

MTConnect プロバイダ ユーザーズ ガイド

Version 1.0.2

June 27, 2023

備考：

© 2018 DENSO WAVE INCORPORATED

この取扱説明書の著作権は、株式会社デンソーウェーブにあります。

本書に掲載されている会社名や製品は、一般に各社の商標または登録商標です。

仕様は予告なく変更することがあります。

【改版履歴】

バージョン	日付	内容
1.0.0	2020-11-04	初版.
1.0.1	2022-07-14	@Current_RawData 変数, @Probe_RawData 変数, ユーザー任意指定変数の追加
1.0.2	2023-06-27	特定条件で一時ファイルが作成され続ける問題を修正.

【動作確認 Agent】

Agent URL	バージョン	注意事項
https://smstestbed.nist.gov/vds/	Ver1.3.1	Smart Manufacturing Systems (SMS) Test Bed です. 開発時期とはデータが異なる可能性があります.

この取扱説明書の一部または全部を無断で複製・転載することはお断りします.

- この説明書の内容は将来予告なしに変更することがあります.
- 本書の内容については, 万全を期して作成いたしましたが, 万が一不審の点や誤り, 記載もれなど, お気づきの点がありましたらご連絡ください.
- 運用した結果の影響については, 上項にかかわらず責任を負いかねますのでご了承ください.

目次

1. はじめに.....	6
1.1. プロバイダのデータアクセス.....	7
2. アプリケーション開発のための環境セットアップ.....	8
2.1. PC 開発環境のセットアップ.....	8
2.1.1. プロバイダの自動インストール.....	8
3. コマンドリファレンス.....	9
3.1. メソッド/プロパティ一覧.....	9
3.2. メソッド・プロパティ.....	9
3.2.1. CaoWorkspace クラス.....	9
3.2.1.1. AddController メソッド.....	9
3.2.2. CaoController クラス.....	10
3.2.2.1. GetExtensionNames メソッド.....	10
3.2.2.2. Extensions プロパティ.....	12
3.2.2.3. GetVariableNames メソッド.....	12
3.2.2.4. Variables プロパティ.....	12
3.2.2.5. AddExtension メソッド.....	12
3.2.2.6. AddVariable メソッド.....	15
3.2.2.7. Execute メソッド.....	16
3.2.3. CaoExtension クラス.....	16
3.2.3.1. GetVariableNames プロパティ.....	16
3.2.3.2. Variables プロパティ.....	16
3.2.3.3. AddVariable メソッド.....	17
3.2.4. CaoVariable クラス.....	17
3.2.4.1. Value プロパティ.....	17
3.3. 拡張コマンド一覧.....	17
3.3.1.1. GetDeviceNames.....	17
3.3.1.2. GetTypeNames.....	18
3.3.1.3. GetDataItemIDs.....	18
3.4. 変数一覧.....	19
3.4.1. CaoController クラス変数.....	19
3.4.1.1. @LoadingTime.....	20
3.4.1.2. @UpdateDateTime.....	20

3.4.2. CaoExtension クラス変数	20
3.4.2.1. 動的変数.....	21
3.4.2.2. 静的変数.....	24
3.4.2.3. ユーザー任意指定変数.....	27
4. プロバイダによるプログラミング	29
4.1. POSITIN と ANGLE データを取得するサンプルプログラミング	29
4.1.1. サンプルプログラム	29
4.1.1.1. 接続	31
4.1.1.2. POSITION データと ANGLE データを取得する.	32
4.1.1.3. 切断	33
5. プロバイダエラーコード.....	34

1. はじめに

本書は、MTConnect の Agent を介して各アダプターが収集したデータにアクセスするプロバイダのユーザーズガイドです。図 1-1 はプロバイダが想定するシステム構成図になります。以降本プロバイダを単にプロバイダと呼称します。

