

# OPC UA Multiple プロバイダ OPC UA 用ゲートウェイ

Version 1.1.0

## ユーザーズ ガイド

July 1, 2024

備考:

**【改版履歴】**

| バージョン | 日付         | 内容            |
|-------|------------|---------------|
| 1.0.0 | 2022-07-29 | 初版.           |
| 1.1.0 | 2024-07-01 | 内部使用ライブラリの更新. |
|       |            |               |
|       |            |               |
|       |            |               |
|       |            |               |
|       |            |               |
|       |            |               |
|       |            |               |
|       |            |               |
|       |            |               |
|       |            |               |
|       |            |               |
|       |            |               |

**【対応機器】**

| 機種 | バージョン | 注意事項 |
|----|-------|------|
|    |       |      |
|    |       |      |
|    |       |      |
|    |       |      |
|    |       |      |
|    |       |      |
|    |       |      |
|    |       |      |
|    |       |      |
|    |       |      |
|    |       |      |
|    |       |      |
|    |       |      |
|    |       |      |

## 目次

|  |    |
|--|----|
| 1. はじめに .....  | 4  |
| 2. プロバイダの概要 .....  | 5  |
| 2.1. 概要 .....  | 5  |
| 2.2. メソッド・プロパティ .....                                    | 7  |
| 2.2.1. CaoWorkspace::AddController メソッド .....            | 7  |
| 2.2.2. CaoController::AddVariable メソッド .....             | 10 |
| 2.2.2.1. ノード ID .....                                    | 11 |
| 2.2.3. CaoController::get_VariableNames プロパティ .....      | 12 |
| 2.2.4. CaoVariable::get_DateTime プロパティ .....             | 12 |
| 2.2.5. CaoVariable::get_Value プロパティ .....                | 12 |
| 3. コマンドリファレンス .....                                      | 13 |
| 3.1. Controller クラス .....                                | 13 |
| 3.1.1. CaoController::Execute("MultipleRead") コマンド ..... | 13 |
| 4. エラーコード .....  | 14 |
| 5. 証明書ファイルについて .....                                     | 15 |
| 5.1. アプリケーション証明書 .....                                   | 15 |
| 5.2. 証明書による認証の為の証明書 .....                                | 15 |
| 5.3. 証明書の作成 .....  | 16 |
| 6. サンプルプログラム .....                                       | 17 |
| 6.1. CaoOPCUA .....                                      | 17 |
| 6.1.1. アクセスパス使用 .....                                    | 17 |
| 6.1.2. ノード ID 使用 .....                                   | 20 |

## 1. はじめに

本書は CAO から OPC UA(OLE for Process Control Unified Architecture)サーバを介して、PLC (Programmable Logic Controller)にアクセスする手段を与える「OPC UA 接続用の CAO プロバイダ(一括送受信対応)」(以下、単に OPC UA Multiple プロバイダと呼ぶ) のユーザーズガイドです。

このプロバイダにより、CAO 対応ツールはロボットのみならず OPC UA サーバを持つ PLC や表示盤に対してもロボットと同様にアクセスすることができます。

本書で解説する OPC UA Multiple プロバイダは OPC UA サーバに対して安全に接続できるためのセキュリティ機能と接続されているデバイスが持っている情報へのアクセス機能を提供します。

## 2. プロバイダの概要

### 2.1. 概要

OPC UA Multiple プロバイダは、ORiN の CAO と接続する純正の ORiN プロバイダの一種でありながら OPC UA 規格に準じる OPC UA クライアントでもあります(図 2-1)。このプロバイダにより、OPC UA 規格に準じるサーバに対する接続を確立し、ロボット、PLC のみならず OPCUA に準じる様々なデバイスに対しても同様にアクセスすることが可能になります。

