataItem nodes that are less than or equal to Device node with the specified name properties.
- Device Options and Type options.
Retrieves the identity lists of all DataItem nodes that have type attribute of the specified data type, below Device node with name attribute specified by the device name.

Usage example

```
// Argument omitted
String[] extensionNames1 = controller.GetExtensionNames("");

// Specify the device name.
String[] extensionNames2 = controller.GetExtensionNames("GFAGie01");
```

```
// Specify device name and data type
```

```
String[] extensionNames3 = controller.GetExtensionNames(new string[2]{ "GFAgie01", "POSITION" });
```

3.2.2.2. Extensions Properties

Gets the expansion board collection held by the controller.

Usage example

```
// Expansion Board Collection Acquisition
```

```
ORiN2.ManagedCAO.CCaoExtensions extensions = controller.Extensions;
```

```
// Expansion board acquisition
```

```
ORiN2.ManagedCAO.CCaoExtension extension = extensions[0];
```

3.2.2.3. GetVariableNames method

Gets a list of variable names that can be connected. The variable name obtained by this property can be used as the first argument of AddVariable method described later. AddVariable method.

SYNOPSIS

GetVariableNames

```
(
    "<Option>" // Option character string (not used)
)
```

Usage example

```
// Get variable name list
```

```
String[] variableNames = controller.GetVariableNames("");
```

3.2.2.4. Variables Properties

Gets a collection of variables that the controller holds.

Usage example

```
// Variable Collection Retrieval
```

```
ORiN2.ManagedCAO.CCaoVariables variables = controller.Variables;
```

```
// Variable acquisition
```

```
ORiN2.ManagedCAO.CCaoVariable variable = variables[0];
```

3.2.2.5. AddExtension method

Add an extension board object to CaoController. The following are the specifics of AddExtension method:

SYNOPSIS

AddExtension

```
(
    "<expansion board name>", // Specify the value of the ID attribute of DataItem node.
    "<Option>"                // Option character string (not used) )
```

Option

The following is an optional specification for Option character string: Option character string is a comma (,) string consisting of the options listed below.

Option	Required	Description	Value Range
Array	-	<p>Specifies whether or not the value of the dynamic variable (P. 20) is to be obtained as an array type.²²</p> <p>When False is specified in Array option, only the first one is acquired even if Current contains more than one data with the same id as the id attribute of DataItem specified in AddExtension.</p> <p>When True is specified, data is always acquired using the array type regardless of the data count.</p> <p>The default value is false.</p>	<p>True</p> <p>False</p>

Usage example

```
// Adding an expansion board
ORiN2.ManagedCAO.CCaoExtension extension = controller.AddExtension("GFAgie01-X_2", "");
```

3.2.2.5.1. How to check the expansion board name

The extended board name can be the id attribute value of DataItem node defined in probe XML of Agent, and the id attribute value can be retrieved by GetExtensionNames method or viewed in the browser.

How to check on the browser

You can confirm this by entering "Server Option + "/" probe" of AddController" in the URL entry field.

