# 株式会社アイエイアイ
# IAI SEL プロバイダ

## Version 1.0.2

# ユーザーズ ガイド

## December 21, 2021

備考：

## 【改版履歴】

| バージョン | 日付 | 内容 |
|---|---|---|
| 1.0.0 | 2017-09-22 | 初版. |
| 1.0.1 | 2018-10-31 | 接続失敗時にメモリリークが発生する問題を修正. |
| | 2020-10-6 | ライセンス追加事項追記. |
| 1.0.2 | 2021-12-21 | エラー誤検知を修正しました. |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

## 【対応機器】

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

## 【ご注意】

**本プロバイダを使用する場合は “IAI プログラムタイプコントローラプロバイダ” ライセンスが必要です.**

1. はじめに ..... 6
1.1. ライセンスの追加 ..... 7
2. プロバイダの概要 ..... 8
2.1. 概要 ..... 8
2.2. 軸パターンについて ..... 9
2.3. メソッド・プロパティ ..... 9
2.3.1. CaoWorkspace::AddController メソッド ..... 9
2.3.2. CaoController:: Execute メソッド ..... 11
2.3.2.1. CaoController::Execute("TEST_CALL")コマンド ..... 14
2.3.2.2. CaoController::Execute("VERSION_CODE")コマンド ..... 14
2.3.2.3. CaoController::Execute("POSITION_DATA_NUM ")コマンド ..... 15
2.3.2.4. CaoController::Execute("POSITION_DATA ")コマンド ..... 15
2.3.2.5. CaoController::Execute("INPUT_PORT ")コマンド ..... 16
2.3.2.6. CaoController::Execute("OUTPUT_PORT")コマンド ..... 16
2.3.2.7. CaoController::Execute("FLAG")コマンド ..... 17
2.3.2.8. CaoController::Execute("INTEGER_VARIABLE")コマンド ..... 17
2.3.2.9. CaoController::Execute("REAL_VARIABLE ")コマンド ..... 18
2.3.2.10. CaoController::Execute("STRING_VARIABLE ")コマンド ..... 18
2.3.2.11. CaoController::Execute("AXIS_STATUS ")コマンド ..... 19
2.3.2.12. CaoController::Execute("PROGRAM_STATUS ")コマンド ..... 19
2.3.2.13. CaoController::Execute("SYSTEM_STATUS ")コマンド ..... 20
2.3.2.14. CaoController::Execute("ERROR_INFO ")コマンド ..... 21
2.3.2.15. CaoController::Execute("POSITION_DATA_NUM 2")コマンド ..... 22
2.3.2.16. CaoController::Execute("POSITION_DATA 2")コマンド ..... 22
2.3.2.17. CaoController::Execute("POSITION_DATA3")コマンド ..... 23
2.3.2.18. CaoController::Execute("SERVO ")コマンド ..... 24
2.3.2.19. CaoController::Execute("ORIGIN ")コマンド ..... 24
2.3.2.20. CaoController::Execute("MOVE_ABSOLUITE")コマンド ..... 25
2.3.2.21. CaoController::Execute("MOVE_RELATIVE ")コマンド ..... 25
2.3.2.22. CaoController::Execute("MOVE_JOG_INCHUNG ")コマンド ..... 26
2.3.2.23. CaoController::Execute("MOVE_POSITION_NO ")コマンド ..... 26
2.3.2.24. CaoController::Execute("STOP_CANCEL ")コマンド ..... 27
2.3.2.25. CaoController::Execute("WRITE_POSITION_DATA_RANGE")コマンド ..... 27

2.3.2.26. CaoController::Execute("WRITE_CHANGE_POSITION_DATA ")コマンド.....28
2.3.2.27. CaoController::Execute("CLEAR_POSITION_DATA ")コマンド .....28
2.3.2.28. CaoController::Execute("CHANGE_OUTPUT_PORT_STATUS ")コマンド.....29
2.3.2.29. CaoController::Execute("CHANGE_FLUG_STATUS ")コマンド.....29
2.3.2.30. CaoController::Execute("CHANGE_INTEGER_VARIABLE")コマンド.....30
2.3.2.31. CaoController::Execute("CHANGE_REAL_VARIABLE ")コマンド .....31
2.3.2.32. CaoController::Execute("CHANGE_STRING_VARIABLE")コマンド .....31
2.3.2.33. CaoController::Execute("RESET_ALARM ")コマンド.....32
2.3.2.34. CaoController::Execute("EXECUTION_PROGRAM ")コマンド.....32
2.3.2.35. CaoController::Execute("END_PROGRAM ")コマンド .....32
2.3.2.36. CaoController::Execute("PAUSE_PROGRAM ")コマンド.....32
2.3.2.37. CaoController::Execute("EXECUTION_ONE_STEP_PROGRAM ")コマンド.....33
2.3.2.38. CaoController::Execute("EXECUTION_RESUME_PROGRAM ")コマンド.....33
2.3.2.39. CaoController::Execute("RESET_SOFTWARE ")コマンド .....33
2.3.2.40. CaoController::Execute("REQUEST_RESTORATION_DRIVE ")コマンド .....33
2.3.2.41. CaoController::Execute("REQUEST_PAUSE_RELEASE_OPERATION ")コマンド34
2.3.2.42. CaoController::Execute("CHANGE_SPEED")コマンド.....34
2.3.2.43. CaoController::Execute("MOVE_POSITION_NO2")コマンド .....34
2.3.2.44. CaoController::Execute("WRITE_POSITION_DATA_RANGE2")コマンド.....35
2.3.2.45. CaoController::Execute("WRITE_CHANGE_POSITION_DATA2")コマンド .....36
2.3.2.46. CaoController::Execute("CLEAR_POSITION_DATA2")コマンド .....36
2.3.2.47. CaoController::Execute("CONTROLLER_FUNCTION2")コマンド .....37
2.3.2.48. CaoController::Execute("WRITE_POSITION_DATA_RANGE3")コマンド.....38
2.3.2.49. CaoController::Execute("WRITE_CHANGE_POSITION_DATA3")コマンド .....39
2.3.2.50. CaoController::Execute("COORDINATE_DATA_RANGE ")コマンド.....40
2.3.2.51. CaoController::Execute("UNIT_AXIS_STATUS ")コマンド.....41
2.3.2.52. CaoController::Execute("CHECK_ZONE_DATA_RANGE ")コマンド.....42
2.3.2.53. CaoController::Execute("UNIT_AXIS_STATUS2")コマンド.....43
2.3.2.54. CaoController::Execute("COORDINATE_DATA_RANGE2")コマンド.....44
2.3.2.55. CaoController::Execute("MOVE_UNIT_ABSOLUIT ")コマンド.....44
2.3.2.56. CaoController::Execute("MOVE_UNIT_RELATIVE ")コマンド.....45
2.3.2.57. CaoController::Execute("MOVE_UNIT_POSITION_NO")コマンド.....45
2.3.2.58. CaoController::Execute("MOVE_UNIT_POSITION_NO2")コマンド.....46
2.3.3. CaoController::AddVariable メソッド .....47
2.3.4. CaoController::get_VariableNames メソッド .....47
2.3.5. CaoVariable::get_Value プロパティ .....47
2.3.6. CaoVariable::set_Value プロパティ .....47

2.4. 変数一覧....48
2.4.1. コントローラクラス....48
2.4.1.1. @MAKER_NAME....49
2.4.1.2. @VERSION....49
2.4.1.3. VERSION_CODE<??>....49
2.4.1.4. @POSITION_DATA_NUM....50
2.4.1.5. POSITION_DATA<??>....50
2.4.1.6. INPUT_PORT<??>....51
2.4.1.7. OUTPUT_PORT<??>....51
2.4.1.8. FLAG<??>....52
2.4.1.9. INTEGER_VARIABLE<??>....53
2.4.1.10. REAL_VARIABLE<??>....54
2.4.1.11. STRING_VARIABLE<??>....54
2.4.1.12. AXIS_STATUS<??>....55
2.4.1.13. PROGRAM_STATUS<??>....55
2.4.1.14. @SYSTEM_STATUS....56
2.4.1.15. ERROR_INFO<??>....57
2.4.1.16. @EX2_POSITION_DATA_NUM....58
2.4.1.17. EX2_POSITION_DATA<??>....58
2.4.1.18. EX3_POSITION_DATA<??>....59
2.4.1.19. COORDINATE_DATA_RANGE<??>....60
2.4.1.20. UNIT_AXIS_STATUS<??>....61
2.4.1.21. CHECK_ZONE_DATA_RANGE<??>....62
2.4.1.22. EX2_UNIT_AXIS_STATUS<??>....63
2.4.1.23. EX2_COORDINATE_DATA_RANGE<??>....64
2.5. エラーコード....65
3. サンプルプログラム....66

# 1. はじめに

本書は，SEL デバイスからデータの取得を行なう，SEL プロバイダのユーザーズガイドです.