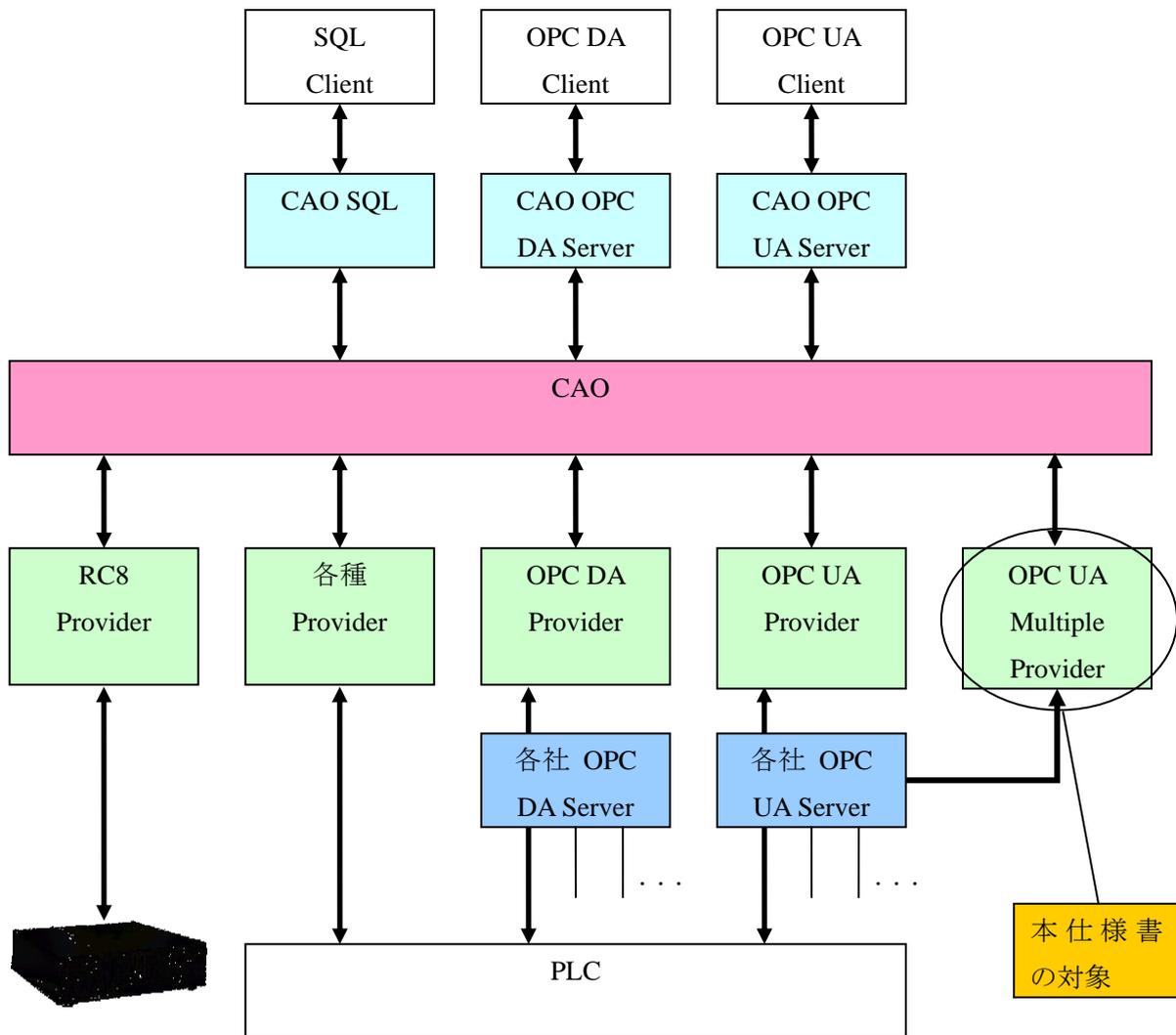


図 2-1 OPC UA 接続用の CAO プロバイダ

この OPC UA Multiple プロバイダは、OPC UA サーバが保持する Item の値を CaoVariable オブジェクトから参照することを可能としています。OPC DA プロバイダと OPC UA Multiple プロバイダの違いは、規格上の互換がないこと以外位置づけ的には同じです。

表 2-1 OPC UA Multiple プロバイダ

|                      |                                      |
|----------------------|--------------------------------------|
| ファイル名                | CaoProvOPCUAMultiple.dll             |
| ProgID               | CaoProv.OPCUAMultiple                |
| レジストリ登録 <sup>1</sup> | regsvr32 CaoProvOPCUAMultiple.dll    |
| レジストリ登録の抹消           | regsvr32 /u CaoProvOPCUAMultiple.dll |

<sup>1</sup> ORiN SDK でインストールした場合は手動で登録/抹消する必要はありません。

## 2.2. メソッド・プロパティ

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

CAO の CaoWorkspace クラスの AddController メソッドの仕様を下記に示します。

```

AddController
(
    "<Controller 名>",           // コントローラ名
    "CaoProv. OPCUAMultiple",    // プロバイダ名. 固定.
    "<マシン名>",               // プロバイダの実行マシン名.
    "<オプション>"              // オプション文字列 (OPC UA Multiple オプション)
)