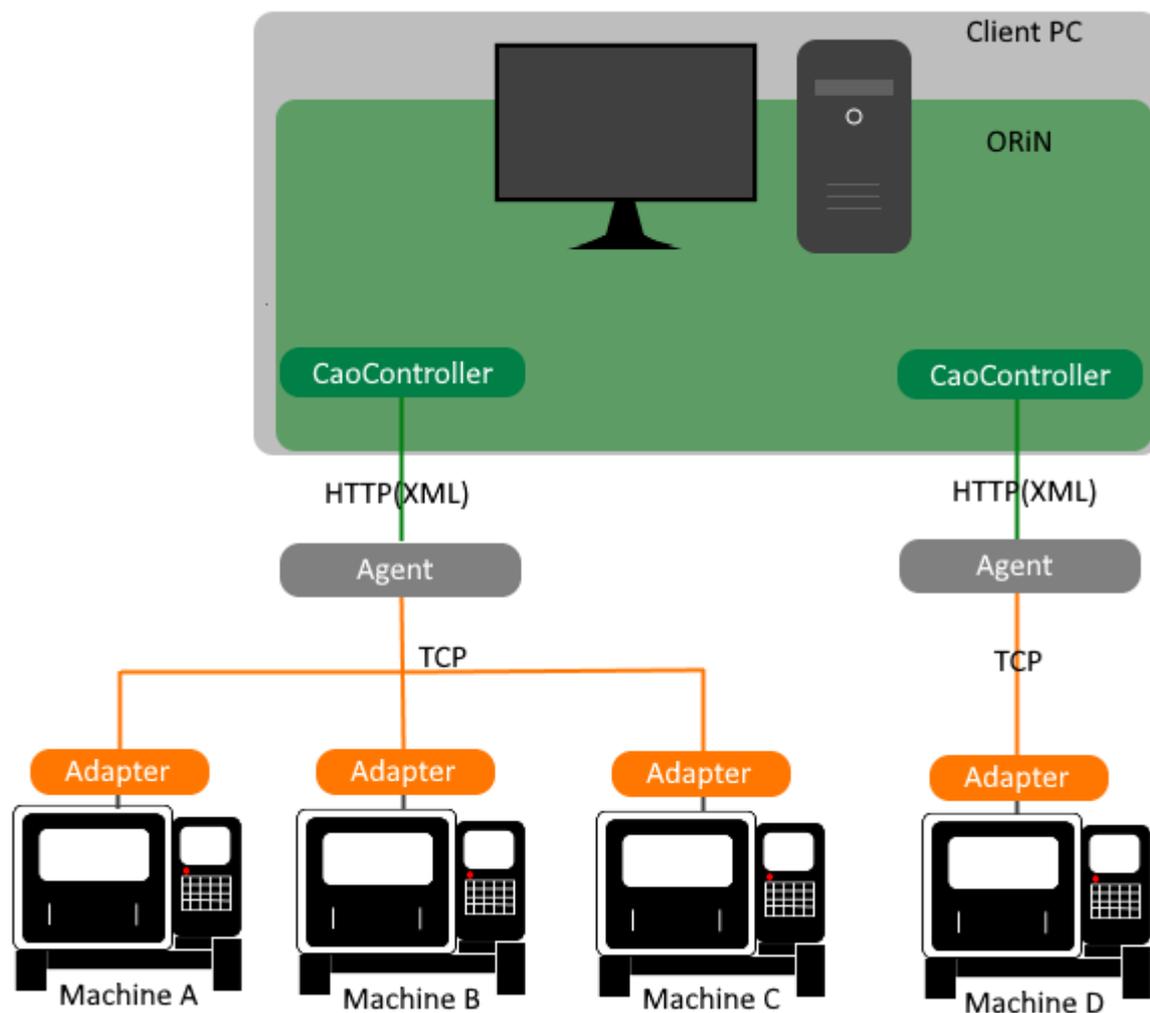


図 1-1 構成図

プロバイダは、CaoController オブジェクトを通して MTConnect の Agent と HTTP で接続します。Agent は最低 1 つの Adapter からのデータが格納されています。

また、プロバイダのオブジェクトと MTConnect の要素の対応を図 1-2 に表します。
(※一例です。全てを表しているわけではありません。)

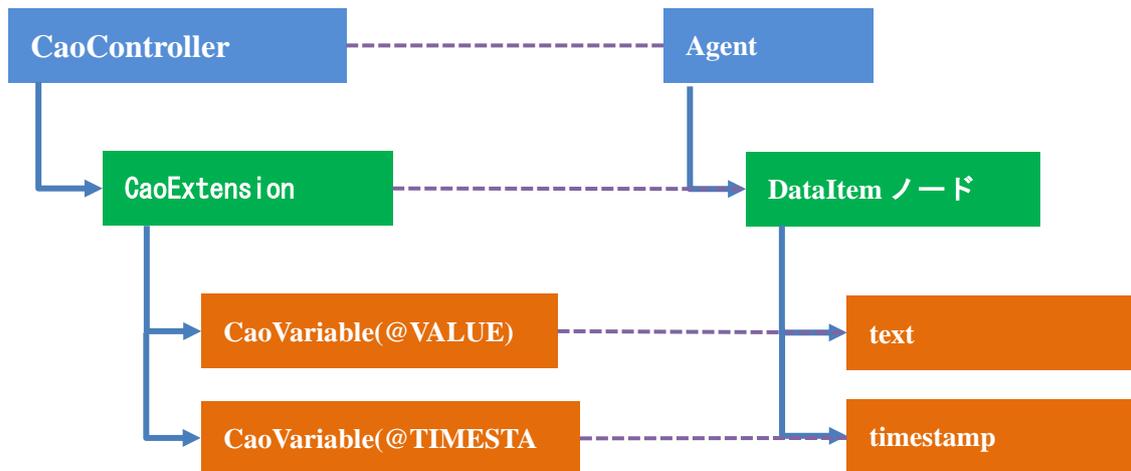


図 1-2 プロバイダのオブジェクトと MTConnect 要素との対応図

CaoExtension が Agent の DataItem ノードに対応します。CaoExtension は DataItem ノードの id 属性を指定して作成することができます。CaoExtension 下位層の CaoVariable は、CaoExtension の Name プロパティと同じ DataItemId 属性を持つ current XML 内のノードの各属性や、Text に対応しています。

1.1. プロバイダのデータアクセス

プロバイダは Agent への接続に成功すると自動でワークスレッドを起動し、ワークスレッド上で Agent に対して周期的に current データを取得し続けます。プロバイダのデータはそのワークスレッドが取得した値にアクセスしているため、必ずしも Agent の現在値を表していないことに注意してください。

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. PC 開発環境のセットアップ

2.1.1. プロバイダの自動インストール

ORiN2 SDK がインストールされている環境であれば、プロバイダを使用する準備は完了です。ただし、開発環境のセットアップは別途、Component Object Model をサポートするプログラミング環境をご準備ください。

3. コマンドリファレンス

3.1. メソッド/プロパティ一覧

表 3-1 メソッド/プロパティ一覧

カテゴリ	メソッド/プロパティ ¹	機能	参照
CaoWorkspace			
	AddController	M コントローラに接続	P. 9
CaoController			
	GetExtensionNames	M 接続可能な拡張ボード名リストの取得	P. 10
	Extensions	P コントローラが保持する拡張ボードコレクションの取得	P. 12
	GetVariableNames	M 接続可能な変数名リストの取得	P. 12
	Variables	P コントローラが保持する変数コレクションの取得	P. 12
	AddExtension	M 拡張ボードオブジェクトの追加	P. 12
	AddVariable	M 変数オブジェクトの追加	P. 15
	Execute	M 拡張コマンドの実行	P. 16
CaoExtension			
	GetVariableNames	M 接続可能な変数名リストの取得	P. 16
	Variables	P タスクが保持する変数コレクションの取得	P. 16
	AddVariable	M 変数オブジェクトの追加	P. 17
CaoVariable			
	Value	P 値の取得/設定	P. 17

3.2. メソッド・プロパティ

3.2.1. CaoWorkspace クラス

3.2.1.1. AddController メソッド

CaoWorkspace に、コントローラオブジェクトを追加します。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 Agent 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

(

¹ M:メソッド, P:プロパティ, E:イベントをそれぞれ示します。

```

"<コントローラ名>", // コントローラ名(任意)
"CaoProv. MTConnect", // プロバイダ名(固定)
"<マシン名>", // プロバイダ実行マシン名(任意)
"<オプション>" // オプション文字列(省略可能)
)

```

オプション

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。オプション文字列は下記に示す各オプションをカンマ(,)でつなげた文字列となります。

オプション	必須	説明	値範囲
Server	○	接続先のサーバー名を指定してください。 (例). http://192.168.0.1	-
Timeout	-	値取得時のタイムアウト(ms)を指定してください。 CaoVariable::Value アクセス時にこのタイムアウトをオーバーするとエラーとなります。 デフォルト値は 500ms です。	0 以上
Sampling	-	ワークスレッドのサンプリング周期を指定してください。デフォルト値は 500ms です。	0 以上

使用例

```

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

```

3.2.2. CaoController クラス

3.2.2.1. GetExtensionNames メソッド

AddExtension メソッドで使用できる拡張ボード名のリストを取得します。プロバイダで使用できる拡張ボード名は Agent 存在する DataItem ノードの id 属性の値です。

書式

GetExtensionNames

```
(
    "<オプション>" // オプション文字列
)
```

オプション

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。オプション文字列は下記に示す各オプションをカンマ(,)でつなげた文字列となります。