このプロバイダを用いれば，IAI 社製の SEL デバイスへのデータの読み書きをすることが容易になります.

本書は，この SEL プロバイダの機能と実装されているメソッドについて説明します.

## 1.1. ライセンスの追加

本プロバイダを使用可能にするには ORiN2 SDK をインストール後，別途「IAI プログラムタイプコントローラプロバイダ」ライセンスを入力する必要があります．評価用にインストールする場合は下記のライセンスキーをご使用ください．

**IIEA-A262-RYCA-CR9G （評価用 3 ヶ月）**

下記に「IAI プログラムタイプコントローラプロバイダ」ライセンスの追加手順を示します．

1. CaoConfig を起動し，[Cao Provider]タブを選択する
2. Provider List から[IAI SEL CAO Provider]項目を選択する
3. License 項目の[…]ボタンをクリックする
4. ORiN2 License Manager で[Add]ボタンをクリックする
5. 入手したライセンスキーを入力後，[OK]ボタンをクリックする
6. [Close] ボタンをクリックし，CaoConfig を終了する

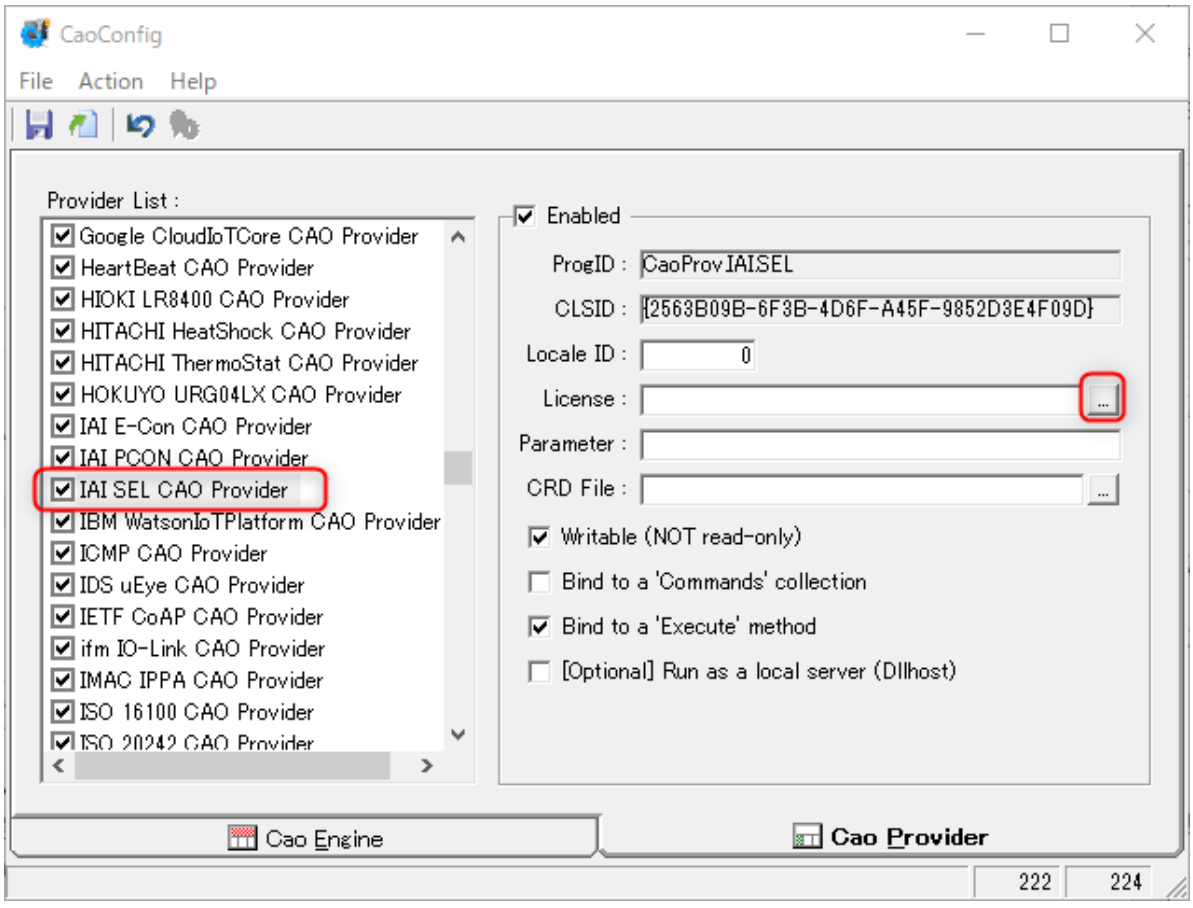


**図 1-1「IAI プログラムタイプコントローラプロバイダ」ライセンス追加**

# 2. プロバイダの概要

## 2.1. 概要

SEL プロバイダは，COM 通信 もしくは TCP/IP 通信によってデータを読み書きします．SEL プロバイダのファイル形式は DLL(Dynamic Link Library)となっており，その詳細は表 2-1 のようになっています．

表 2-1 SEL プロバイダ

| | |
|---|---|
| ファイル名 | CaoProvIAISEL.dll |
| ProgID | CaoProv.IAI.SEL |
| レジストリ登録 | regsvr32 CaoProvIAISEL.dll |
| レジストリ登録の抹消 | regsvr32 /u CaoProvIAISEL.dll |

また，SEL プロバイダと SEL デバイスそれぞれの対応を表した図が下図 2-1 となります．

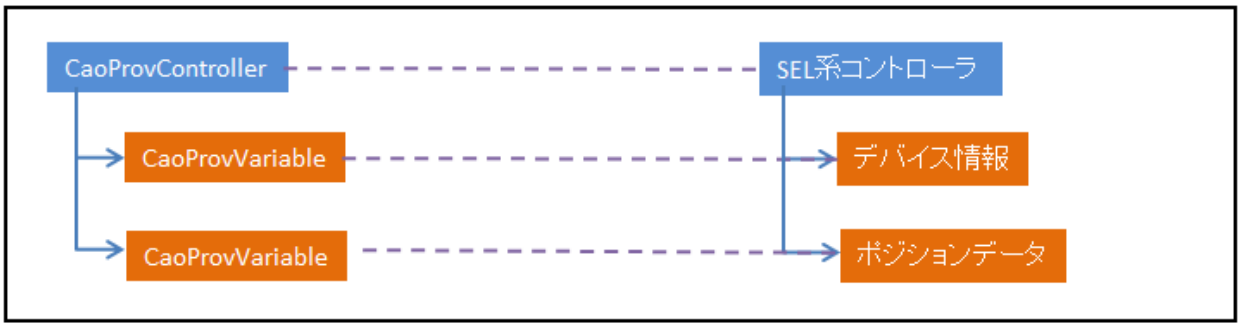


図 2-1 CaoProvController と SEL デバイスの対応図

## 2.2. 軸パターンについて

値で軸パターンと記載がある箇所は，対象の軸番号を:(コロン)区切りで複数指定できるパラメータです.