```

ここで、<プロバイダ名>は固定、<実行マシン名>は他のプロバイダと同じ意味です。

<オプション>の書式は次のとおりです。

<オプション> ::= Server=<OPC UA サーバ URL>[,AccessPath=<アクセスパス>][,Security=<セキュリティポリシー>:<セキュリティモード>][,Der=<セキュリティ証明書ファイル名>][,Pem=<セキュリティプライベートキーファイル名>][,Password=<セキュリティパスワード>][,User=<ユーザ名>:<パスワード>][,Certificate=<ユーザ認証証明書ファイル名>][,PrivateKey=<ユーザ認証プライベートキーファイル名>][,TrustServer=<True|False>]

<OPC UA サーバ URL>の指定は必須で、OPC UA サーバの URL を入力します。これらの中で“[]”で囲まれたパラメータは必須ではなく、指定しなかった場合はデフォルト値が設定されます。設定した値の解釈は呼び出している OPC UA サーバに依存します。その他のオプションについては以下の表 2-2 を参照して下さい。



AddController(<bstrCtrlName:BSTR>,<bstrProvName>,  
<bstrPCName:BSTR>,<bstrOption:BSTR>))

|                |   |   |
|----------------|---|---|
| <bstrCtrlName> | : | [in] コントローラ名 (VT_BSTR)<br>任意の文字列を指定   |
| <bstrProvName> | : | [in] プロバイダ ProgID (VT_BSTR)<br>”CaoProv.OPCUAMultiple”の固定文字列を指定   |
| <bstrPCName>   | : | [in] PC 名 (VT_BSTR)<br>リモート接続する場合の PC 名を指定。通常のローカル接続の場合は空白文字列(“”)を指定。   |
| <bstrOption>   | : | [in] オプション文字列 (VT_BSTR)<br>接続を確実にするためのオプション文字列を指定。<br>各オプションはカンマ(,)区切りで次の書式で指定します。<br><OptionName>=<Value>,<OptionName>=<Value>,... |

例 : ”Server=opc.tcp://192.168.188.128:4890/CaoOPCUA,  
AccessPath=OPCUA.CAO/DataStore”

以下に<bstrOption>オプション文字列に指定するオプションリストを示す。

表 2-2 CaoWorkspace::AddController のオプション文字列

| オプション                                 | 意味   |  |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
|---------------------------------------|--|--|---|----|------|---|-----------|---------------|---|--|----------|---|--|----------------|---|-------------|---------------------|---|-----------|--------------------|---|------------|----|---|----|------|---|-----------|------|---|-------------------------|----------------|---|-----------------------|------------|-----------|------|------|---------------|------|---------------|----------------|----------|------|
| Server=<OPC UA サーバ URL>               | OPC UA サーバの URL. 【必須】<br>例 : “Server=opc.tcp://192.168.188.128:4890/CaoOPCUA”  |  |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| AccessPath=[アクセスパス]                   | デフォルトのアクセスパス.<br>AddVariable の AccessPath オプションで個別に設定することができます. 個別に設定しなかった場合はここで設定した値が使われます.   |  |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| Security=[セキュリティポリシー<br>[:セキュリティモード]] | <p>セキュリティの設定. 以下の値が指定できます. (デフォルト:0:0)</p> <p>セキュリティポリシー:</p> <table border="1"> <thead> <tr> <th>設定</th> <th>値</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>None</td> <td>0</td> <td>セキュリティなし.</td> </tr> <tr> <td>Basic128Rsa15</td> <td>1</td> <td>非推奨. この機能は互換性のために残されています.<sup>2</sup></td> </tr> <tr> <td>Basic256</td> <td>2</td> <td>非推奨. この機能は互換性のために残されています.<sup>2</sup></td> </tr> <tr> <td>Basic256Sha256</td> <td>3</td> <td>平均的なセキュリティ.</td> </tr> <tr> <td>Aes128Sha256RsaOaep</td> <td>4</td> <td>高いセキュリティ.</td> </tr> <tr> <td>Aes256Sha256RsaPss</td> <td>5</td> <td>最高のセキュリティ.</td> </tr> </tbody> </table> <p>セキュリティモード:</p> <table border="1"> <thead> <tr> <th>設定</th> <th>値</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>None</td> <td>0</td> <td>セキュリティなし.</td> </tr> <tr> <td>Sign</td> <td>1</td> <td>メッセージに署名は付けるが, 暗号化はしない.</td> </tr> <tr> <td>SignAndEncrypt</td> <td>2</td> <td>メッセージに署名を付け, かつ暗号化する.</td> </tr> </tbody> </table> <p>以下の組み合わせを設定することが可能です.</p> <table border="1"> <thead> <tr> <th>セキュリティポリシー</th> <th>セキュリティモード</th> </tr> </thead> <tbody> <tr> <td>None</td> <td>None</td> </tr> <tr> <td>Basic128Rsa15</td> <td>Sign</td> </tr> <tr> <td>Basic128Rsa15</td> <td>SignAndEncrypt</td> </tr> <tr> <td>Basic256</td> <td>Sign</td> </tr> </tbody> </table> | 設定                                     | 値 | 説明 | None | 0 | セキュリティなし. | Basic128Rsa15 | 1 | 非推奨. この機能は互換性のために残されています. <sup>2</sup> | Basic256 | 2 | 非推奨. この機能は互換性のために残されています. <sup>2</sup> | Basic256Sha256 | 3 | 平均的なセキュリティ. | Aes128Sha256RsaOaep | 4 | 高いセキュリティ. | Aes256Sha256RsaPss | 5 | 最高のセキュリティ. | 設定 | 値 | 説明 | None | 0 | セキュリティなし. | Sign | 1 | メッセージに署名は付けるが, 暗号化はしない. | SignAndEncrypt | 2 | メッセージに署名を付け, かつ暗号化する. | セキュリティポリシー | セキュリティモード | None | None | Basic128Rsa15 | Sign | Basic128Rsa15 | SignAndEncrypt | Basic256 | Sign |
| 設定                                    | 値  | 説明                                     |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| None                                  | 0  | セキュリティなし.                              |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| Basic128Rsa15                         | 1  | 非推奨. この機能は互換性のために残されています. <sup>2</sup> |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| Basic256                              | 2  | 非推奨. この機能は互換性のために残されています. <sup>2</sup> |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| Basic256Sha256                        | 3  | 平均的なセキュリティ.                            |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| Aes128Sha256RsaOaep                   | 4  | 高いセキュリティ.                              |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| Aes256Sha256RsaPss                    | 5  | 最高のセキュリティ.                             |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| 設定                                    | 値  | 説明                                     |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| None                                  | 0  | セキュリティなし.                              |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| Sign                                  | 1  | メッセージに署名は付けるが, 暗号化はしない.                |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| SignAndEncrypt                        | 2  | メッセージに署名を付け, かつ暗号化する.                  |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| セキュリティポリシー                            | セキュリティモード  |  |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| None                                  | None   |  |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| Basic128Rsa15                         | Sign   |  |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| Basic128Rsa15                         | SignAndEncrypt   |  |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |
| Basic256                              | Sign   |  |   |    |      |   |           |               |   |  |          |   |  |                |   |             |                     |   |           |                    |   |            |    |   |    |      |   |           |      |   |                         |                |   |                       |            |           |      |      |               |      |               |                |          |      |