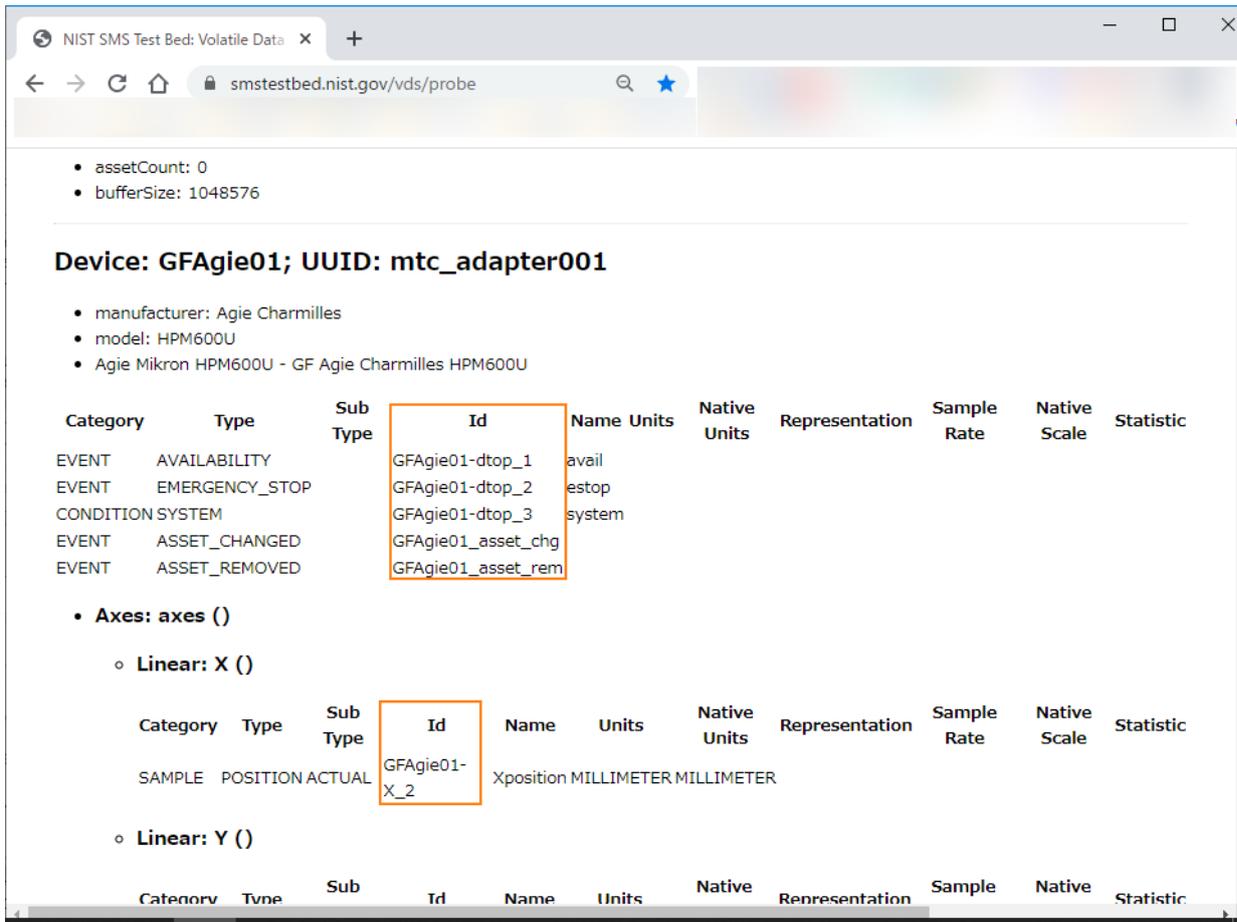


Fig. 3-1 Browser-based probe

Shows probe XML of Agent when it is checked in the browser. The id attribute value of DataItem node is defined in the part enclosed by GFAgie01-X_2 in the figure.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/xsl" href="/styles/Devices.xsl"?>
3 <MTConnectDevices xmlns:tns="urn:mtconnect.org:MTConnectDevices:1.3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4 xsi:schemaLocation="urn:mtconnect.org:MTConnectDevices:1.3 /schemas/MTConnectDevices_1.3.xsd"/>
5 <Header creationTime="2020-03-10T22:54:55Z" sender="7e5bfda23f29" instanceId="1569424750" version="1.4.0.10" assetBufferSize="1024" assetCount="0" bufferSize="1048576"/>
6 <Devices>
7 <Device id="GFAGie01" name="GFAGie01" uuid="mtc_adapter001">
8 <Description manufacturer="Agie Charmilles" model="HPM600U">Agie Mikron HPM600U - GF Agie Charmilles HPM600U</Description>
9 <DataItem category="EVENT" id="GFAGie01-dtop_1" name="avail" type="AVAILABILITY"/>
10 <DataItem category="EVENT" id="GFAGie01-dtop_2" name="estop" type="EMERGENCY_STOP"/>
11 <DataItem category="CONDITION" id="GFAGie01-dtop_3" name="system" type="SYSTEM"/>
12 <DataItem category="EVENT" id="GFAGie01_asset_chg" type="ASSET_CHANGED"/>
13 <DataItem category="EVENT" id="GFAGie01_asset_rem" type="ASSET_REMOVED"/>
14 </DataItems>
15 <Components>
16 <Axes id="GFAGie01-axes_1" name="axes">
17 <Components>
18 <Linear id="GFAGie01-X_1" name="X">
19 <DataItems>
20 <DataItem category="SAMPLE" id="GFAGie01-X_2" name="Xposition" nativeUnits="MILLIMETER" subType="ACTUAL" type="POSITION" units="MILLIMETER"/>
21 </DataItems>
22 </Linear>
23 <Linear id="GFAGie01-Y_1" name="Y">
24 <DataItems>
25 <DataItem category="SAMPLE" id="GFAGie01-Y_2" name="Yposition" nativeUnits="MILLIMETER" subType="ACTUAL" type="POSITION" units="MILLIMETER"/>
26 </DataItems>
27 </Linear>
28 <Linear id="GFAGie01-Z_1" name="Z">
29 <DataItems>
30 <DataItem category="SAMPLE" id="GFAGie01-Z_2" name="Zposition" nativeUnits="MILLIMETER" subType="ACTUAL" type="POSITION" units="MILLIMETER"/>
31 </DataItems>
32 </Linear>
33 <Rotary id="GFAGie01-C_1" name="C">
34 <DataItems>
35 <DataItem category="SAMPLE" id="GFAGie01-C_2" name="Cposition" nativeUnits="DEGREE" subType="ACTUAL" type="ANGLE" units="DEGREE"/>
36 </DataItems>
37 </Rotary>
38 <Rotary id="GFAGie01-A_1" name="A">
39 <DataItems>
40 <DataItem category="SAMPLE" id="GFAGie01-A_2" name="Aposition" nativeUnits="DEGREE" subType="ACTUAL" type="ANGLE" units="DEGREE"/>
41 </DataItems>
42 </Rotary>
43 </Components>
44 </Axes>
45 <Controller id="GFAGie01-controller_basico_1" name="controller_basico">
46 <DataItems>
47 <DataItem category="EVENT" id="GFAGie01-controller_basico_2" name="Fovr" nativeUnits="PERCENT" subType="OVERRIDE" type="PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
48 <DataItem category="EVENT" id="GFAGie01-controller_basico_3" name="Sovr" nativeUnits="PERCENT" type="ROTARY_VELOCITY_OVERRIDE" units="PERCENT"/>
49 <DataItem category="CONDITION" id="GFAGie01-controller_basico_4" name="Servo" type="ACTUATOR"/>
50 <DataItem category="CONDITION" id="GFAGie01-controller_basico_5" name="conme" type="COMMUNICATIONS"/>
51 <DataItem category="CONDITION" id="GFAGie01-controller_basico_6" name="pneumatic" type="COMMUNICATIONS"/>
52 <DataItem category="CONDITION" id="GFAGie01-controller_basico_7" name="hydraulic" type="COMMUNICATIONS"/>
53 <DataItem category="CONDITION" id="GFAGie01-controller_basico_8" name="logic" type="LOGIC_PROGRAM"/>
54 <DataItem category="CONDITION" id="GFAGie01-controller_basico_9" name="motion" type="MOTION_PROGRAM"/>

```

Fig. 3-2 probe XML Files

Shows probe XML of Agent when it is checked in the XML file editor. In the diagram, the part enclosed by □ is the id attribute value of DataItem node.

3.2.2.6. AddVariable method

Adds a variable object to CaoController. Only the variable names shown in 3.4.1 can be used.

AddVariable is specified as follows.

SYNOPSIS

AddVariable