オプション	必須	説明
Device	-	Device ノードの name 属性を指定します。 Device オプションを指定すると、指定された Device ノード内に限定して DataItem ノードの ID リストを取得します。 大文字小文字は区別されます。
Type	-	DataItem ノードの type 属性を指定します。 Type オプションを指定すると、データ種別を限定した DataItem ノードの ID リストを取得します。 Device オプションを指定されていた場合のみ意味を持ちます。Device オプションを省略していた場合は、Type オプションの指定は無視されます。 大文字小文字は区別されます。

オプションの組み合わせは3パターンありそれぞれ対象となる DataItem ノードが変化します。

1. どちらも省略

Agent に存在するすべての DataItem ノードの ID リストを取得します。

2. Device オプションを指定

指定された name 属性をもつ Device ノード以下の、すべての DataItem ノードの ID リストを取得します。

3. Device オプションと Type オプションを指定

デバイス名で指定された name 属性をもつ Device ノード以下の、指定されたデータ種別を type 属性でもつ、すべての DataItem ノードの ID リストを取得します。

使用例

```
// 引数省略
```

```
string[] extensionNames1 = controller.GetExtensionNames("");
```

```
// デバイス名を指定
```

```
string[] extensionNames2 = controller.GetExtensionNames("Device = GFAGie01");
```

```
// デバイス名とデータ種別を指定
```

```
string[] extensionNames3 = controller.GetExtensionNames("Device = GFAgie01, Type = POSITION");
```

3.2.2.2. Extensions プロパティ

コントローラが保持する、拡張ボードコレクションを取得します。

使用例

```
// 拡張ボードコレクション取得
```

```
ORiN2.ManagedCA0.CCaoExtensions extensions = controller.Extensions;
```

```
// 拡張ボード取得
```

```
ORiN2.ManagedCA0.CCaoExtension extension = extensions[0];
```

3.2.2.3. GetVariableNames メソッド

接続可能な変数名リストを取得します。本プロパティで取得した変数名は、後述する AddVariable メソッドの第一引数に使用することができます。

書式

GetVariableNames

```
(  
    "＜オプション＞" // オプション文字列(使用しません)  
)
```

使用例

```
// 変数名リスト取得
```

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

3.2.2.4. Variables プロパティ

コントローラが保持する、変数コレクションを取得します。

使用例

```
// 変数コレクション取得
```

```
ORiN2.ManagedCA0.CCaoVariables variables = controller.Variables;
```

```
// 変数取得
```

```
ORiN2.ManagedCA0.CCaoVariable variable = variables[0];
```

3.2.2.5. AddExtension メソッド

CaoController に、拡張ボードオブジェクトを追加します。以下に、AddExtension メソッドの仕様を示します。

書式**AddExtension**

```
(
    "<拡張ボード名>", // DataItem ノードの id 属性の値を指定してください.
    "<オプション>" // オプション文字列(使用しません.)
)
```

オプション

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。オプション文字列は下記に示す各オプションをカンマ(,)でつなげた文字列となります。

オプション	必須	説明	値範囲
Array	-	<p>動的変数の値 (P. 21) を配列型で取得するか否かを指定します。</p> <p>Array オプションに False を指定した際は、AddExtension で指定した DataItem の id 属性と同一の id を持つデータが Current に複数あった場合でも、先頭の 1 つのみを取得します。</p> <p>True を指定した際は、データ数にかかわらず常に配列型でデータを取得します。</p> <p>デフォルト値は False です。</p>	<p>True</p> <p>False</p>

使用例

```
// 拡張ボードの追加
```

```
ORiN2.ManagedCA0.CCaoExtension extension = controller.AddExtension("GFAgie01-X_2", "");
```

3.2.2.5.1. 拡張ボード名の確認方法

拡張ボード名には、Agent の probe XML に定義されている DataItem ノードの id 属性値を指定することができます。id 属性値は GetExtensionNames メソッドで取得するかブラウザ上で確認することができます。