<sup>2</sup> 暗号化アルゴリズムに脆弱性が発見されています. 使用は避けてください.

|                                 |  |                |
|---------------------------------|--|----------------|
|                                 | Basic256   | SignAndEncrypt |
|                                 | Basic256Sha256   | Sign           |
|                                 | Basic256Sha256   | SignAndEncrypt |
|                                 | Aes128Sha256RsaOaep  | Sign           |
|                                 | Aes128Sha256RsaOaep  | SignAndEncrypt |
|                                 | Aes256Sha256RsaPss   | Sign           |
|                                 | Aes256Sha256RsaPss   | SignAndEncrypt |
|                                 | セキュリティポリシーが 0 の場合、セキュリティモードは省略することができます。その場合セキュリティモードは 0 となります。                      |                |
| Der=[セキュリティ証明書ファイル名]            | セキュリティ用の証明書のファイル名を指定します。(デフォルト:空文字)  |                |
| Pem=[セキュリティプライベートキーファイル名]       | セキュリティ用のプライベートキーのファイル名を指定します。(デフォルト:空文字)   |                |
| Password=[セキュリティパスワード]          | セキュリティ用のパスワードを指定します。(デフォルト:空文字)  |                |
| User=[ユーザ名:パスワード]               | ユーザ認証を行う場合にユーザ名とパスワードを指定します。(デフォルト:空文字)  |                |
| Certificate=[ユーザ認証証明書ファイル名]     | 証明書による認証を行う場合に証明書のファイル名を指定します。(デフォルト:空文字)  |                |
| PrivateKey=[ユーザ認証プライベートキーファイル名] | 証明書による認証を行う場合にプライベートキーのファイル名を指定します。(デフォルト:空文字)                                       |                |
| TrustServer=[<True False>]      | サーバ証明書が証明書信頼リストに存在しない場合の動作を指定します。(デフォルト:True)<br>True:信頼し接続します。<br>False:信頼せず接続しません。 |                |

**使用例**

```
private CCaoController _caoController = null;

_caoController = _caoWorkspace.AddController("OpcUa", "CaoProv.OPCUAMultiple", "",
"Server=opc.tcp://192.168.188.128:4890/CaoOPCUA,AccessPath=OPCUA.CAO/DataStore");
```