```

(
    "<variable name>", // Variable name
    "<Option>" // Option character string (optional)
)

```

3.2.2.7. Execute method

Execute ConController extension. Execute is specified as follows.

SYNOPSIS

Controller Execute

```
(
    "<extended command name>",           // Extended command name
    Argument                             // Argument (by command) )
)
```

For details on the extended command list that can be specified with Execute, see section 3.3. The usage examples are described in detail in the extended commands.

3.2.3. CaoExtension classes

3.2.3.1. GetVariableNames Properties

Gets the list of connected variable names. You can use the variable name retrieved by this property as the first argument of the 3.2.2.6 AddVariable method.

SYNOPSIS

GetVariableNames

```
(
    "<Option>"                          // Option character string (not used)
)
```

Usage example

```
// Get variable name list
String[] variableNames = extension.GetVariableNames("");
```

3.2.3.2. Variables Properties

Gets the variable collection held by the expansion board.

Usage example

```
// Variable Collection Retrieval
ORiN2.ManagedCAO.CCaoVariables variables = extension.Variables;
// Variable acquisition
ORiN2.ManagedCAO.CCaoVariable variable = variables[0];
```

3.2.3.3. AddVariable method

Adds a variable object to CaoExtension. Only the variable names shown in 3.4.2 can be used.

AddVariable is specified as follows.

SYNOPSIS

AddVariable

```
(
    "<variable name>",           // Variable name
    "<Option>"                  // Option character string (optional)
)
```

3.2.4. CaoVariable classes

3.2.4.1. Value Properties

Gets/Sets the value. The behavior depends on the variable name. For details, refer to section 3.4, Variable list.

3.3. Command list

This is a list of the extended commands that can be used in CaoController class.

Command	Description	See Also
GetDeviceNames	Gets the list of devices.	P.18
GetTypeNames	Retrieves the list of data types that exist in the specified device.	P.19
GetDataItemIDs	Retrieves the ID of DataItem belonging to the specified device and of the specified data type.	P.19

3.3.1.1. GetDeviceNames

Gets the list of device names that exist in the connected Agent. The acquired device name can be used by GetTypeNames command or GetDataItemIDs command.

Item	Type Description	
Argument	None	
Return Value	VT_BSTR VT_ARRAY	List of device names.

Usage example

```
// Get device name list
String[] deviceNames = controller.Execute("GetDeviceNames") as string[];
```

3.3.1.2. GetTypeNames

Retrieves the list of data types that exist in the specified device. The retrieved data type name can be used by GetDataItemsIDs command.

Item	Type Description	
Argument	VT_BSTR	Target device name
Return Value	VT_BSTR VT_ARRAY	Data Type List

Usage example

// Get device name list