ブラウザ上での確認方法

URL 入力欄に、「AddController の Server オプション + "/probe"」を入力すれば確認することができます。

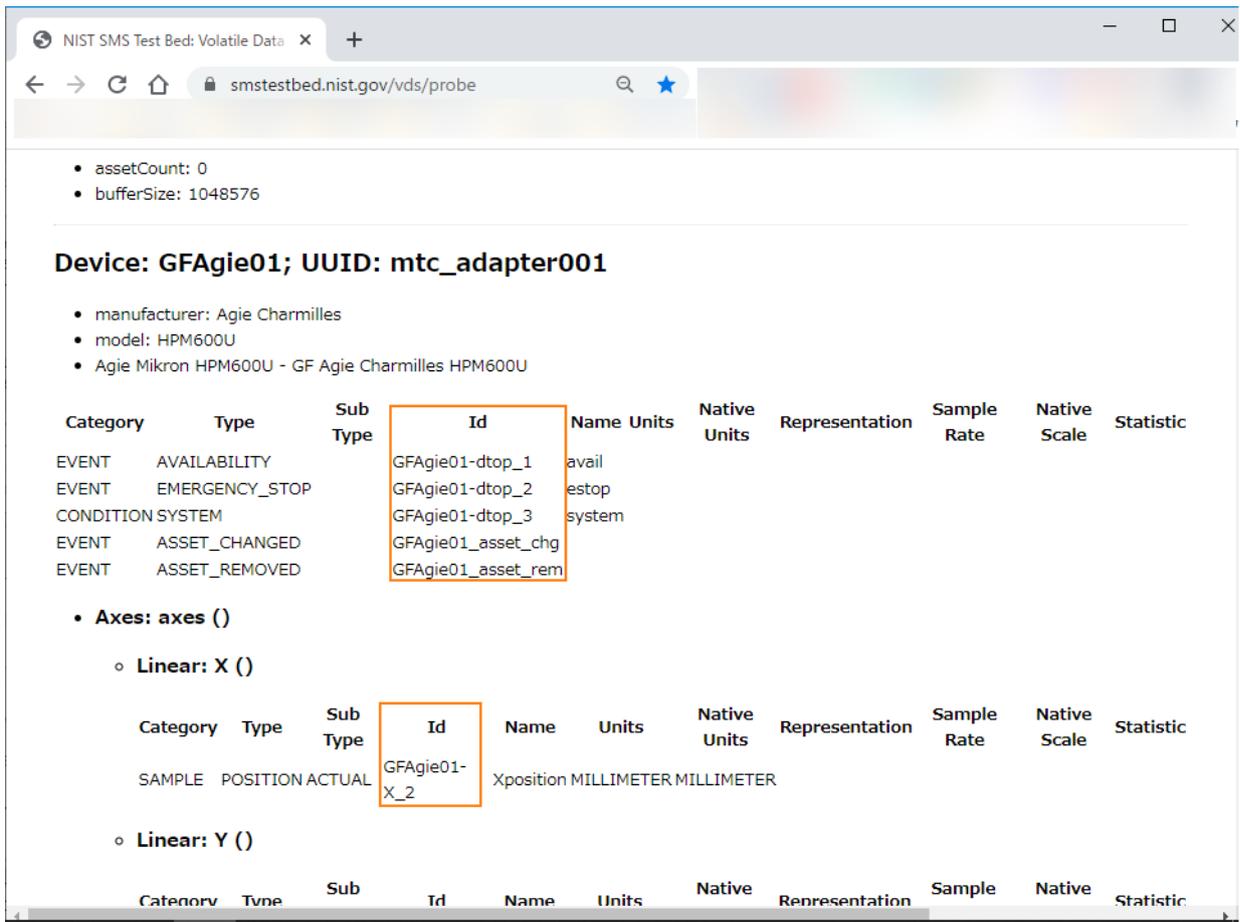


図 3-1 ブラウザ上での probe

Agent の probe XML をブラウザ上で確認した時の様子を示しています。DataItem ノードの id 属性値は図中の口で囲まれた部分に定義されています。

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/xsl" href="/styles/Devices.xsl"?>
3 <MTConnectDevices xmlns="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="astop" 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_ren" 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 <DataItem category="SAMPLE" id="GFAgie01-X_2" name="Xposition" nativeUnits="MILLIMETER" subType="ACTUAL" type="POSITION" units="MILLIMETER"/>
20 </DataItem>
21 </Linear>
22 <Linear id="GFAgie01-Y_1" name="Y">
23 <DataItem category="SAMPLE" id="GFAgie01-Y_2" name="Yposition" nativeUnits="MILLIMETER" subType="ACTUAL" type="POSITION" units="MILLIMETER"/>
24 </DataItem>
25 </Linear>
26 <Linear id="GFAgie01-Z_1" name="Z">
27 <DataItem category="SAMPLE" id="GFAgie01-Z_2" name="Zposition" nativeUnits="MILLIMETER" subType="ACTUAL" type="POSITION" units="MILLIMETER"/>
28 </DataItem>
29 </Linear>
30 <Rotary id="GFAgie01-C_1" name="C">
31 <DataItem category="SAMPLE" id="GFAgie01-C_2" name="Cposition" nativeUnits="DEGREE" subType="ACTUAL" type="ANGLE" units="DEGREE"/>
32 </DataItem>
33 </Rotary>
34 <Rotary id="GFAgie01-A_1" name="A">
35 <DataItem category="SAMPLE" id="GFAgie01-A_2" name="Aposition" nativeUnits="DEGREE" subType="ACTUAL" type="ANGLE" units="DEGREE"/>
36 </DataItem>
37 </Rotary>
38 </Components>
39 </Axes>
40 <Controller id="GFAgie01-controller_basic_1" name="controller_basic">
41 <DataItem category="EVENT" id="GFAgie01-controller_basic_2" name="Fovr" nativeUnits="PERCENT" subType="OVERRIDE" type="PATH_FEEDRATE_OVERRIDE" units="PERCENT"/>
42 <DataItem category="EVENT" id="GFAgie01-controller_basic_3" name="Sovr" nativeUnits="PERCENT" type="ROTARY_VELOCITY_OVERRIDE" units="PERCENT"/>
43 <DataItem category="CONDITION" id="GFAgie01-controller_basic_4" name="servo" type="ACTUATOR"/>
44 <DataItem category="CONDITION" id="GFAgie01-controller_basic_5" name="comms" type="COMMUNICATIONS"/>
45 <DataItem category="CONDITION" id="GFAgie01-controller_basic_6" name="pneumatic" type="COMMUNICATIONS"/>
46 <DataItem category="CONDITION" id="GFAgie01-controller_basic_7" name="hydraulic" type="COMMUNICATIONS"/>
47 <DataItem category="CONDITION" id="GFAgie01-controller_basic_8" name="logic" type="LOGIC_PROGRAM"/>
48 <DataItem category="CONDITION" id="GFAgie01-controller_basic_9" name="motion" type="MOTION_PROGRAM"/>
49 </Controller>
50 </Components>
51 </Devices>
52 </MTConnectDevices>
53 </XML>
54 </XML>

```

図 3-2 probe XML ファイル

Agent の probe XML を XML ファイルエディタで確認した時の様子を示しています。図中で□で囲まれた部分が DataItem ノードの id 属性値です。

3.2.2.6. AddVariable メソッド

CaoController に変数オブジェクトを追加します。変数名には 3.4.1 に示すもののみ使用できます。

以下に、AddVariable の仕様を示します。

書式

AddVariable

(

"<変数名>", // 変数名

"<オプション>" // オプション文字列(省略可能)

)

3.2.2.7. Execute メソッド

CaoController の拡張コマンドを実行します。以下に、Execute の仕様を示します。

書式

Controller Execute

```
(  
    "<拡張コマンド名>",           // 拡張コマンド名  
    引数                           // 引数(コマンドにより異なります。)  
)
```

Execute で指定できる拡張コマンド一覧の詳細は 3.3 章を参照してください。使用例は拡張コマンドの詳細で記述しています。

3.2.3. CaoExtension クラス

3.2.3.1. GetVariableNames プロパティ

接続可能な変数名リストを取得します。本プロパティで取得した変数名は、3.2.2.6 AddVariable メソッドの第一引数に使用することができます。

書式

GetVariableNames

```
(  
    "<オプション>"                // オプション文字列(使用しません)  
)
```

使用例

```
// 変数名リスト取得  
string[] variableNames = extension.GetVariableNames("");
```

3.2.3.2. Variables プロパティ

拡張ボードが保持する、変数コレクションを取得します。

使用例

```
// 変数コレクション取得  
ORiN2.ManagedCA0.CCaoVariables variables = extension.Variables;  
// 変数取得  
ORiN2.ManagedCA0.CCaoVariable variable = variables[0];
```

3.2.3.3. AddVariable メソッド

CaoExtension に変数オブジェクトを追加します。変数名には 3.4.2 に示すもののみ使用できます。以下に、AddVariable の仕様を示します。

書式

AddVariable