## 2.2.2. CaoController::AddVariable メソッド

CAO の CaoController::AddVariable メソッドの仕様を下記に示します。

```
CaoController::AddVariable
(
    "<変数名>",           // 変数名
    "<オプション>"       // オプション文字列
)
```

OPC UA Multiple プロバイダを使う場合は、この引数を次のように設定します。

<変数名>::=<アイテム ID>

<オプション>::=[[AccessPath=<アクセスパス>],[RequestType=<受け取り変数型>]  
[,NamespaceIndex=<ネームスペースインデックス>],IdentifierType=<識別子タイプ>],Identifier=<識別子>]]

これらの中で“[]”で囲まれたパラメータは必須ではなく、指定しなかった場合はデフォルト値が設定されます。設定した値の解釈は呼び出している OPC UA サーバに依存します。受け取り変数型は VT\_TYPE を使用します。VT\_TYPE とその数値については表 2-4 を参照して下さい。その他のオプションについては以下の表 2-3 を参照して下さい。

使用可能なシステム変数については、2.2.3 を参照してください。



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

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

| OPC UA Multiple オプション名            | 説明   |   |                |
|-----------------------------------|--|---|----------------|
| AccessPath [=<アクセスパス>]            | アクセスパスの設定。(デフォルト:親コントローラの AccessPath オプション値)<br>AddController の AccessPath オプションで設定した値を上書きします。 |   |                |
| RequestType<br>[=<受け取り変数型>]       | 受け取り変数型の設定。(デフォルト:VT_EMPTY)  |   |                |
| NamespaceIndex [=<ネームスペースインデックス>] | サーバで定義されているノード ID の名前空間を指定します。(デフォルト:指定なし)<br>(参照 2.2.2.1)                                     |   |                |
| IdentifierType [=<識別子タイプ>]        | 識別子のデータ型を指定します。(デフォルト:指定なし)  |   |                |
|                                   | 設定   | 値 | 説明             |
|                                   | 数値   | 0 | 識別子を数値で指定します。  |
|                                   | 文字列  | 1 | 識別子を文字列で指定します。 |
|                                   | (参照 2.2.2.1)   |   |                |

|                     |   |
|---------------------|---|
| Identifier [=<識別子>] | ノードの識別子を指定します。(デフォルト: 指定なし)<br>(参照 2.2.2.1) |
|---------------------|---|

### 2.2.2.1. ノード ID

ネームスペースインデックス・識別子タイプ・識別子を指定した場合、ノード ID による指定を行います。ノード ID による指定を行う場合、これらの 3 種を全て指定しないとエラーとなります。

ノード ID による指定を行う場合、アイテム名とアクセスパスによる検索は行われません。

例)

```

▼ NodeId ns=5;s=Item_1
  NamespaceIndex 5
  IdentifierType String
  Identifier Item_1
  ...
  
```

OPCUA サーバ上で上記設定のアイテムの場合、

NamespaceIndex=5,IdentifierType=1,Identifier=Item\_1

となります。

表 2-4: VARIANT 型データ(一部)

| データ型       | 指定値  | バイト数 | 説明                             |
|------------|------|------|--------------------------------|
| VT_I2      | 2    | 2    | 単精度(16ビット)整数                   |
| VT_I4      | 3    | 4    | 倍精度(32ビット)整数                   |
| VT_R4      | 4    | 4    | 単精度(32ビット)浮動小数                 |
| VT_R8      | 5    | 8    | 倍精度(64ビット)浮動小数                 |
| VT_CY      | 6    | 8    | VT_UI8と同じ 通貨.                  |
| VT_DATE    | 7    | 8    | VT_R8と同じ 1899/12/30からの通算日時.    |
| VT_BSTR    | 8    | 可変   | 文字列<br>UNICODE 文字と NULL ターミネータ |
| VT_BOOL    | 11   | 2    | VT_I2と同じ 0:FALSE, -1::TRUE     |
| VT_VARIANT | 12   | 可変   | バリエーション型 (バリエーション型配列にのみ使用)     |
| VT_UI1     | 17   | 1    | 符号なし文字                         |
| VT_ARRAY   | 8192 | 可変   | 上記データタイプの一次元配列                 |

**使用例**

```
private CCaoVariable _caoVariable1 = null;
private CCaoVariable _caoVariable2 = null;
private CCaoVariable _caoVariable3 = null;

_caoVariable1 = _caoController.AddVariable("Item_I4", "");
_caoVariable2 = _caoController.AddVariable("Item_R8", "");
_caoVariable3 = _caoController.AddVariable("Item_BSTR", "");
```