例 1:2:3:4:5:6:7:8

ただし，重複で設定した場合エラーとなります.

例 1:2:3:4

## 2.3. メソッド・プロパティ

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

Controller オブジェクトの生成時に SEL デバイスに接続するために必要なオプションを指定します.

以下に，AddController の仕様を示します.

書式

```
AddController
(
        “<コントローラ名>”,          // コントローラ名(任意)
        “CaoProv.IAI.SEL”,          // プロバイダ名 (固定)
        “<マシン名>”,               // プロバイダ実行マシン名(未使用)
        “<オプション>”,             // オプション文字列
)
```

以下にオプション文字列に指定する文字列を示します.

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

| オプション | 必須 | 説明 | 値範囲 | デフォルト値 |
|---|---|---|---|---|
| CONN= COM:<接続先 COM ポート番号> | ○<br>※1 | 接続先の COM ポート番号を環境に合わせて指定します. | ----- | ----- |
| CONN= ETH:<接続先 IP>[:<接続先ポート>] | ○<br>※1 | 接続先の IP アドレスおよびポートを環境に合わせて指定します. | ----- | ポート：<br>64611 |
| TIMEOUT=<応答待機時間> | - | 応答待機時間を指定します(ms). | 1 - 65535 | 500 |
| STATION=<局番 > | - | 接続先のデバイスの局番を環境に合わせて指定します. | 0 - 255 | 99 |

※1 COM か ETH はいずれかが必須

使用例：

```
Dim controller As Object
controller = Cao.AddController("SEL", "CaoProv.IAI.SEL", "", " CONN=COM:1,TIMEOUT=500,STATION=1")
```

### 2.3.2. CaoController:: Execute メソッド

CaoController にて Execute を使用することによってデータの読み書きができるものがあります.
以下に, Execute の仕様を示します.

書式

```
Execute
(
        “<メソッド名>”,         // メソッド名
        “<引数>”,               // 引数
)
```

表 2-3 CaoController::Execute メソッド一覧

| コマンド名 | 説明 | リンク |
|---|---|---|
| TEST_CALL | コミュニケーションのテストを実行します. | P.14 |
| VERSION_CODE | バージョンコードの取得を実行します. | P.14 |
| POSITION_DATA_NUM | 有効ポジションデータ数の取得を実行します. | P.15 |
| POSITION_DATA | 有効ポジションデータの取得を実行します. | P.15 |
| INPUT_PORT | 入力ポートの取得を実行します. | P.16 |
| OUTPUT_PORT | 出力ポートの取得を実行します. | P.16 |
| FLAG | フラグの取得を実行します. | P.17 |
| INTEGER_VARIABLE | 整数変数の取得を実行します. | P.18 |
| REAL_VARIABLE | 実数変数の取得を実行します. | P.18 |
| STRING_VARIABLE | ストリング変数の取得を実行します. | P.18 |
| AXIS_STATUS | 軸ステータスの取得を実行します. | P.19 |
| PROGRAM_STATUS | プログラムステータスの取得を実行します. | P.19 |
| SYSTEM_STATUS | システムステータスの取得を実行します. | P.20 |
| ERROR_INFO | エラー詳細情報の取得を実行します. | P.21 |
| POSITION_DATA_NUM2 | 有効ポジションデータ数 2 の取得を実行します. | P.22 |
| POSITION_DATA2 | 有効ポジションデータ 2 の取得を実行します. | P.22 |
| POSITION_DATA3 | 有効ポジションデータ 3 の取得を実行します. | P.23 |
| SERVO | サーボのオン／オフを実行します. | P.24 |
| ORIGIN | 原点復帰を実行します. | P.24 |

| | | |
|---|---|---|
| MOVE_ABSOLUITE | 絶対座標指定移動を実行します. | P.25 |
| MOVE_RELATIVE | 相対座標指定移動を実行します. | P.25 |
| MOVE_JOG_INCHUNG | ジョグ・インチング移動を実行します. | P.26 |
| MOVE_POSITION_NO | ポジション No.指定移動を実行します. | P.26 |
| STOP_CANCEL | 動作停止＆キャンセルを実行します. | P.27 |
| WRITE_POSITION_DATA_RANGE | ポジションデータ範囲指定連続書込みを実行します. | P.27 |
| WRITE_CHANGE_POSITION_DATA | 変更ポジションデータ連続書込みを実行します. | P.28 |
| CLEAR_POSITION_DATA | ポジションデータクリアを実行します. | P.28 |
| CHANGE_OUTPUT_PORT_STATUS | 出力ポート状態変更を実行します. | P.29 |
| CHANGE_FLUG_STATUS | フラグ状態変更を実行します. | P.29 |
| CHANGE_INTEGER_VARIABLE | 整数変数変更を実行します. | P.30 |
| CHANGE_REAL_VARIABLE | 実数変数変更を実行します. | P.31 |
| CHANGE_STRING_VARIABLE | ストリング変数変更を実行します. | P.31 |
| RESET_ALARM | アラームリセットを実行します. | P.32 |
| EXECUTION_PROGRAM | プログラム実行を実行します. | P.32 |
| END_PROGRAM | プログラム終了を実行します. | P.32 |
| PAUSE_PROGRAM | プログラム一時停止を実行します. | P.32 |
| EXECUTION_ONE_STEP_PROGRAM | プログラム 1 ステップ実行を実行します. | P.33 |
| EXECUTION_RESUME_PROGRAM | プログラム実行再開を実行します. | P.33 |
| RESET_SOFTWARE | ソフトウェアリセットを実行します. | P.33 |
| REQUEST_RESTORATION_DRIVE | 駆動源復旧要求を実行します. | P.33 |
| REQUEST_PAUSE_RELEASE_OPERATION | 動作一時停止解除要求を実行します. | P.34 |
| CHANGE_SPEED | 速度チェンジを実行します. | P.34 |
| MOVE_POSITION_NO2 | ポジション No.指定移動 2 を実行します. | P.34 |
| WRITE_POSITION_DATA_RANGE2 | ポジションデータ範囲指定連続書込み 2 を実行します. | P.35 |
| WRITE_CHANGE_POSITION_DATA2 | 変更ポジションデータ連続書込み 2 を実行します. | P.36 |
| CLEAR_POSITION_DATA2 | ポジションデータクリア 2 を実行します. | P.36 |
| CONTROLLER_FUNCTION2 | コントローラ機能指定 2 を実行します. | P.37 |
| WRITE_POSITION_DATA_RANGE3 | ポジションデータ範囲指定連続書込み 3 を実行します. | P.38 |
| WRITE_CHANGE_POSITION_DATA3 | 変更ポジションデータ連続書込み 3 を実行します. | P.39 |
| COORDINATE_DATA_RANGE | 座標系定義データ範囲指定連続の取得を実行します. | P.40 |
| UNIT_AXIS_STATUS | ユニット軸ステータスの取得を実行します. | P.41 |
| CHECK_ZONE_DATA_RANGE | 簡易干渉チェックゾーン定義データ範囲指定連続の取得を実行します. | P.42 |

| | | |
|---|---|---|
| UNIT_AXIS_STATUS2 | ユニット軸ステータス 2 の取得を実行します. | P.43 |
| COORDINATE_DATA_RANGE2 | 座標系定義データ範囲指定連続 2 の取得を実行します. | P.44 |
| MOVE_UNIT_ABSOLUITE | ユニット絶対座標指定移動を実行します. | P.44 |
| MOVE_UNIT_RELATIVE | ユニット相対座標指定移動を実行します. | P.45 |
| MOVE_UNIT_POSITION_NO | ユニットポジション No. 指定移動を実行します. | P.45 |
| MOVE_UNIT_POSITION_NO2 | ユニットポジション No. 指定移動 2 を実行します. | P.46 |

#### 2.3.2.1. CaoController::Execute("TEST_CALL")コマンド

コミュニケーションのテストを実行します.