```
(
    "<変数名>",           // 変数名
    "<オプション>"       // オプション文字列(省略可能)
)
```

3.2.4. CaoVariable クラス

3.2.4.1. Value プロパティ

値を取得/設定します。変数名によって動作が異なります。詳細は、3.4.変数一覧を参照してください。

3.3. 拡張コマンド一覧

CaoController クラスで使用できる拡張コマンド一覧を列挙します。

コマンド	説明	参照
GetDeviceNames	デバイス一覧を取得します。	P. 17
GetTypeNames	指定したデバイスに存在するデータ種別一覧を取得します。	P. 18
GetDataItemIDs	指定したデバイスに属し、指定したデータ種別の DataItem の ID を取得します。	P. 18

3.3.1.1. GetDeviceNames

接続した Agent に存在するデバイス名一覧を取得します。取得したデバイス名は、GetTypeNames コマンド、GetDataItemIDs コマンドで使用することができます。

項目	型説明
引数	なし
戻り値	VT_BSTR VT_ARRAY デバイス名一覧。

使用例

```
// デバイス名一覧を取得
string[] deviceNames = controller.Execute("GetDeviceNames") as string[];
```

3.3.1.2. GetTypeNames

指定したデバイスに存在するデータ種別一覧を取得します。取得したデータ種別名は、GetDataItemsIDs コマンドで使用することができます。

項目	型説明	
引数	VT_BSTR	対象のデバイス名
戻り値	VT_BSTR VT_ARRAY	データ種別一覧

使用例

// デバイス名一覧を取得

```
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

指定したデバイスに属し、指定したデータ種別の DataItem の ID を取得します。取得した DataItem の ID は、CaoController.AddExtension メソッドで使用することができます。

データ型

項目	型説明	
引数	VT_BSTR VT_ARRAY	
	0	対象のデバイス名。 大文字、小文字は区別されます。
	1	対象のデータ種別名。 大文字、小文字は区別されます。
戻り値	VT_BSTR VT_ARRAY	条件に合う DataItem の ID 一覧

引数

引数の組み合わせが3パターンあり、それぞれ対象となる DataItem ノードが異なります。

1. どちらも省略

Agent に存在するすべての DataItem ノードの ID リストを取得します。

引数	なし
----	----

2. デバイス名を指定

指定された name 属性をもつ Device ノード以下の、すべての DataItem ノードの ID リストを取得します。

引数	VT_BSTR	デバイス名。大文字小文字は区別されます。
----	---------	----------------------

3. デバイス名とデータ種別を指定

デバイス名で指定された name 属性をもつ Device ノード以下の、指定されたデータ種別を type 属性をもつ、すべての DataItem ノードの ID リストを取得します。

引数	VT_BSTR VT_ARRAY	
	0	対象のデバイス名。 大文字、小文字は区別されます。
	1	対象のデータ種別名。 大文字、小文字は区別されます。

使用例

```
// 各デバイスの@type="POSITION"のDataItem IDを取得する例
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. 変数一覧

各クラスで使用可能な変数一覧を定義します。変数は、CaoVariable クラスのオブジェクトを指します。

3.4.1. CaoController クラス変数

変数名	名前
@Version	プロバイダのバージョンを文字列で取得します。(*.*.*)
@LoadingTime	プロバイダが current XML の直近の読み込み所要時間を取得します。

@UpdateDateTime	プロバイダが current XML を読み込んだ直近の日時を取得します。
-----------------	---------------------------------------

3.4.1.1. @LoadingTime

プロバイダが current XML の直近の読み込み所要時間をミリ秒単位で取得します。切断中でも値取得はエラーになりません。

データ型

データ型	説明
VT_UI4	current XML の読み込みに要した時間(ミリ秒)

3.4.1.2. @UpdateDateTime

プロバイダが current XML を読み込んだ直近の日時を取得します。切断中でも値取得はエラーになりません。

データ型

データ型	説明
VT_DATE	current XML の読み込んだ日時

3.4.2. CaoExtension クラス変数

CaoExtension クラスの変数には常に値が変わらない静的変数と、取得するタイミングで値が変わる動変数の2種類の変数があります。静的変数は probe XML の DataItem ノードから取得し、動変数は current XML の対応する dataItemId を持つノードから取得します。

任意の文字列で変数名を指定した場合は、オプションにより指定した属性の値を取得します。

変数名	説明	詳細
動変数		
@Value	ノードの Text を取得します。	P. 22
@TimeStamp	timestamp 属性の値を取得します。	P. 22
@Sequence	sequence 属性の値を取得します。	P. 22
@ElemName	ノード名を取得します。	P. 22
@DynamicValues	動変数の値をまとめて取得します。	P. 23
@Current_RawData	属性と値をまとめて取得します。	P. 23
静的変数		
@Category	category 属性の値を取得します。	P. 24

変数名	説明	詳細
@Type	type 属性の値を取得します。	P. 25
@SubType	subType 属性の値を取得します。	P. 25
@Units	Units 属性の値を取得します。	P. 25
@NativeUnits	nativeUnits 属性の値を取得します。	P. 25
@Representation	Representation 属性の値を取得します。	P. 26
@SampleRate	sampleRate 属性の値を取得します。	P. 26
@NativeScale	nativeScale 属性の値を取得します。	P. 26
@Statistic	statistic 属性の値を取得します。	P. 26
@Probe_RawData	属性と値をまとめて取得します。	P. 27
ユーザー変数		
<任意の文字列>	指定した属性の値を取得します。	P. 27

3.4.2.1. 動的変数

下図は動的変数と、Agent の Current XML ノードの属性との対応を示しています。動的変数はサーバーと接続している状態でのみ値取得が可能です。

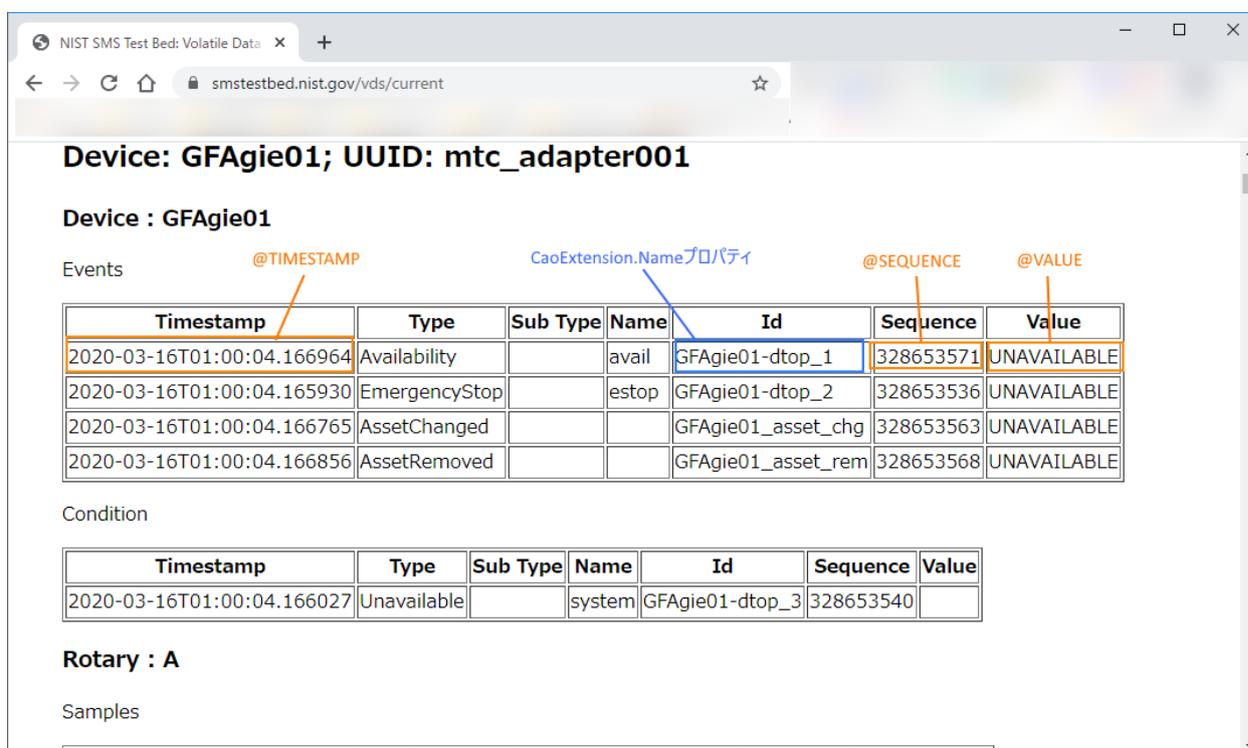


図 3-3 動的変数と Current ノードとの対応

オレンジ色の枠線は変数との対応を、青色の枠線は CaoExtension.Name プロパティとの対応を示し

ています。

3.4.2.1.1. @Value

dataItemId 属性が一致するノードの Text を文字列で取得します。どのようなデータが取得されるかは DataItem により異なります。ノードの Text が存在しない場合は空文字となります。

データ型

データ型	説明
VT_BSTR	Text 内容が格納されます。

3.4.2.1.2. @TimeStamp

サンプリングデータが Agent に報告された時間、もしくは計算された時間を文字列で取得します。この変数には必ず値が入っています。

データ型

データ型	説明
VT_BSTR	timestamp 属性の値が格納されます。

3.4.2.1.3. @Sequence

サンプリングデータが Agent のデータバッファ内のサンプルの発生の連続位置を表す番号を文字列で取得します。この変数には必ず値が入っています。

データ型

データ型	説明
VT_BSTR	sequence 属性の値が格納されます。

3.4.2.1.4. @ElemName

dataItemId 属性が一致する Curren XML ノードの要素名を文字列で取得します。CONDITION カテゴリに属するデータは要素名が現在の値を示すデータとなりますので現在値を取得する場合は本変数を使用してください。

データ型

データ型	説明
VT_BSTR	ノードの要素名が格納されます。

3.4.2.1.5. @DynamicValues

動的変数の値をまとめて取得します。

データ型

データ型	説明
VT_BSTR VT_ARRAY	1 ノード分のデータが格納されている配列です。
0	@Value の値が格納されます。
1	@TimeStamp の値が格納されます。
2	@Sequence の値が格納されます。
3	@ElemName の値が格納されます。

使用例