**2.2.3. CaoController::get\_VariableNames プロパティ**

この get\_VariableNames プロパティは、システム変数の一覧を取得します。

**2.2.4. CaoVariable::get\_DateTime プロパティ**

この get\_DateTime プロパティは、アイテム ID のタイムスタンプを取得します。

get\_Value プロパティを実行した際に取得したタイムスタンプであることに注意して下さい。

**2.2.5. CaoVariable::get\_Value プロパティ**

この get\_Value プロパティは、CaoVariable 内のキャッシュ値を取得します。

通常、キャッシュ値の更新は行われません。

最新の OPCUA サーバ上の値を取得する時は「CaoController::Execute(“MultipleRead”) コマンド」を使用してキャッシュ値の更新を行ってください。

## 3. コマンドリファレンス

### 3.1. Controller クラス

表 3-1 CaoController::Execute コマンド一覧

| コマンド         | 機能           | ページ |
|--------------|--------------|-----|
| MultipleRead | 変数値を一括取得します。 | 13  |

#### 3.1.1. CaoController::Execute("MultipleRead") コマンド

変数値を一括取得します。

本コマンドはコントローラ上にある全ての CaoVariable の値を一括取得して、それぞれのキャッシュ値を更新します。

**書式** MultipleRead()

引数 : なし  
戻り値 : なし

**使用例**

```
_caoController.Execute("MultipleRead", null);
```

## 4. エラーコード

OPC UA Multiple プロバイダでは、以下の固有エラーコードが定義されています。ORiN2 共通エラーについては、「ORiN2 プログラミングガイド」のエラーコードの章を参照してください。

表 4-1 独自エラーコード一覧

| エラー名                               | エラー番号      | 説明                                   |
|------------------------------------|------------|--------------------------------------|
| S OPCUA_TRUST                      | 0x00100800 | サーバ証明書がなく信頼しました。                     |
| E OPCUA_NOT_TRUST                  | 0x80100800 | サーバ証明書がなく信頼しませんでした。                  |
| E OPCUA_PKI_NOTFOUND               | 0x80100801 | PKI(公開鍵基盤)が見つかりませんでした。               |
| E OPCUA_FAILED_USERCERTIFICATION   | 0x80100802 | ユーザ認証に失敗しました。                        |
| E OPCUA_FAILED_START               | 0x80100803 | クライアントアプリケーションの開始に失敗しました。            |
| E OPCUA_FAILED_CONNECTSESSION      | 0x80100804 | セッションの接続に失敗しました。                     |
| E OPCUA_FAILED_GETENDPOINT         | 0x80100805 | エンドポイントの取得に失敗しました。                   |
| E OPCUA_FAILED_OPENUSERCERTIFICATE | 0x80100806 | ユーザ証明書を開けませんでした。                     |
| E OPCUA_FAILED_LOADUSERPRIVATEKEY  | 0x80100807 | ユーザ秘密鍵のロードに失敗しました。                   |
| E OPCUA_ENDPOINT_NOTFOUND          | 0x80100808 | 指定したエンドポイントが存在しませんでした。               |
| E OPCUA_FAILED_ADDSESSION          | 0x80100809 | セッションの追加に失敗しました。                     |
| E OPCUA_CERTIFICATE_SETTINGNOW     | 0x8010080a | 個別の証明書が設定中である状態で他のインスタンスを開こうとしました。   |
| E OPCUA_OTHERINSTANCE_EXIST        | 0x8010080b | 他のインスタンスが存在する状態で個別の証明書を設定しようとしてしました。 |
| E OPCUA_INVALID_ACCESSPATH         | 0x8010080c | アクセスパスが無効です。                         |
| E OPCUA_VARIABLENAME_NOTEXIST      | 0x8010080d | 存在しない変数名です。                          |
| E OPCUA_TYPEMISMATCH               | 0x8010080e | 型が一致しません。                            |
| E OPCUA_BADVARTYPE                 | 0x8010080f | 不正な型です。                              |
| E OPCUA_SECURITY_CHECKS_FAILED     | 0x80100810 | セキュリティの検証中にエラーが発生しました。               |
| E OPCUA_NODEID_NOTEXIST            | 0x80100811 | 存在しないノード ID です。                      |
| E OPCUA_NODEID_NOTENOUGHOPTIONS    | 0x80100812 | ノード ID を指定するオプション文字列が不足しています。        |
| E OPCUA_BAD_MULTIPLE_READ_ERROR    | 0x80100813 | 一括取得中にエラーが発生しました。                    |

## 5. 証明書ファイルについて

OPC UA Multiple プロバイダには以下の証明書があります。

- ・ アプリケーション証明書
- ・ 証明書による認証の為の証明書

ここではこれらの証明書と作成方法について説明します。

### 5.1. アプリケーション証明書

OPC UA Multiple プロバイダが起動するためにはアプリケーションの証明書とプライベートキーファイルが存在する必要があります。この証明書はセキュリティ接続時に使用するものと兼用となっています。