引数：

| | |
|---|---|
| VT_BSTR | 送信する文字列を指定します. |

戻り値：

| | |
|---|---|
| VT_BSTR | 受信した文字列が返却されます.<br>(送信する文字列で指定したものと同じものが返却される) |

使用例：

```
Dim sVal As String
sVal = controller.Execute("TEST_CALL" , "01234567489")
```

#### 2.3.2.2. CaoController::Execute("VERSION_CODE")コマンド

バージョンコードの取得を実行します.

引数：

| | | |
|---|---|---|
| VT_ARRAY \| VT_UI2 | | |
| 0 | VT_UI2 | 照会先頭ポジションNo.を指定します. |
| 1 | VT_UI2 | 照会レコード数を指定します. |
| 2 | VT_UI2 | 有効軸数を指定します. |

戻り値：

| | | |
|---|---|---|
| VT_ARRAY \| VT_UI2 | | |
| 0 | VT_UI2 | 機種コードが返却されます. |
| 1 | VT_UI2 | ユニットコードが返却されます. |
| 2 | VT_UI2 | バージョンNo.が返却されます. |
| 3 | VT_UI2 | 時刻が返却されます. |
| 4 | VT_UI2 | 時刻が返却されます. |
| 5 | VT_UI2 | 時刻が返却されます. |
| 6 | VT_UI2 | 時刻が返却されます. |
| 7 | VT_UI2 | 時刻が返却されます. |
| 8 | VT_UI2 | 時刻が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("VERSION_CODE" , Array(1, 2, 3))
```

#### 2.3.2.3. CaoController::Execute("POSITION_DATA_NUM ")コマンド

有効ポジションデータ数の取得を実行します.

引数：なし

戻り値：

| | |
|---|---|
| VT_UI2 | 有効ポジションデータ数が返却されます. |

使用例：

```
Dim lVal As Long
lVal = controller.Execute("POSITION_DATA_NUM ")
```

#### 2.3.2.4. CaoController::Execute("POSITION_DATA ")コマンド

有効ポジションデータの取得を実行します.

引数：

| | | | |
|---|---|---|---|
| VT_ARRAY \| VT_UI2 | | | |
| 0 | VT_UI2 | | 照会先頭ポジション No. を指定します. |
| 1 | VT_UI2 | | 照会レコード数を指定します. |
| 2 | VT_UI2 | | 有効軸数を指定します. |

戻り値：

| | | | | |
|---|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | | |
| 0 | VT_ARRAY \| VT_VARIANT | | | |
| | 0 | VT_UI2 | | ポジション No. が返却されます. |
| | 1 | VT_BSTR | | 軸パターンが返却されます. |
| | 2 | VT_UI2 | | 加速度が返却されます. |
| | 3 | VT_UI2 | | 減速度が返却されます. |
| | 4 | VT_UI2 | | 速度が返却されます. |
| | 5 | VT_ARRAY \| VT_I4 | | |
| | | n | VT_I4 | 位置データが返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("POSITION_DATA ", Array(1, 2, 3))
```

#### 2.3.2.5. CaoController::Execute("INPUT_PORT ")コマンド

入力ポートの取得を実行します.

引数：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| 0 | VT_UI2 | 照会開始ポート No. を指定します. |
| 1 | VT_UI2 | 照会ポート数を指定します. |

戻り値：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| n | VT_UI2 | 入力ポートデータが返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("INPUT_PORT ", Array(1, 2))
```

#### 2.3.2.6. CaoController::Execute("OUTPUT_PORT")コマンド

出力ポートの取得を実行します.

引数：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| 0 | VT_UI2 | 照会開始ポート No. を指定します. |
| 1 | VT_UI2 | 照会ポート数を指定します. |

戻り値：

使用例：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| n | VT_UI2 | 出力ポートデータが返却されます. |

```
Dim vArray As Variant
vArray = controller.Execute("OUTPUT_PORT", Array(1, 2))
```

#### 2.3.2.7. CaoController::Execute("FLAG")コマンド

引数：

| VT_ARRAY \| VT_VARIANT | | |
|---|---|---|
| 0 | VT_UI2 | プログラム No. を指定します. |
| 1 | VT_UI2 | 開始フラグ No. を指定します. |
| 2 | VT_UI2 | フラグ数を指定します. |
| 3 | VT_BOOL | 配列有無を指定します.<br>(true かつ フラグ数=1 の場合, フラグデータを VT_UI2 で返却) |

戻り値：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| n | VT_UI2 | フラグデータが返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("FLAG", Array(1, 2, 3, true))
```

#### 2.3.2.8. CaoController::Execute("INTEGER_VARIABLE")コマンド

整数変数の取得を実行します.

引数：

| VT_ARRAY \| VT_VARIANT | | |
|---|---|---|
| 0 | VT_UI2 | プログラム No. を指定します. |
| 1 | VT_UI2 | 開始変数 No. を指定します. |
| 2 | VT_UI2 | 変数データ数を指定します. |
| 3 | VT_BOOL | 配列有無を指定します.<br>(true かつ 変数データ数=1 の場合, 整数変数データを VT_UI2 で返却) |

戻り値：

| VT_ARRAY \| VT_I4 | | |
|---|---|---|
| n | VT_I4 | 整数変数データが返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("INTEGER_VARIABLE", Array(1, 2, 3, true))
```

#### 2.3.2.9. CaoController::Execute("REAL_VARIABLE ")コマンド

実数変数の取得を実行します.

引数：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| 0 | VT_UI2 | プログラム No. を指定します. |
| 1 | VT_UI2 | 開始変数 No. を指定します. |
| 2 | VT_UI2 | 変数データ数を指定します. |
| 3 | VT_BOOL | 配列有無を指定します.<br>(true かつ 変数データ数=1 の場合, 整数変数データを VT_UI2 で返却) |
| 4 | VT_UI2 | エンディアンを指定します. |

戻り値：

| VT_ARRAY \| VT_R8 | | |
|---|---|---|
| n | VT_R8 | 実数変数データが返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("REAL_VARIABLE ", Array())
```

#### 2.3.2.10. CaoController::Execute("STRING_VARIABLE ")コマンド

ストリング変数の取得を実行します.

引数：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| 0 | VT_UI2 | プログラム No. を指定します. |
| 1 | VT_UI2 | 開始変数 No. を指定します. |
| 2 | VT_UI2 | 変数データ数を指定します. |

戻り値：

| VT_BSTR | ストリング変数データが返却されます. |
|---|---|

使用例：

```
Dim sVal As String
sVal = controller.Execute("STRING_VARIABLE ", Array(1, 2 ,3))
```

#### 2.3.2.11. CaoController::Execute("AXIS_STATUS ")コマンド

軸ステータスの取得を実行します.

引数：

| | |
|---|---|
| VT_BSTR | 照会軸パターンを指定します. |

戻り値：

| | | | |
|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | |
| 0 | VT_ARRAY \| VT_I4 | | |
| | 0 | VT_I4 | 軸ステータスが返却されます. |
| | 1 | VT_I4 | 軸センサー入力ステータスが返却されます. |
| | 2 | VT_I4 | 軸関連エラーコードが返却されます. |
| | 3 | VT_I4 | エンコーダステータスが返却されます.<br>（リセット時） |
| | 4 | VT_I4 | 現在位置が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("AXIS_STATUS ", "1:2:3")
```

#### 2.3.2.12. CaoController::Execute("PROGRAM_STATUS ")コマンド

プログラムステータスの取得を実行します.

引数：

| | |
|---|---|
| VT_UI2 | プログラム No. を指定します. |

戻り値：

| | | |
|---|---|---|
| VT_ARRAY \| VT_UI2 | | |
| 0 | VT_UI2 | ステータスが返却されます. |
| 1 | VT_UI2 | 実行中プログラムステップ No. が返却されます. |
| 2 | VT_UI2 | プログラム依存エラーコードが返却されます. |
| 3 | VT_UI2 | エラー発生ステップ No. が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("PROGRAM_STATUS ", 1)
```

#### 2.3.2.13. CaoController::Execute("SYSTEM_STATUS ")コマンド

システムステータスの取得を実行します.