```
String[] deviceNames = controller.Execute("GetDeviceNames", "") as string[];
If (deviceNames != null)
{
    Foreach (string deviceName in deviceNames)
    {
        String[] typeNames = controller.Execute("GetTypeNames", deviceName);
    }
}
```

3.3.1.3. GetDataItemIDs

Retrieves the ID of DataItem belonging to the specified device and of the specified data type. The identity of the retrieved DataItem can be used in CaoController.AddExtension method.

Data Type

Item	Type Description	
Argument	VT_BSTR VT_ARRAY	
	0	Target device name Uppercase and lowercase letters are distinguished.
	1	Name of the target data type. Uppercase and lowercase letters are distinguished.
Return Value	VT_BSTR VT_ARRAY	List of DataItem IDs that meet the criteria

Argument

There are three combinations of arguments, and the target DataItem nodes are different.

- Both omitted.

Retrieves an identity listing of all DataItem nodes present in Agent.

Argument	None
----------	------

- Specify the device name

Retrieves an identity listing for all DataItem nodes that are less than or equal to Device node with the specified name properties.

Argument	VT_BSTR	Device name . Case is case sensitive.
----------	---------	---------------------------------------

3. Specify the device name and data type.

Retrieves the identity lists of all DataItem nodes that have type attribute of the specified data type, below Device node with name attribute specified by the device name.

Argument	VT_BSTR VT_ARRAY	
	0	Target device name Uppercase and lowercase letters are distinguished.
	1	Name of the target data type. Uppercase and lowercase letters are distinguished.

Usage example

```
// Examples of obtaining @type="POSITION" DataItem ID for devices
String[] deviceNames = controller.Execute("GetDeviceNames", "") as string[];
If (deviceNames != null)
{
    Foreach (string deviceName in deviceNames)
    {
        String[] dataItemIDs = controller.Execute("GetDataItemIDs",
            New string[2](deviceName, "POSITION"));
    }
}
```

3.4. Variable list

Defines a list of variables that can be used in each class. A variable points to an object of CaoVariable class.

3.4.1. CaoController class-variable

Variable name	NAME
@Version	Get the version of the provider as a string. (*. *. *)
@LoadingTime	Providers get the last read duration of current XML.
@UpdateDateTime	Gets the last date and time the provider read current XML.

3.4.1.1. @LoadingTime

Providers get the last read duration of current XML in milliseconds. No value acquisition error occurs during disconnection.

Data Type

Data Type	Description
VT_UI4	Duration (in milliseconds) required to read current XML

3.4.1.2. @UpdateDateTime

Gets the last date and time the provider read current XML. No value acquisition error occurs during disconnection.

Data Type

Data Type	Description
VT_DATE	Date and time current XML was read

3.4.2. CaoExtension class-variable

Variables in CaoExtension class have two types of variables: static variables whose values do not change at all times, and dynamic variables whose values change at the timing of retrieval. Static variables are taken from DataItem node of probe XML, and dynamic variables are taken from the node with the corresponding dataItemId of current XML.

If the variable name is specified in any character string, the value of the attribute specified by the option is acquired.

Variable name	Description	Advanced
Dynamic variable		
@Value	Gets Text of nodes.	P.22
@TimeStamp	Gets the value of timestamp attribute.	P.23
@Sequence	Gets the value of sequence attribute.	P.23
@ElemName	Get the node name.	P.23
@DynamicValues	Gets the values of dynamic variables at once.	P.23
@Current_RawData	Get attributes and values at once.	P.24
Static variable		
@Category	Gets the value of category attribute.	P.25
@Type	Gets the value of type attribute.	P.25
@SubType	Gets the value of subType attribute.	P.26
@Units	Gets the value of Units attribute.	P.26
@NativeUnits	Gets the value of nativeUnits attribute.	P.26

Variable name	Description	Advanced
@Representation	Gets the value of Representation attribute.	P.26
@SampleRate	Gets the value of sampleRate attribute.	P.27
@NativeScale	Gets the value of nativeScale attribute.	P.27
@Statistic	Gets the value of statistic attribute.	P.27
@Probe_RawData	Get attributes and values at once.	P.27
User variable		
<Arbitrary string>	Gets the value of the specified attribute.	P. 28

3.4.2.1. Dynamic variable

The figure below shows the correspondence between dynamic variables and the properties of Current XML of Agent. Dynamic variables can only be retrieved while connected to the server.

Device: GFAGie01; UUID: mtc_adapter001

Device : GFAGie01

Events

Timestamp	Type	Sub Type	Name	Id	Sequence	Value
2020-03-16T01:00:04.166964	Availability		avail	GFAGie01-dtop_1	328653571	UNAVAILABLE
2020-03-16T01:00:04.165930	EmergencyStop		estop	GFAGie01-dtop_2	328653536	UNAVAILABLE
2020-03-16T01:00:04.166765	AssetChanged			GFAGie01_asset_chg	328653563	UNAVAILABLE
2020-03-16T01:00:04.166856	AssetRemoved			GFAGie01_asset_rem	328653568	UNAVAILABLE

Condition

Timestamp	Type	Sub Type	Name	Id	Sequence	Value
2020-03-16T01:00:04.166027	Unavailable		system	GFAGie01-dtop_3	328653540	

Rotary : A

Samples

Fig. 3-2 Relationship between Dynamic-Variables and Current Nodes

An orange border indicates the correspondence to the variable, and a blue border indicates the correspondence to CaoExtension.Name property.

3.4.2.1.1. @Value

Retrieves Text of nodes with matching dataItemId properties as a string. What is retrieved depends on DataItem.

If Text does not exist, it is an empty character.

Data Type

Data Type	Description
VT_BSTR	Text content is stored.

3.4.2.1.2. @TimeStamp

Retrieves the time at which the sampled data was reported to Agent, or the calculated time, as a string. This variable must contain a value.

Data Type

Data Type	Description
VT_BSTR	Contains the value of timestamp attribute.

3.4.2.1.3. @Sequence

Retrieves a string-numbered representation of the sequence of occurrences of a sample in the data buffer where the sampled data is Agent. This variable must contain a value.

Data Type

Data Type	Description
VT_BSTR	Contains the value of sequence attribute.

3.4.2.1.4. @ElemName

The element name of Current XML node whose dataItemId attribute matches is acquired as a character string. The data belonging to CONDITION category is the data whose element name indicates the current value. Use this variable to acquire the current value.

Data Type

Data Type	Description
VT_BSTR	Contains the element name of the node.

3.4.2.1.5. @DynamicValues

Gets the values of dynamic variables at once.

Data Type

Data Type	Description
-----------	-------------

Data Type	Description
VT_BSTR VT_ARRAY	This is an array that contains data for one node.
0	The @Value is stored.
1	The @TimeStamp is stored.
2	The @Sequence is stored.
3	The @ElemName is stored.

Usage example