後述する手順で作成するとアプリケーションのインストールフォルダの「¥PKI¥store」以下の「certs」フォルダ、「private」フォルダの中に作成されます。

「certs」フォルダ中の「cert\_multiple\_client\_self\_signed.der」が証明書ファイルです。

「private」フォルダ中の「private\_key\_multiple\_client\_self\_signed.pem」がプライベートキーファイルです。

「Der」「Pem」「Password」オプションを使用して指定することも可能です。

※これらのファイルを指定した場合複数のコントローラを扱うことはできなくなります。

例)

- ・ コントローラが存在する場合アプリケーション証明書を指定したコントローラの追加はできません
- ・ アプリケーション証明書を指定したコントローラ使用中は新たなコントローラの追加はできません

正しいこれらのファイルが存在しない場合、または正しく指定しなかった場合、プロバイダの起動は失敗します。

### 5.2. 証明書による認証の為の証明書

認証には以下の方法があります。

- ・ Anonymous (匿名) ユーザによる認証
- ・ ユーザ名による認証
- ・ 証明書による認証

この証明書は「証明書による認証」に必要な証明書です。

後述する手順で作成するとアプリケーションのインストールフォルダの「¥UserCertificate」以下の「certs」フォルダ、「private」フォルダの中に作成されます。

「certs」フォルダ中の「cert\_multiple\_client\_user.der」が証明書ファイルです。

「private」フォルダ中の「private\_key\_multiple\_client\_user.pem」がプライベートキーファイルです。

正しいこれらのファイルが存在しない場合、または正しく指定しなかった場合、証明書による認証は失敗します。

### 5.3. 証明書の作成

ここでは上記の証明書の作成方法について説明します。

以下の手順で作成します。

- ① 「create\_store.bat」を起動します
- ② 自動的に設定保存先に作成されます。

## 6. サンプルプログラム

### 6.1. CaoOPCUA

以下に CaoOPCUA に接続し、一括取得ボタンでアイテムの値を一括取得し、リストビューに[変数名][型][値]を表示するサンプルを示します。

#### 6.1.1. アクセスパス使用

**List 6-1****Form1.cs**

```
using System;
using System.Windows.Forms;

using ORiN2.ManagedCAO;

namespace Multiple
{
    public partial class Form1 : Form
    {
        private CCaoEngine _caoEngine = null;
        private CCaoWorkspace _caoWorkspace = null;
        private CCaoController _caoController = null;
        private CCaoVariable _caoVariable1 = null;
        private CCaoVariable _caoVariable2 = null;
        private CCaoVariable _caoVariable3 = null;

        public Form1()
        {
            InitializeComponent();
        }

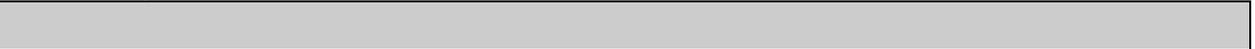
        private void Form1_Load(object sender, EventArgs e)
        {
            _caoEngine = new CCaoEngine();
            if (_caoEngine != null)
            {
                _caoWorkspace = _caoEngine.Workspaces[0];
            }
            if (_caoWorkspace != null)
            {
                _caoController = _caoWorkspace.AddController("OpcUa",
                    "CaoProv.OPCUAMultiple", "",
                    "Server=opc.tcp://192.168.188.128:4890/CaoOPCUA"
                    + ",AccessPath=OPCUA.CAO/DataStore");
            }
            if (_caoController != null)
            {
                _caoVariable1 = _caoController.AddVariable("Item_I4", "");
                _caoVariable2 = _caoController.AddVariable("Item_R8", "");
                _caoVariable3 = _caoController.AddVariable("Item_BSTR", "");
            }
        }
    }
}
```

```
    }

    private void Form1_FormClosed(object sender, FormClosedEventArgs e)
    {
        if (_caoVariable1 != null)
        {
            if (_caoController != null)
            {
                _caoController.Variables.Remove(_caoVariable1.Index);
            }
            _caoVariable1 = null;
        }
        if (_caoVariable2 != null)
        {
            if (_caoController != null)
            {
                _caoController.Variables.Remove(_caoVariable2.Index);
            }
            _caoVariable2 = null;
        }
        if (_caoVariable3 != null)
        {
            if (_caoController != null)
            {
                _caoController.Variables.Remove(_caoVariable3.Index);
            }
            _caoVariable3 = null;
        }
        if (_caoController != null)
        {
            if (_caoWorkspace != null)
            {
                _caoWorkspace.Controllers.Remove(_caoController.Index);
            }
            _caoController = null;
        }
        if (_caoWorkspace != null)
        {
            _caoWorkspace = null;
        }
        if (_caoEngine != null)
        {
            _caoEngine.Dispose();
            _caoEngine = null;
        }
    }

    private void btnGet_Click(object sender, EventArgs e)
    {
        lvItems.Items.Clear();
        if (_caoController != null)
        {
            _caoController.Execute("MultipleRead", null);
        }
        var item1 = lvItems.Items.Add(_caoVariable1.Name);
    }
}
```

```
item1.SubItems.Add(_caoVariable1.Value.GetType().ToString());
item1.SubItems.Add(_caoVariable1.Value.ToString());
var item2 = lvltems.Items.Add(_caoVariable2.Name);
item2.SubItems.Add(_caoVariable2.Value.GetType().ToString());
item2.SubItems.Add(_caoVariable2.Value.ToString());
var item3 = lvltems.Items.Add(_caoVariable3.Name);
item3.SubItems.Add(_caoVariable3.Value.GetType().ToString());
item3.SubItems.Add(_caoVariable3.Value.ToString());
    }
}
}
```