引数：なし

戻り値：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| 0 | VT_UI2 | システムモード |
| 1 | VT_UI2 | 最重レベルシステム<br>エラーNo. |
| 2 | VT_UI2 | 最新システム<br>エラーNo. |
| 3 | VT_UI2 | システムステータス<br>バイト1 |
| 4 | VT_UI2 | システムステータス<br>バイト2 |
| 5 | VT_UI2 | システムステータス<br>バイト3 |
| 6 | VT_UI2 | システムステータス<br>バイト4 |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("SYSTEM_STATUS", Array())
```

#### 2.3.2.14. CaoController::Execute("ERROR_INFO ")コマンド

エラー詳細情報の取得を実行します.

引数：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| 0 | VT_UI2 | 種別 1 を指定します. |
| 1 | VT_UI2 | 種別 2 を指定します. |
| 2 | VT_UI2 | レコード No. を指定します. |

戻り値：

| VT_ARRAY \| VT_VARIANT | | |
|---|---|---|
| 0 | VT_UI4 | エラーNo. が返却されます. |
| 1 | VT_UI4 | 詳細情報 1 が返却されます. |
| 2 | VT_UI4 | 詳細情報 2 が返却されます. |
| 3 | VT_UI4 | 詳細情報 3 が返却されます. |
| 4 | VT_I4 | 詳細情報 4 が返却されます. |
| 5 | VT_UI4 | 詳細情報 5 が返却されます. |
| 6 | VT_UI4 | 詳細情報 6 が返却されます. |
| 7 | VT_UI4 | 詳細情報 7 が返却されます. |
| 8 | VT_UI4 | 詳細情報 8 が返却されます. |
| 9 | VT_BSTR | メッセージ文字列が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("ERROR_INFO ", Array(1, 2, 3))
```

#### 2.3.2.15. CaoController::Execute("POSITION_DATA_NUM 2")コマンド

有効ポジションデータ数 2 の取得を実行します.

引数：なし

戻り値：

| | |
|---|---|
| VT_UI2 | 有効ポジションデータ数が返却されます. |

使用例：

```
Dim lVal As Long
lVal = controller.Execute("POSITION_DATA_NUM2")
```

#### 2.3.2.16. CaoController::Execute("POSITION_DATA 2")コマンド

有効ポジションデータ 2 の取得を実行します.

引数：

| | | |
|---|---|---|
| VT_ARRAY \| VT_UI2 | | |
| 0 | VT_UI2 | 照会先頭ポジション No. を指定します. |
| 1 | VT_UI2 | 照会レコード数を指定します. |
| 2 | VT_UI2 | 有効軸数を指定します. |

戻り値：

| | | | | |
|---|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | | |
| 0 | VT_ARRAY \| VT_VARIANT | | | |
| | 0 | VT_UI2 | | ポジション No. が返却されます. |
| | 1 | VT_BSTR | | 軸パターンが返却されます. |
| | 2 | VT_UI2 | | 加速度が返却されます. |
| | 3 | VT_UI2 | | 減速度が返却されます. |
| | 4 | VT_UI2 | | 速度が返却されます. |
| | 5 | VT_ARRAY \| VT_I4 | | |
| | | n | VT_I4 | 位置データが返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("POSITION_DATA2", Array(1, 2, 3))
```

### 2.3.2.17. CaoController::Execute("POSITION_DATA3")コマンド

有効ポジションデータ 3 の取得を実行します.

引数：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| 0 | VT_UI2 | レコードフォーマット拡張指定種別を指定します. |
| 1 | VT_UI2 | 照会先頭ポジション No. を指定します. |
| 2 | VT_UI2 | 照会レコード数を指定します. |
| 3 | VT_UI2 | 有効軸数を指定します. |

戻り値：

| VT_ARRAY \| VT_VARIANT | | | |
|---|---|---|---|
| 0 | VT_ARRAY \| VT_VARIANT | | |
| | 0 | VT_UI2 | ポジション No. が返却されます. |
| | 1 | VT_UI2 | 軸パターンが返却されます. |
| | 2 | VT_UI2 | 加速度が返却されます. |
| | 3 | VT_UI2 | 減速度が返却されます. |
| | 4 | VT_UI2 | 速度が返却されます. |
| | 5 | VT_ARRAY \| VT_I4 | |
| | | n VT_I4 | 位置データが返却されます. |
| | 6 | VT_UI2 | 拡張データが返却されます. |
| | 7 | VT_UI4 | 出力ファンクションコードが返却されます. |
| | 8 | VT_UI4 | 出力ポート・フラグ No. が返却されます. |
| | 9 | VT_UI4 | ファンクションパラメータ 1 が返却されます. |
| | 10 | VT_UI4 | ファンクションパラメータ 2 が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("POSITION_DATA3", Array(1, 2, 3, 4))
```

#### 2.3.2.18. CaoController::Execute("SERVO ")コマンド

サーボのオン／オフを実行します.

引数：

| VT_ARRAY \| VT_VARIANT | | |
|---|---|---|
| 0 | VT_BSTR | 軸パターンを指定します. |
| 1 | VT_UI2 | 動作種別を指定します. |

戻り値：なし

使用例：

```
controller.Execute("SERVO ", Array("1:2:3", 1))
```

#### 2.3.2.19. CaoController::Execute("ORIGIN ")コマンド

原点復帰を実行します.

引数：

| VT_ARRAY \| VT_VARIANT | | |
|---|---|---|
| 0 | VT_BSTR | 軸パターンを指定します. |
| 1 | VT_UI2 | 原点復帰時エンドサーチ速度を指定します. |
| 2 | VT_UI2 | 原点復帰時クリープ速度を指定します. |

戻り値：なし

使用例：

```
controller.Execute("ORIGIN ", Array("1:2:3", 2, 3))
```

#### 2.3.2.20. CaoController::Execute("MOVE_ABSOLUITE")コマンド

絶対座標指定移動を実行します.

引数：

| | | | | |
|---|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | | |
| 0 | VT_BSTR | | | 軸パターンを指定します. |
| 1 | VT_UI2 | | | 加速度を指定します. |
| 2 | VT_UI2 | | | 減速度を指定します. |
| 3 | VT_UI2 | | | 速度を指定します. |
| 4 | VT_ARRAY \| VT_I4 | | | |
| | | n | VT_I4 | 絶対座標データを指定します. |

戻り値：なし

使用例：

```
controller.Execute("MOVE_ABSOLUITE", Array("1:2:3", 2, 3, 4, Array(10, 20)))
```

#### 2.3.2.21. CaoController::Execute("MOVE_RELATIVE ")コマンド

相対座標指定移動を実行します.

引数：

| | | | | |
|---|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | | |
| 0 | VT_BSTR | | | 軸パターンを指定します. |
| 1 | VT_UI2 | | | 加速度を指定します. |
| 2 | VT_UI2 | | | 減速度を指定します. |
| 3 | VT_UI2 | | | 速度を指定します. |
| 4 | VT_ARRAY \| VT_I4 | | | |
| | | n | VT_I4 | 相対座標データを指定します. |

戻り値：なし

使用例：

```
controller.Execute("MOVE_RELATIVE ", Array("1:2:3", 2, 3, 4, Array(10, 20)))
```

#### 2.3.2.22. CaoController::Execute("MOVE_JOG_INCHUNG ")コマンド

ジョグ・インチング移動を実行します.