```
// Add Variable
ORiN2.ManagedCAO.CCaoVariable variable = extension.AddVariable("@DynamicValues");

String[] typedDynamicValues = variable.Value as string[]
String value = typedDynamicValues[0];
String timestamp = typedDynamicValues[1];
String sequence = typedDynamicValues[2];
```

3.4.2.1.6. @Current_RawData

Get the attributes and values of the Current XML node corresponding to the CaoExtension.Name property at once.

データ型

データ型	説明
VT_VARIANT VT_ARRAY	It is an array that stores a list of attributes and attribute values for one node.
0	VT_BSTR VT_ARRAY List of attributes included in one node
n	Attribute name
1	VT_BSTR VT_ARRAY List of attribute values included in one node
n	Attribute value

使用例

```
// 変数追加
ORiN2.ManagedCAO.CCaoVariable variable = extension.AddVariable("@Current_RawData");

object[] currentRawDataValues = variable.Value as object[]
object[] currentRawValue = currentRawDataValues[0];
string[] attributeValue = currentRawValue[0];
string[] attributeValue = currentRawValue[1];
```

3.4.2.2. Static variable

Static variables are a set of variables whose values do not change, so it is recommended that they be retrieved

only once, for example, when adding a variable. Static variables can be retrieved while disconnecting from the server.

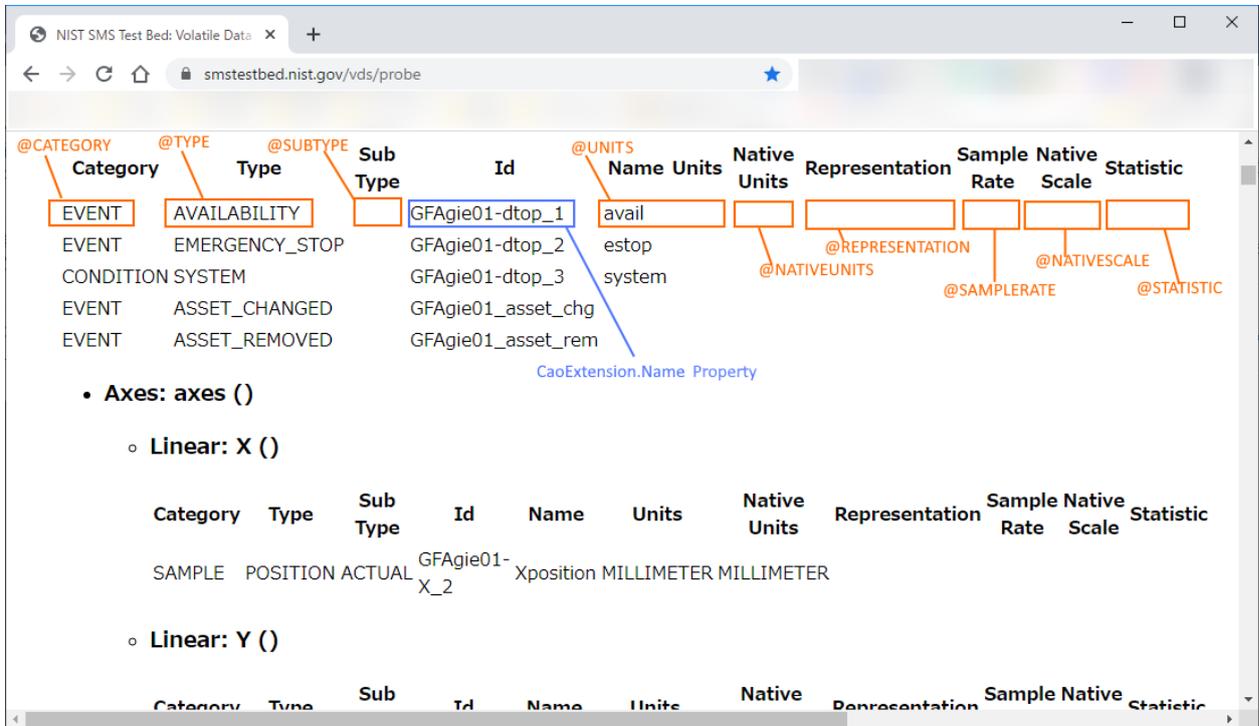


Fig. 3-3 Correspondence between Static Variables and Probe Nodes

An orange border indicates the correspondence to the variable, and a blue border indicates the correspondence to CaoExtension.Name property.

3.4.2.2.1. @Category

Gets the category of information as a string. Data in the same category has the same characteristics. Categories are always defined for all DataItem.

Data Type

Data Type	Description
VT_BSTR	Contains the value of category attribute. e.g. SAMPLE, CONDITION, EVENT

3.4.2.2.2. @Type

Gets the type of data to be measured as a string. The datatype is always defined for all DataItem.

Data Type

Data Type	Description
VT_BSTR	Contains the value of type attribute. e.g., POSITION, VELOCITY, ANGLE, BLOCK, ROTARY_VELOCITY

3.4.2.2.3. @SubType

Retrieves a subcategory in a data type as a string. A subcategory does not exist for all data types.

Data Type

Data Type	Description
VT_BSTR	Contains the value of subType attribute. If it is omitted, it is an empty character. e.g. POSITION, ACTUAL, COMMANDED

3.4.2.2.4. @Units

Gets the unit as a string. It is always present if the category is SAMPLE.

Data Type

Data Type	Description
VT_BSTR	Contains the value of units attribute. If it is omitted, it is an empty character. e.g. MILIMETER, DEGREE, PERCENT

3.4.2.2.5. @NativeUnits

Gets the native units of Component as a string. Native units do not exist on all DataItem.

Data Type

Data Type	Description
VT_BSTR	Contains the value of nativeUnits attribute. If it is omitted, it is an empty character. e.g. MILIMETER, DEGREE, PERCENT

3.4.2.2.6. @Representation

Gets the representation of the data as a string. If it is not retrieved, it is assumed to be VALUE.

Data Type

Data Type	Description
VT_BSTR	Contains the value of representation attribute. If it is omitted, an VALUE

Data Type	Description
	occurs. e.g.) VALUE, DISCRETE, MP3 of TIME_SERIES,

3.4.2.2.7. @SampleRate

Gets the number of samples per second as a string. The number of samples may be omitted.