```
// 変数追加
```

```
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

CaoExtension.Name プロパティに対応する Current XML ノードの属性と値をまとめて取得します。

データ型

データ型	説明
VT_VARIANT VT_ARRAY	1 ノード分の属性と属性値の一覧が格納されている配列です。
0	VT_BSTR VT_ARRAY
	n
	属性名
1	VT_BSTR VT_ARRAY
	n
	属性値

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[] attributeNames = currentRawValue[0];
```

```
string[] attributeValues = currentRawValue[1];
```

3.4.2.2. 静的変数

静的変数は、値が変わることがない変数群のため、変数追加時などに一度だけ取得することをお勧めします。静的変数はサーバーと切断中でも値取得が可能です。

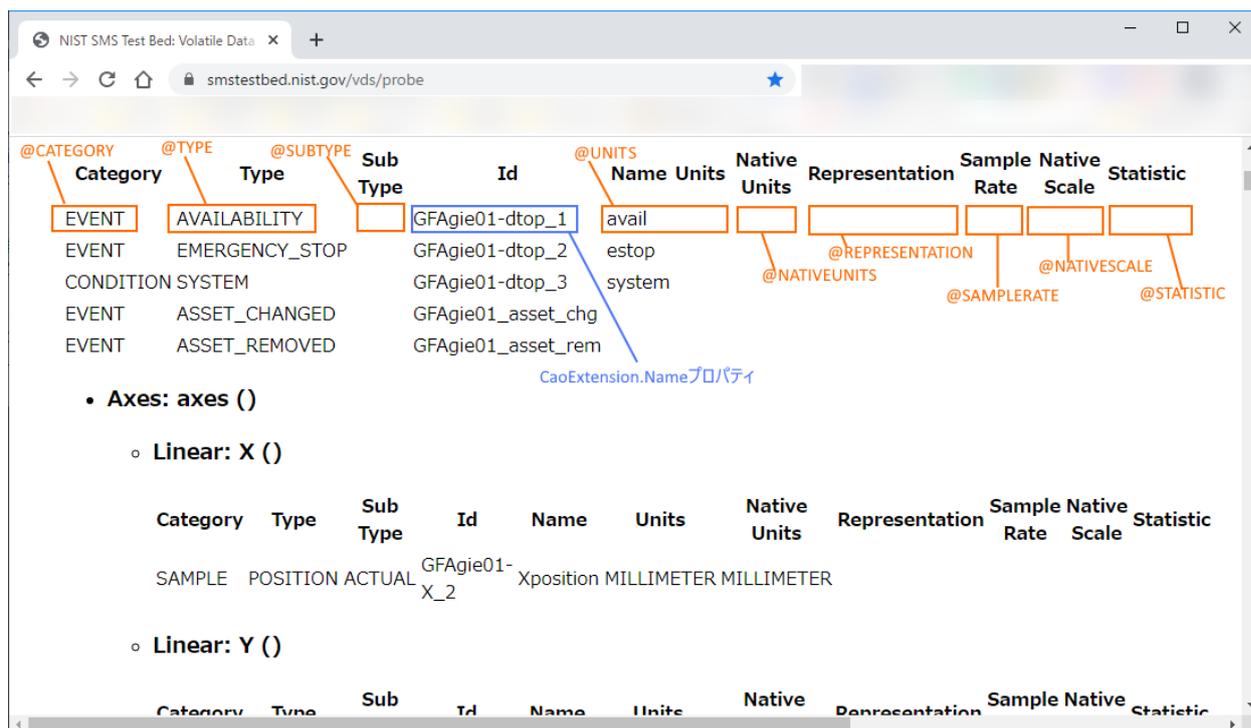


図 3-4 静的変数と Probe ノードとの対応

オレンジ色の枠線は変数との対応を、青色の枠線は CaoExtension.Name プロパティとの対応を示しています。

3.4.2.2.1. @Category

情報のカテゴリを文字列で取得します。同じカテゴリのデータであれば同様の特性を持っています。カテゴリはすべての DataItem に必ず定義されています。

データ型

データ型	説明
VT_BSTR	category 属性の値が格納されます。 例) SAMPLE, CONDITION, EVENT

3.4.2.2.2. @Type

測定されるデータ種別を文字列で取得します。データ種別はすべての DataItem に必ず定義されています。

データ型

データ型	説明
VT_BSTR	type 属性の値が格納されます。 例) POSITION, VELOCITY, ANGLE, BLOCK, ROTARY_VELOCITY

3.4.2.2.3. @SubType

データ種別内のサブカテゴリを文字列で取得します。サブカテゴリはすべてのデータ種別に存在するわけではありません。

データ型

データ型	説明
VT_BSTR	subType 属性の値が格納されます。省略されていた場合は空文字となります。 例) POSITION, ACTUAL, COMMANDED

3.4.2.2.4. @Units

単位を文字列で取得します。カテゴリが SAMPLE である場合、必ず存在します。

データ型

データ型	説明
VT_BSTR	units 属性の値が格納されます。省略されていた場合は空文字となります。 例) MILIMETER, DEGREE, PERCENT

3.4.2.2.5. @NativeUnits

Component のネイティブ単位を文字列で取得します。ネイティブ単位はすべての DataItem に存在するわけではありません。

データ型

データ型	説明
VT_BSTR	nativeUnits 属性の値が格納されます。省略されていた場合は空文字となります。

データ型	説明
	例) MILIMETER, DEGREE, PERCENT

3.4.2.2.6. @Representation

データの表現方法を文字列で取得します。値が取得されない場合は VALUE であると判断します。

データ型

データ型	説明
VT_BSTR	representation 属性の値が格納されます。省略されていた場合は VALUE となります。 例) VALUE, TIME_SERIES, DISCRETE, MP3

3.4.2.2.7. @SampleRate

1 秒当たりのサンプル数を文字列で取得します。サンプル数は省略されていることがあります。

データ型

データ型	説明
VT_BSTR	sampleRate 属性の値が格納されます。省略されていた場合は空文字となります。 例) 10 秒に 1 回であれば 0.1

3.4.2.2.8. @NativeScale

ネイティブ単位の乗数を文字列で取得します。受信したデータは変換前にこの値で除算される場合があります。ネイティブ単位は省略されていることがあります。

データ型

データ型	説明
VT_BSTR	nativeScale 属性の値が格納されます。省略されていた場合は空文字となります。

3.4.2.2.9. @Statistic

DataItem 固有に計算されたデータ。本データは省略されていることがあります。

データ型

データ型	説明
VT_BSTR	statistic 属性の値が格納されます。省略されていた場合は空文字と

データ型	説明
	なります。 例). AVERAGE, MINIMUM, MAXIMUM, RANGE etc.

3.4.2.2.10. @Probe_RawData

CaoExtension.Name プロパティに対応する Probe XML ノードの属性と値をまとめて取得します。

データ型

データ型	説明
VT_VARIANT VT_ARRAY	指定した ID の属性と属性値一覧が格納されている配列です。
0	VT_BSTR VT_ARRAY
n	1 ノードに含まれている属性一覧
1	VT_BSTR VT_ARRAY
n	1 ノードに含まれている属性値一覧

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[] attributeValue = probeRawValue[1];