引数：

| VT_ARRAY \| VT_VARIANT | | |
|---|---|---|
| 0 | VT_BSTR | 軸パターンを指定します. |
| 1 | VT_UI2 | 加速度を指定します. |
| 2 | VT_UI2 | 減速度を指定します. |
| 3 | VT_UI2 | 速度を指定します. |
| 4 | VT_UI4 | インチング距離を指定します. |
| 5 | VT_UI2 | 動作種別を指定します. |

戻り値：なし

使用例：

```
controller.Execute("MOVE_JOG_INCHUNG", Array("1:2:3", 1, 2, 3, 4, 1))
```

#### 2.3.2.23. CaoController::Execute("MOVE_POSITION_NO ")コマンド

ポジション No. 指定移動を実行します.

引数：

| VT_ARRAY \| VT_VARIANT | | |
|---|---|---|
| 0 | VT_BSTR | 軸パターンを指定します. |
| 1 | VT_UI2 | 加速度を指定します. |
| 2 | VT_UI2 | 減速度を指定します. |
| 3 | VT_UI2 | 速度を指定します. |
| 4 | VT_UI2 | ポジション No. を指定します. |

戻り値：なし

使用例：

```
controller.Execute("MOVE_POSITION_NO", Array("1:2:3", 1, 2, 3, 4))
```

#### 2.3.2.24. CaoController::Execute("STOP_CANCEL ")コマンド

動作停止＆キャンセルを実行します.

引数：

| | | |
|---|---|---|
| VT_ARRAY \| VT_VARIANT | | |
| 0 | VT_BSTR | 停止軸パターンを指定します. |
| 1 | VT_UI2 | 付加コマンドバイトを指定します. |

戻り値：なし

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("STOP_CANCEL ", Array("1:2:3", 2))
```

#### 2.3.2.25. CaoController::Execute("WRITE_POSITION_DATA_RANGE")コマンド

ポジションデータ範囲指定連続書込みを実行します.

引数：

| | | | | |
|---|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | | |
| 0 | VT_UI2 | | | 変更開始ポジションデータ No. を指定します. |
| 1 | VT_UI2 | | | 変更ポジションデータ数を指定します. |
| 2 | VT_ARRAY \| VT_VARIANT | | | |
| | 0 | VT_BSTR | | 軸パターンを指定します. |
| | 1 | VT_UI2 | | 加速度を指定します. |
| | 2 | VT_UI2 | | 減速度を指定します. |
| | 3 | VT_UI2 | | 速度を指定します. |
| | 4 | VT_ARRAY \| VT_I4 | | |
| | | n | VT_I4 | 位置データを指定します. |

戻り値：

| | | |
|---|---|---|
| VT_ARRAY \| VT_UI2 | | |
| 0 | VT_UI2 | 変更開始ポジションデータ No. が返却されます. |
| 1 | VT_UI2 | 変更完了ポジションデータ数が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("WRITE_POSITION_DATA_RANGE ", Array(1, 2, Array("1:2:3", 4, 5,
6, Array(10, 20)))
```

### 2.3.2.26. CaoController::Execute("WRITE_CHANGE_POSITION_DATA ")コマンド

変更ポジションデータ連続書込みを実行します.

引数：

| | | | | |
|---|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | | |
| 0 | VT_UI2 | | | 変更ポジションデータ数を指定します. |
| 1 | VT_ARRAY \| VT_VARIANT | | | |
| | 0 | VT_UI2 | | 変更ポジションデータ No. を指定します. |
| | 1 | VT_BSTR | | 軸パターンを指定します. |
| | 2 | VT_UI2 | | 加速度を指定します. |
| | 3 | VT_UI2 | | 減速度を指定します. |
| | 4 | VT_UI2 | | 速度を指定します. |
| | 5 | VT_ARRAY \| VT_I4 | | |
| | | n | VT_I4 | 位置データを指定します. |

戻り値：

| | |
|---|---|
| VT_UI2 | 変更完了ポジションデータ数が返却されます. |

使用例：

```
Dim lVal As Long
lVal = controller.Execute("WRITE_CHANGE_POSITION_DATA", Array(1, Array(2, "1:2:3", 4, 5, 6, Array(10, 20))))
```

### 2.3.2.27. CaoController::Execute("CLEAR_POSITION_DATA ")コマンド

ポジションデータクリアを実行します.

引数：

| | | |
|---|---|---|
| VT_ARRAY \| VT_UI2 | | |
| 0 | VT_UI2 | クリア開始ポジションデータ No. を指定します. |
| 1 | VT_UI2 | クリアポジションデータ数を指定します. |

戻り値：なし

使用例：

```
controller.Execute("CLEAR_POSITION_DATA ", Array(1, 2))
```

#### 2.3.2.28. CaoController::Execute("CHANGE_OUTPUT_PORT_STATUS ")コマンド

出力ポート状態変更を実行します.

引数：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| 0 | VT_UI2 | 出力ポート No.を指定します. |
| 1 | VT_UI2 | 変更種別を指定します. |

戻り値：なし

使用例：

```
controller.Execute("CHANGE_OUTPUT_PORT_STATUS ", Array(1, 1))
```

#### 2.3.2.29. CaoController::Execute("CHANGE_FLUG_STATUS ")コマンド

フラグ状態変更を実行します.

引数：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| 0 | VT_UI2 | プログラム No.を指定します. |
| 1 | VT_UI2 | フラグ No.を指定します. |
| 2 | VT_UI2 | 変更種別を指定します. |

戻り値：なし

使用例：

```
controller.Execute("CHANGE_FLUG_STATUS ", Array(1, 2, 1))
```

### 2.3.2.30. CaoController::Execute("CHANGE_INTEGER_VARIABLE")コマンド

整数変数変更を実行します.

引数：

| | | | | |
|---|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | | |
| 0 | VT_UI2 | | | プログラム No. を指定します. |
| 1 | VT_UI2 | | | 変更開始変数 No. を指定します. |
| 2 | VT_UI2 | | | 変更変数データ数を指定します. |
| 3 | VT_ARRAY \| VT_I4 | | | |
| | | n | VTI4 | 整数変数データを指定します. |

戻り値：

| | | |
|---|---|---|
| VT_ARRAY \| VT_UI2 | | |
| 0 | VT_UI2 | プログラム No. が返却されます. |
| 1 | VT_UI2 | 変更開始変数 No. が返却されます. |
| 2 | VT_UI2 | 変更完了データ数が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("CHANGE_INTEGER_VARIABLE", Array(1, 2, 3, Array(10, 20, 30)))
```

### 2.3.2.31. CaoController::Execute("CHANGE_REAL_VARIABLE ")コマンド

実数変数変更を実行します.

引数：

| VT_ARRAY \| VT_VARIANT | | | | |
|---|---|---|---|---|
| 0 | VT_UI2 | | | プログラム No. を指定します. |
| 1 | VT_UI2 | | | 変更開始変数 No. を指定します. |
| 2 | VT_UI2 | | | 変更変数データ数を指定します. |
| 3 | VT_ARRAY \| VT_R8 | | | |
| | | n | VT_R8 | 実数変数データを指定します. |

戻り値：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| 0 | VT_UI2 | プログラム No. が返却されます. |
| 1 | VT_UI2 | 変更開始変数 No. が返却されます. |
| 2 | VT_UI2 | 変更完了データ数が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("CHANGE_REAL_VARIABLE ", Array(1, 2, 3, Array(10, 20, 30)))
```

### 2.3.2.32. CaoController::Execute("CHANGE_STRING_VARIABLE")コマンド

ストリング変数変更を実行します.

引数：

| VT_ARRAY \| VT_VARIANT | | |
|---|---|---|
| 0 | VT_UI2 | プログラム No. を指定します. |
| 1 | VT_UI2 | 変更開始変数 No. を指定します. |
| 2 | VT_UI2 | 変更変数データ数を指定します. |
| 3 | VT_BSTR | ストリング変数データを指定します. |

戻り値：なし

使用例：

```
controller.Execute("CHANGE_STRING_VARIABLE", Array(1, 2, 3, "ABC"))
```

#### 2.3.2.33. CaoController::Execute("RESET_ALARM ")コマンド

アラームリセットを実行します.