Data Type

Data Type	Description
VT_BSTR	Contains the value of sampleRate attribute. If it is omitted, it is an empty character. Example) 0.1 if it is once every 10 seconds

3.4.2.2.8. @NativeScale

Gets the native unit multiplier as a string. The received data may be divided by this value before conversion. Native units may be omitted.

Data Type

Data Type	Description
VT_BSTR	Contains the value of nativeScale attribute. If it is omitted, it is an empty character.

3.4.2.2.9. @Statistic

Data calculated specifically for DataItem. This data may be omitted.

Data Type

Data Type	Description
VT_BSTR	Contains the value of statistic attribute. If it is omitted, it is an empty character. Example). AVERAGE, MINIMUM, MAXIMUM, RANGE etc.

3.4.2.2.10. @Probe_RawData

Get the attributes and values of the Probe XML node corresponding to the CaoExtension.Name property at once.

データ型

データ型	説明
VT_VARIANT VT_ARRAY	It is an array that stores a list of attributes and attribute values for one node.

データ型		説明
0	VT_BSTR VT_ARRAY	List of attributes included in one node
	n	Attribute name
1	VT_BSTR VT_ARRAY	List of attribute values included in one node
	n	Attribute value

使用例

// 変数追加

```
ORiN2.ManagedCAO.CCaoVariable variable = extension.AddVariable("@Probe_RawData");
```

```
object[] probeRawDataValues = variable.Value as object[]
object[] probeRawValue = probeRawDataValues[0];
string[] attributeNames = probeRawValue[0];
string[] attributeValues = probeRawValue[1];
```

3.4.2.3. User-specified variable

Acquires the value corresponding to the specified attribute value. Specify any variable name for the variable name.

オプション

オプション	必須	説明	
Attribute=	○	Specify the attribute name.	
Node=	-	Specify the node. (Optional)	
		値	内容
		0	Obtained from the Current XML node. (Default value)
		1	Obtained from the Probe XML node.

データ型

データ型	説明
VT_BSTR	The value of the attribute specified by Attribute is stored.

4. PROGRAMMING BY PROVIDERS

4.1. Sample Programming to Retrieve POSITIN and ANGLE

Sample programs that connect to Agent of Smart Manufacturing Systems (SMS) Test Bed and acquire POSITION and ANGLE data for GFAGie01 devices.

Table 4-1 Sample program requirements

Requirements	Description
Host	https://smstestbed.nist.gov/vds
Process Description	Gets the present value of POSITION data of GFAGie01 device.
	Gets the present value of ANGLE data of GFAGie01 device.

Specific codes are given in the following sections.

4.1.1. Sample program

The following is an overview of the sample program.

Sample	GetPositionAndAngleValues.cs
	<pre> // Object Private ORiN2.ManagedCAO.CCaoEngine m_caoEngine = null; Private ORiN2.ManagedCAO.CCaoWorkspace m_caoWorkspace = null; Private ORiN2.ManagedCAO.CCaoController m_caoController = null; Public void Main() { // Connection This.Connect(); List<ORiN2.ManagedCAO.CCaoExtension>positionExtensions = CreateExtensions(controller, "GFAGie01", "POSITION"); Foreach (var extension in positionExtensions) { ORiN2.ManagedCAO.CCaoVariable variable = extension.Variables["@DynamicValues"]; String[] typedValues = variable.Value as string[]; } List<ORiN2.ManagedCAO.CCaoExtension>angleExtensions = CreateExtensions(controller, "GFAGie01", "ANGLE"); Foreach (var extension in angleExtensions) { ORiN2.ManagedCAO.CCaoVariable variable = extension.Variables["@DynamicValues"]; </pre>