```

3.4.2.3. ユーザー任意指定変数

指定した属性値に対応する値を取得します。変数名は任意の変数名を指定してください。

オプション

オプション	必須	説明	
Attribute=	○	属性名を指定します。	
Node=	—	ノードを指定します。(省略可)	
		値	内容
		0	Current XML ノードから取得します。(デフォルト値)
		1	Probe XML ノードから取得します。

データ型

データ型	説明
VT_BSTR	Attribute で指定した属性の値が格納されます。

4. プロバイダによるプログラミング

4.1. POSITIN と ANGLE データを取得するサンプルプログラミング

Smart Manufacturing Systems (SMS) Test Bed の Agent に接続し、GFAgie01 デバイスの POSITION データと ANGLE データを取得するサンプルプログラムを例にします。

表 4-1 サンプルプログラムの要件

要件	説明
接続先	https://smstestbed.nist.gov/vds
処理内容	GFAgie01 デバイスの POSITION データの現在値を取得する。
	GFAgie01 デバイスの ANGLE データの現在値を取得する。

以降の節から具体的なコードを示します。

4.1.1. サンプルプログラム

以下にサンプルプログラムの全体像を示します。

Sample	GetPositionAndAngleValues.cs
	<pre>// オブジェクト private ORiN2.ManagedCAO.CCaoEngine m_caoEngine = null; private ORiN2.ManagedCAO.CCaoWorkspace m_caoWorkspace = null; private ORiN2.ManagedCAO.CCaoController m_caoController = null; public void Main() { // 接続 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) {</pre>

```
        ORiN2.ManagedCAO.CCaoVariable variable = extension.Variables["@DynamicValues"];
        string[] typedValues = variable.Value as string[];
    }

    // 切断
    this.Disconnect();
}

// 接続メソッド
private void Connect()
{
    // CaoEngineオブジェクトの生成
    this.m_caoEngine = new ORiN2.ManagedCAO.CCaoEngine();
    // CaoWorkspaceオブジェクトの生成
    this.m_caoWorkspace = this.m_caoEngine.AddWorkspace("NewWrks", "");
    // CaoControllerオブジェクトの生成
    this.m_caoController = workspace.AddController("MTCConnect",
                                                "CaoProv.MTCConnect",
                                                "",
                                                "server=https://smstestbed.nist.gov/vds");
}

// 切断メソッド
private void Disconnect()
{
    this.m_caoEngine.Dispose();
    this.m_caoEngine = null;
}

/// <summary>
/// 指定されたデバイスの指定されたデータ型をもつExtensionsを作成します。
/// 作成されたExtensionには追加できるすべての変数を追加しています。
/// </summary>
/// <param name="controller">CaoControllerインスタンス</param>
/// <param name="deviceName">デバイス名称</param>
/// <param name="typeName">タイプ名称</param>
/// <returns>作成されたExtensionリスト</returns>
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. 接続

Agent と接続するためには、以下の手順を取ります。

- (1) オブジェクトを保持するための変数を用意します。コントローラ接続に必要なオブジェクトは、GaoEngineオブジェクトとGaoWorkspaceオブジェクトとGaoControllerオブジェクトです。GaoWorkspaceオブジェクトは、GaoControllerオブジェクトをCaoWorkspacesから取得する場合には変数を用意する必要はありません。また変数にアクセスするためのCaoVariableオブジェクトも必要になります。以下にC#でのコード例を示します。