引数：なし
戻り値：なし
使用例：

```
controller.Execute("RESET_ALARM ")
```

#### 2.3.2.34. CaoController::Execute("EXECUTION_PROGRAM ")コマンド

プログラム実行を実行します.

引数：

| | |
|---|---|
| VT_UI2 | プログラム No. を指定します. |

戻り値：なし
使用例：

```
controller.Execute("EXECUTION_PROGRAM ", 1)
```

#### 2.3.2.35. CaoController::Execute("END_PROGRAM ")コマンド

プログラム終了を実行します.

引数：

| | |
|---|---|
| VT_UI2 | プログラム No. を指定します. |

戻り値：なし
使用例：

```
controller.Execute("END_PROGRAM ", 1)
```

#### 2.3.2.36. CaoController::Execute("PAUSE_PROGRAM ")コマンド

プログラム一時停止を実行します.

引数：

| | |
|---|---|
| VT_UI2 | プログラム No. を指定します. |

戻り値：なし
使用例：

```
controller.Execute("PAUSE_PROGRAM ", 1)
```

#### 2.3.2.37. CaoController::Execute("EXECUTION_ONE_STEP_PROGRAM ")コマンド

プログラム１ステップ実行を実行します.

引数：

| | |
|---|---|
| VT_UI2 | プログラム No.を指定します. |

戻り値：なし

使用例：

```
controller.Execute("EXECUTION_ONE_STEP_PROGRAM", 1)
```

#### 2.3.2.38. CaoController::Execute("EXECUTION_RESUME_PROGRAM ")コマンド

プログラム実行再開を実行します.

引数：

| | |
|---|---|
| VT_UI2 | プログラム No.を指定します. |

戻り値：なし

使用例：

```
controller.Execute("EXECUTION_RESUME_PROGRAM ", 1)
```

#### 2.3.2.39. CaoController::Execute("RESET_SOFTWARE ")コマンド

ソフトウェアリセットを実行します.

引数：なし

戻り値：なし

使用例：

```
controller.Execute("RESET_SOFTWARE ")
```

#### 2.3.2.40. CaoController::Execute("REQUEST_RESTORATION_DRIVE ")コマンド

駆動源復旧要求を実行します.

引数：なし

戻り値：なし

使用例：

```
controller.Execute("REQUEST_RESTORATION_DRIVE ")
```

#### 2.3.2.41. CaoController::Execute("REQUEST_PAUSE_RELEASE_OPERATION ")コマンド

動作一時停止解除要求を実行します.

引数：なし

戻り値：なし

使用例：

```
controller.Execute("REQUEST_PAUSE_RELEASE_OPERATION ")
```

#### 2.3.2.42. CaoController::Execute("CHANGE_SPEED")コマンド

速度チェンジを実行します.

引数：

| VT_ARRAY \| VT_VARIANT | | | |
|---|---|---|---|
| | 0 | VT_BSTR | 軸パターンを指定します. |
| | 1 | VT_UI2 | 速度を指定します. |

戻り値：なし

使用例：

```
controller.Execute("CHANGE_SPEED", Array("1:2:3", 2))
```

#### 2.3.2.43. CaoController::Execute("MOVE_POSITION_NO2")コマンド

ポジション No.指定移動 2 を実行します.

引数：

| VT_ARRAY \| VT_VARIANT | | | |
|---|---|---|---|
| | 0 | VT_BSTR | 軸パターンを指定します. |
| | 1 | VT_UI2 | 加速度を指定します. |
| | 2 | VT_UI2 | 減速度を指定します. |
| | 3 | VT_UI2 | 速度を指定します. |
| | 4 | VT_UI2 | ポジション No.を指定します. |

戻り値：なし

使用例：

```
controller.Execute("MOVE_POSITION_NO2", Array("1:2:3", 2, 3, 4, 5))
```

### 2.3.2.44. CaoController::Execute("WRITE_POSITION_DATA_RANGE2")コマンド

ポジションデータ範囲指定連続書込み 2 を実行します.

引数：

| | | | | | | |
|---|---|---|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | | | | |
| 0 | VT_UI2 | | | | | 変更開始ポジションデータ No. を指定します. |
| 1 | VT_UI2 | | | | | 変更ポジションデータ数を指定します. |
| 2 | VT_ARRAY \| VT_VARIANT | | | | | |
| | 0 | VT__BSTR | | | | 軸パターンを指定します. |
| | 1 | VT_UI2 | | | | 加速度を指定します. |
| | 2 | VT_UI2 | | | | 減速度を指定します. |
| | 3 | VT_UI2 | | | | 速度を指定します. |
| | 4 | VT_ARRAY \| VT_I4 | | | | |
| | | | n | VT_I4 | | 位置データを指定します. |

戻り値：なし

使用例：

```
controller.Execute("WRITE_POSITION_DATA_RANGE2", Array(1, 1, Array("1:2:3", 2, 3, 4,
Array(10, 20))))
```

### 2.3.2.45. CaoController::Execute("WRITE_CHANGE_POSITION_DATA2")コマンド

変更ポジションデータ連続書込み 2 を実行します.

引数：

| | | | | |
|---|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | | |
| 0 | VT_UI2 | | | 変更ポジションデータ数を指定します. |
| 1 | VT_ARRAY \| VT_VARIANT | | | |
| | 0 | VT_UI2 | | 変更ポジションデータ No.を指定します. |
| | 1 | VT_BSTR | | 軸パターンを指定します. |
| | 2 | VT_UI2 | | 加速度を指定します. |
| | 3 | VT_UI2 | | 減速度を指定します. |
| | 4 | VT_UI2 | | 速度を指定します. |
| | 5 | VT_ARRAY \| VT_I4 | | |
| | | n | VT_I4 | 位置データを指定します. |

戻り値：

| | |
|---|---|
| VT_UI2 | 変更完了ポジションデータ数が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("WRITE_CHANGE_POSITION_DATA2", Array(1, 2))
```

### 2.3.2.46. CaoController::Execute("CLEAR_POSITION_DATA2")コマンド

ポジションデータクリア 2 を実行します.

引数：

| | | |
|---|---|---|
| VT_ARRAY \| VT_UI2 | | |
| 0 | VT_UI2 | クリア開始ポジションデータ No.を指定します. |
| 1 | VT_UI2 | クリアポジションデータ数を指定します. |

戻り値:なし

使用例：

```
call controller.Execute("CLEAR_POSITION_DATA2", Array(1,1))
```

### 2.3.2.47. CaoController::Execute("CONTROLLER_FUNCTION2")コマンド

コントローラ機能指定 2 を実行します.

引数：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| 0 | VT_UI2 | コントローラ機能指定ワード 9 を指定します. |
| 1 | VT_UI2 | コントローラ機能指定ワード 10 を指定します. |
| 2 | VT_UI2 | コントローラ機能指定ワード 11 を指定します. |
| 3 | VT_UI2 | コントローラ機能指定ワード 12 を指定します. |
| 4 | VT_UI2 | コントローラ機能指定ワード 13 を指定します. |
| 5 | VT_UI2 | コントローラ機能指定ワード 14 を指定します. |
| 6 | VT_UI2 | コントローラ機能指定ワード 15 を指定します. |
| 7 | VT_UI2 | コントローラ機能指定ワード 16 を指定します. |

戻り値：なし

使用例：