```

        String[] typedValues = variable.Value as string[];
    }

    // Disconnect
    This.Disconnect();
}

// Connection method
Private void Connect()
{
    // Generate CaoEngine object
    This.m_caoEngine = new ORiN2.ManagedCAO.CCaoEngine();
    // Generate CaoWorkspace object
    This.m_caoWorkspace = this.m_caoEngine.AddWorkspace("NewWrks", "");
    // Generate CaoController object
    This.m_caoController = workspace.AddController("MTConnect",
                                                "CaoProv.MTConnect",
                                                "",
                                                "server=https://smstestbed.nist.gov/vds");
}

// Disconnect method
Private void Disconnect()
{
    This.m_caoEngine.Dispose();
    This.m_caoEngine = null;
}

///<summary>
/// Creates a Extensions with the specified datatype for the specified device.
/// The created Extension has all the variables that you can add.
///</summary>
///<param name="controller">CaoController instance</ param>
///<param name="deviceName"><device-name</param>
///<param name="typeName">>typename</param>
///<returns>Created Extension listing</>
List<ORiN2.ManagedCAO.CCaoExtension>CreateExtensions(ORiN2.ManagedCAO.CCaoController
controller, string deviceName, string typeName)
{
    String[] param = new string[2] {deviceName, typeName};
    String[] positionExtensionNames =
        Controller.Execute("GetDataItemIDs", param) as string[];
    List<ORiN2.ManagedCAO.CCaoExtension>extensions =
        New List<ORiN2.ManagedCAO.CCaoExtension>();
    Foreach (string id in positionExtensionNames)
    {
        ORiN2.ManagedCAO.CCaoExtension extension = controller.AddExtension(id, "");
        String[] variableNames = extension.GetVariableNames("") as string[];
        Foreach (string variableName in variableNames)
        {
            Extension.AddVariable(variableName, "");
        }
        Extensions.Add(extension);
    }
}
}

```

```

    Return extensions;
}

```

4.1.1.1. Connection

To connect to Agent, perform the following steps.

- (1) Prepare a variable to hold the object. The objects required to connect to the controller are CaoEngine object, CaoWorkspace object, and CaoController object. CaoWorkspace object does not need to have a variable to obtain CaoController object from CaoWorkspaces. You will also need a CaoVariable for accessing the variable. The following is a code example for C#.

```

// Variables for CaoEngine
Private ORiN2.ManagedCAO.CCaoEngine m_caoEngine = null;
// Variables for CaoWorkspace
Private ORiN2.ManagedCAO.CCaoWorkspace m_caoWorkspace = null;
// Variables for CaoController
Private ORiN2.ManagedCAO.CCaoController m_caoController = null;

```

- (2) Creates a CaoEngine object. CaoEngine object is generated using the New keyword.

```

// Generate CaoEngine object
This.m_caoEngine = new ORiN2.ManagedCAO.CCaoEngine();

```

- (3) Gets or generates a CaoWorkspace object. When you create a CaoEngine object, it defaults to one CaoWorkspaces object and one object. The following is a sample code/default CaoWorkspace for creating a new CaoWorkspace.

```

// Generate CaoWorkspace object
This.m_caoWorkspace = this.m_caoEngine.AddWorkspace("NewWrks", "");

```

- (4) Creates a CaoController.

```

// Generate CaoController object
This.m_caoController = workspace.AddController("MTConnect",
                                             "CaoProv.MTConnect",
                                             "",
                                             "server=https://smstestbed.nist.gov/vds");

```

4.1.1.2. Retrieves POSITION and ANGLE data.

Generates a CaoExtension corresponding to a particular data type for a particular device and retrieves the value of the @DynamicValues variable for the created CaoExtension.

- (1) Generates CaoExtension corresponding to a particular data type for a particular device. In the example program, CreateExtensions method uses CaoController.GetDataItemIDs command to generate the corresponding CaoExtension.