```
// CaoEngine オブジェクト用の変数
private ORiN2.ManagedCAO.CGaoEngine m_caoEngine = null;
// CaoWorkspace オブジェクト用の変数
private ORiN2.ManagedCAO.CGaoWorkspace m_caoWorkspace = null;
// CaoController オブジェクト用の変数
private ORiN2.ManagedCAO.CGaoController m_caoController = null;
```

- (2) GaoEngineオブジェクトを生成します。GaoEngineオブジェクトはNewキーワードを使って生成します。

```
// CaoEngine オブジェクトの生成
this.m_caoEngine = new ORiN2.ManagedCAO.CGaoEngine();
```

- (3) GaoWorkspaceオブジェクトを取得もしくは生成します。GaoEngineオブジェクトを生成すると、デフォルトでCaoWorkspacesオブジェクトとCaoWorkspaceオブジェクトを1つずつ生成しています。以下にCaoWorkspaceオブジェクトを新しく生成するコード例とデフォルトのCaoWorkspaceを示します。

```
// CaoWorkspace オブジェクトの生成
this.m_caoWorkspace = this.m_caoEngine.AddWorkspace("NewWrks", "");
```

- (4) GaoControllerオブジェクトを生成します。

```
// CaoController オブジェクトの生成
```

```
this.m_caoController = workspace.AddController("MTConnect",
                                             "CaoProv.MTConnect",
                                             "",
                                             "server=https://smstestbed.nist.gov/vds");
```

4.1.1.2. POSITION データと ANGLE データを取得する.

特定のデバイスの特定のデータ種別に対応する CaoExtension を生成し、作成された CaoExtension の @DynamicValues 変数の値を取得します。

- (1) 特定のデバイスの特定のデータ種別に対応する CaoExtension を生成します。サンプルプログラムでは、CreateExtensions メソッドで、CaoController.GetDataItemIDs コマンドを使用して対応する CaoExtension を生成しています。

```
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) CreateExtensions メソッドで作成された CaoExtension の、@DynamicValues 変数の値を取得し、GFAgie01 デバイスの POSITION データと ANGLE データの現在値を取得しています。

```
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[];
        }
    }
}
```

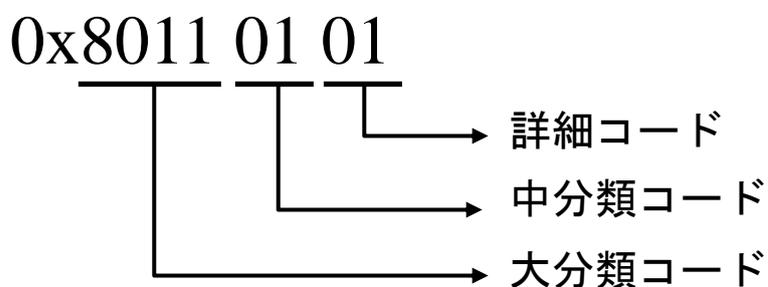
4.1.1.3. 切断

コントローラと切断する場合には、生成したオブジェクトを消去すると共に、オブジェクトを管理するコレクションクラスから消去するオブジェクトを削除します。ただし、ORiN.ManagedCAO を使用した場合は明示的に削除する必要はありません。以下にコード例を示します。

```
// CaoEngine からすべてのオブジェクトを削除
this.m_caoEngine.Dispose();
// CaoEngine の消去
this.m_caoEngine = null;
```

5. プロバイダエラーコード

本プロバイダには、0x8011**** ~ 0x8017****でマスクした独自エラーコードが存在します。



大分類コード

大分類コード	説明
8011	CaoWorkspace.AddController時に発生したエラーです。
8012	CaoController.AddExtension時に発生したエラーです。
8013	CaoController.AddVariable, CaoExtension.AddVariable時に発生したエラーです。
8014	CaoVariable.Valueプロパティの値取得時に発生したエラーです。
8015	CaoController.GetVariableNames, CaoExtension.GetVariableNames実行時エラーです。
8016	CaoController.GetExtensionNames実行時エラーです。
8017	CaoController.Execute実行時のエラーです。

中分類コード

中分類コード	説明
00	分類がありません。
01	オプション指定や引数に関するエラーです。
02	Probe XMLファイルに関するエラーです。
03	Current XMLファイルに関するエラーです。
04	指定された名前に関するエラーです。

詳細コード

詳細コード	説明
00	詳細コードはありません。
01	XMLファイルが読み込めませんでした。ネットワーク, およびAgentの状態を確認してください。

詳細コード	説明
02	id属性が定義されていないDataItemが存在しました。
03	name属性が定義されていないDeviceが存在しました。
04	category属性が定義されていないDataItemが存在しました。
05	type属性が定義されていないDataItemが存在しました。
06	sequence属性が定義されていないCurrentノードが存在しました。
07	timestamp属性が定義されていないCurrentノードが存在しました。
08	排他制御に失敗しました。
09	必須オプション/引数がしていません。
0A	値が範囲外です。
0B	メモリ範囲外です。

独自エラーコード

エラー番号	説明
0x801110xx ²	AddController時にオプションに関するエラーが発生しました。オプションで指定した文字列を見直してください。
0x801120xx ²	AddController時にProbe XMLファイルの解析に関するエラーが発生しました。
0x801130xx ²	AddController時、Current XMLファイルの解析に関するエラーが発生しました。
0x80124000	AddExtensionで指定した名前が正しくありません。存在しないDataItemノードのIDが指定されました。GetExtensionNamesメソッドで取得できるDataItemノードのID以外は指定しないでください。
0x80130400	AddVariableで指定した名前が正しくありません。ユーザーズガイド記載の変数名を指定してください。
0x8014000A	動的変数の値取得時に排他制御に失敗しました。AddControllerのTimeoutオプションにより大きな値を指定してください。
0x801403xx ²	Current XMLの解析に失敗したことが原因で動的変数の値取得が行えませんでした。
0x801500xx ²	GetVariableNames実行時エラーです。
0x801600xx ²	GetExtensionNames実行時エラーです。
0x801701xx ²	コマンド実行時の引数に問題があります。

² 詳細エラーコードを参照してください。