## 6.1.2. ノード ID 使用

**List 6-2**      **Form1.cs**

```
using System;
using System.Windows.Forms;

using ORiN2.ManagedCAO;

namespace Multiple
{
    public partial class Form1 : Form
    {
        private CCaoEngine _caoEngine = null;
        private CCaoWorkspace _caoWorkspace = null;
        private CCaoController _caoController = null;
        private CCaoVariable _caoVariable1 = null;
        private CCaoVariable _caoVariable2 = null;
        private CCaoVariable _caoVariable3 = null;

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            _caoEngine = new CCaoEngine();
            if (_caoEngine != null)
            {
                _caoWorkspace = _caoEngine.Workspaces[0];
            }
            if (_caoWorkspace != null)
            {
                _caoController = _caoWorkspace.AddController("OpcUa",
                    "CaoProv.OPCUAMultiple", "",
                    "Server=opc.tcp://192.168.188.128:4890/CaoOPCUA");
            }
            if (_caoController != null)
            {
                _caoVariable1 = _caoController.AddVariable("Item1",
                    "NamespaceIndex=5,IdentifierType=1,Identifier=Item_I4");
                _caoVariable2 = _caoController.AddVariable("Item2",
                    "NamespaceIndex=5,IdentifierType=1,Identifier=Item_R8");
                _caoVariable3 = _caoController.AddVariable("Item3",
                    "NamespaceIndex=5,IdentifierType=1,Identifier=Item_BSTR");
            }
        }

        private void Form1_FormClosed(object sender, FormClosedEventArgs e)
        {
            if (_caoVariable1 != null)
            {
                if (_caoController != null)
            
```

```
        {
            _caoController.Variables.Remove(_caoVariable1.Index);
        }
        _caoVariable1 = null;
    }
    if (_caoVariable2 != null)
    {
        if (_caoController != null)
        {
            _caoController.Variables.Remove(_caoVariable2.Index);
        }
        _caoVariable2 = null;
    }
    if (_caoVariable3 != null)
    {
        if (_caoController != null)
        {
            _caoController.Variables.Remove(_caoVariable3.Index);
        }
        _caoVariable3 = null;
    }
    if (_caoController != null)
    {
        if (_caoWorkspace != null)
        {
            _caoWorkspace.Controllers.Remove(_caoController.Index);
        }
        _caoController = null;
    }
    if (_caoWorkspace != null)
    {
        _caoWorkspace = null;
    }
    if (_caoEngine != null)
    {
        _caoEngine.Dispose();
        _caoEngine = null;
    }
}

private void btnGet_Click(object sender, EventArgs e)
{
    lvItems.Items.Clear();
    if (_caoController != null)
    {
        _caoController.Execute("MultipleRead", null);
    }
    var item1 = lvItems.Items.Add(_caoVariable1.Name);
    item1.SubItems.Add(_caoVariable1.Value.GetType().ToString());
    item1.SubItems.Add(_caoVariable1.Value.ToString());
    var item2 = lvItems.Items.Add(_caoVariable2.Name);
    item2.SubItems.Add(_caoVariable2.Value.GetType().ToString());
    item2.SubItems.Add(_caoVariable2.Value.ToString());
    var item3 = lvItems.Items.Add(_caoVariable3.Name);
    item3.SubItems.Add(_caoVariable3.Value.GetType().ToString());
}
```

```
        item3.SubItems.Add(_caoVariable3.Value.ToString());  
    }  
}
```