```
String[] param = new string[2] { deviceName, typeName };
String[] positionExtensionNames =
    Controller.Execute("GetDataItemIDs", param) as string[];
List<ORiN2.ManagedCAO.CCaoExtension>extensions =
    New List<ORiN2.ManagedCAO.CCaoExtension>();
Foreach (string id in positionExtensionNames)
{
    ORiN2.ManagedCAO.CCaoExtension extension = controller.AddExtension(id, "");
    String[] variableNames = extension.GetVariableNames("") as string[];
    Foreach (string variableName in variableNames)
    {
        Extension.AddVariable(variableName, "");
    }
    Extensions.Add(extension);
}

```

- (2) Retrieves the value of the @DynamicValues variable in CaoExtension created by CreateExtensions method, and obtains the current value of POSITION data and ANGLE data of GFAGie01 device.

```
List<ORiN2.ManagedCAO.CCaoExtension>positionExtensions =
CreateExtensions(controller, "GFAGie01", "POSITION");
Foreach (var extension in positionExtensions)
{
    ORiN2.ManagedCAO.CCaoVariable variable =
        Extension.Variables["@DynamicValues"];
    Object[] values = variable.Value as object[];
    Foreach (object value in values)
    {
        String[] typedValues = value as string[];
    }
}
List<ORiN2.ManagedCAO.CCaoExtension>angleExtensions =
    CreateExtensions(controller, "GFAGie01", "ANGLE");
Foreach (var extension in angleExtensions)
{
    ORiN2.ManagedCAO.CCaoVariable variable =
        Extension.Variables["@DynamicValues"];
    Object[] values = variable.Value as object[];
    Foreach (object value in values)
    {
        String[] typedValues = value as string[];
    }
}

```

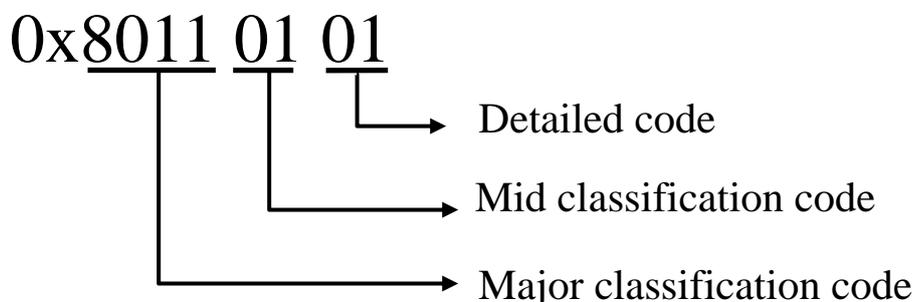
4.1.1.3. Disconnect

To disconnect from the controller, you can erase the generated objects and delete the objects that you want to erase from the collection class that manages the objects. However, you do not need to explicitly remove it if you use ORiN.ManagedCAO. The following is a code example.

```
// Remove all objects from CaoEngine  
This.m_caoEngine.Dispose();  
// Clear CaoEngine  
This.m_caoEngine = null;
```

5. Provider error code

This provider has its own error code masked with 0x8011**** to 0x8017****.



Major classification code

Major classification code	Description
8011	This error occurred during CaoWorkspace.AddController.
8012	This error occurred during CaoController.AddExtension.
8013	This error occurred during CaoController.AddVariable,CaoExtension.AddVariable.
8014	This error occurred when acquiring CaoVariable.Value property.
8015	CaoController.GetVariableNames,CaoExtension.GetVariableNames run-time failure.
8016	CaoController.GetExtensionNames run-time failure.
8017	Errors during CaoController.Execute.

Mid classification code

Mid classification code	Description
00	There is no classification.
01	Errors related to optional specifications and arguments.
02	Probe XML file-related error.
03	Current XML file-related error.
04	Error with the specified name.

Detailed code

Detailed code	Description
00	There is no detailed code.
01	The XML file could not be read. Check the networking and Agent status.
02	An DataItem exists that does not have the 'id' attribute defined.
03	There was a Device for which no name was defined.

Detailed code	Description
04	There was a DataItem for which no category was defined.
05	There was a DataItem for which no type was defined.
06	There was a Current node with no sequence defined.
07	There was a Current node with no timestamp defined.
08	Exclusive control failed.
09	Required options/arguments are not present.
0A	The value is out of range.
0B	Memory out of range.

Unique Error Codes

Error Number	Description
0x801110xx ²	An optional failure occurred during AddController. Review the string specified in the option.
0x801120xx ²	Error parsing Probe XML file during AddController.
0x801130xx ²	Error parsing Current XML file during AddController.
0x80124000	The name specified in AddExtension is incorrect. The ID of DataItem node that does not exist was specified. Do not specify anything other than the ID of GetExtensionNames node that can be acquired by OOC method.
0x80130400	The name specified in AddVariable is incorrect. Specify the variable name described in the user's guide.
0x8014000A	Specify a larger value for Timeout options in AddController where the exclusive control failed when you retrieve the value of the dynamic variable.
0x801403xx ²	Dynamic variables could not be retrieved because Current XML parsing failed.
0x801500xx ²	GetVariableNames run-time failure.
0x801600xx ²	GetExtensionNames run-time failure.
0x801701xx ²	There is a problem with the arguments when executing the command.

² Refer to the detailed error code.