```
controller.Execute("CONTROLLER_FUNCTION2", Array(1, 2, 3, 4, 5, 6, 7, 8))
```

### 2.3.2.48. CaoController::Execute("WRITE_POSITION_DATA_RANGE3")コマンド

ポジションデータ範囲指定連続書込み 3 を実行します.

引数：

| | | | |
|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | |
| 0 | VT_UI2 | | レコードフォーマット拡張指定種別を指定します. |
| 1 | VT_UI2 | | 変更開始ポジションデータ No. を指定します. |
| 2 | VT_UI2 | | 変更ポジションデータ数を指定します. |
| 3 | VT_ARRAY \| VT_VARIANT | | |
| | 0 | VT_UI2 | 軸パターンを指定します. |
| | 1 | VT_UI2 | 加速度を指定します. |
| | 2 | VT_UI2 | 減速度を指定します. |
| | 3 | VT_UI2 | 速度を指定します. |
| | 4 | VT_ARRAY \| VT_I4 | |
| | | n VT_I4 | 位置データを指定します. |
| | 5 | VT_I4 | 拡張データを指定します. |
| | 6 | VT_I4 | 出力ファンクションコードを指定します. |
| | 7 | VT_I4 | 出力ポート・フラグ No. を指定します. |
| | 8 | VT_I4 | ファンクションパラメータ 1 を指定します. |
| | 9 | VT_I4 | ファンクションパラメータ 2 を指定します. |

戻り値：

| | | |
|---|---|---|
| VT_ARRAY \| VT_UI2 | | |
| 0 | VT_UI2 | レコードフォーマット拡張指定種別が返却されます. |
| 1 | VT_UI2 | 変更開始ポジションデータ No. が返却されます. |
| 2 | VT_UI2 | 変更完了ポジションデータ数が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("WRITE_POSITION_DATA_RANGE3", Array(1 ,2, 1, Array("1:2:3", 2, 3, 4, Array(10, 20), 1, 2, 3, 4, 5)))
```

### 2.3.2.49. CaoController::Execute("WRITE_CHANGE_POSITION_DATA3")コマンド

変更ポジションデータ連続書込み 3 を実行します.

引数：

| | | | |
|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | |
| 0 | VT_UI2 | | レコードフォーマット拡張指定種別を指定します. |
| 1 | VT_UI2 | | 変更ポジションデータ数を指定します. |
| 2 | VT_ARRAY \| VT_VARIANT | | |
| | 0 | VT_UI2 | 変更ポジションデータ No. を指定します. |
| | 1 | VT_UI2 | 軸パターンを指定します. |
| | 2 | VT_UI2 | 加速度を指定します. |
| | 3 | VT_UI2 | 減速度を指定します. |
| | 4 | VT_UI2 | 速度を指定します. |
| | 5 | VT_ARRAY \| VT_I4 | |
| | | n VT_I4 | 位置データを指定します. |
| | 6 | VT_I4 | 拡張データを指定します. |
| | 7 | VT_I4 | 出力ファンクションコードを指定します. |
| | 8 | VT_I4 | 出力ポート・フラグ No. を指定します. |
| | 9 | VT_I4 | ファンクションパラメータ 1 を指定します. |
| | 10 | VT_I4 | ファンクションパラメータ 2 を指定します. |

戻り値：

| | | |
|---|---|---|
| VT_ARRAY \| VT_UI2 | | |
| 0 | VT_UI2 | レコードフォーマット拡張指定種別が返却されます. |
| 1 | VT_UI2 | 変更完了ポジションデータ数が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("WRITE_CHANGE_POSITION_DATA3", Array(1, 1, Array(1, "1:2:3", 2, 3, 4, Array(10, 20), 1, 2, 3, 4, 5))
```

### 2.3.2.50. CaoController::Execute("COORDINATE_DATA_RANGE ")コマンド

座標系定義データ範囲指定連続の取得を実行します.

引数：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| 0 | VT_UI2 | 種別を指定します. |
| 1 | VT_UI2 | 照会先先頭座標系定義データ No.を指定します. |
| 2 | VT_UI2 | 照会レコード数を指定します. |

戻り値：

| VT_ARRAY \| VT_VARIANT | | | |
|---|---|---|---|
| 0 | VT_ARRAY \| VT_I4 | | |
| | 0 | VT_I4 | X 座標オフセット量が返却されます. |
| | 1 | VT_I4 | Y 座標オフセット量が返却されます. |
| | 2 | VT_I4 | Z 座標オフセット量が返却されます. |
| | 3 | VT_I4 | R 座標オフセット量が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("COORDINATE_DATA_RANGE ", Array(1, 2, 3))
```

### 2.3.2.51. CaoController::Execute("UNIT_AXIS_STATUS ")コマンド

ユニット軸ステータスの取得を実行します.

引数：

| VT_ARRAY \| VT_VARIANT | | |
|---|---|---|
| 0 | VT_BSTR | 照会軸パターンを指定します. |
| 1 | VT_UI2 | 種別を指定します. |

戻り値：

| VT_ARRAY \| VT_VARIANT | | | |
|---|---|---|---|
| 0 | VT_UI2 | | ワーク座標系選択 No.が返却されます. |
| 1 | VT_UI2 | | ツール座標系選択 No.が返却されます. |
| 2 | VT_UI2 | | 軸共通ステータスが返却されます. |
| 3 | VT_ARRAY \| VT_I4 | | |
| | 0 | VT_I4 | 軸ステータスが返却されます. |
| | 1 | VT_I4 | 軸センサー入力ステータスが返却されます. |
| | 2 | VT_I4 | 軸関連エラーコードが返却されます. |
| | 3 | VT_I4 | エンコーダステータス（ リセット時）が返却されます. |
| | 4 | VT_I4 | 現在位置が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("UNIT_AXIS_STATUS ", Array("1:2:3", 1))
```

### 2.3.2.52. CaoController::Execute("CHECK_ZONE_DATA_RANGE ")コマンド

簡易干渉チェックゾーン定義データ範囲指定連続の取得を実行します.

引数：

| | | | |
|---|---|---|---|
| VT_ARRAY \| VT_UI2 | | | |
| | 0 | VT_UI2 | 照会先頭簡易干渉チェックゾーン定義データ No.を指定します. |
| | 1 | VT_UI2 | 照会レコード数を指定します. |
| | 2 | VT_UI2 | 有効軸数を指定します. |

戻り値：

| | | | | |
|---|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | | |
| | 0 | VT_ARRAY \| VT_VARIANT | | |
| | | 0 | VT_UI2 | 簡易干渉チェックゾーン定義座標有効軸パターンが返却されます. |
| | | 1 | VT_ARRAY \| VT_I4 | |
| | | | n VT_I4 | 簡易干渉チェックゾーン定義座標 1 が返却されます. |
| | | 2 | VT_ARRAY \| VT_I4 | |
| | | | n VT_I4 | 簡易干渉チェックゾーン定義座標 2 が返却されます. |
| | | 3 | VT_UI2 | 侵入時出力物理的出力ポート No. or グローバルフラグ No.が返却されます. |
| | | 4 | VT_UI2 | 侵入時エラー種別指定が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("CHECK_ZONE_DATA_RANGE", Array(1, 2, 3))
```

### 2.3.2.53. CaoController::Execute("UNIT_AXIS_STATUS2")コマンド

ユニット軸ステータス 2 の取得を実行します.

引数：

| VT_ARRAY \| VT_VARIANT | | |
|---|---|---|
| 0 | VT_BSTR | 照会軸パターンを指定します. |
| 1 | VT_UI2 | 種別を指定します. |

戻り値：

| VT_ARRAY \| VT_VARIANT | | | |
|---|---|---|---|
| 0 | VT_BSTR | | ユニット軸パターンが返却されます. |
| 1 | VT_UI2 | | ワーク座標系選択 No.(ユニット 1)が返却されます. |
| 2 | VT_UI2 | | ツール座標系選択 No.(ユニット 1)が返却されます. |
| 3 | VT_UI2 | | ユニット 1 共通ステータスが返却されます. |
| 4 | VT_UI2 | | ワーク座標系選択 No.(ユニット 2)が返却されます. |
| 5 | VT_UI2 | | ツール座標系選択 No.(ユニット 2)が返却されます. |
| 6 | VT_UI2 | | ユニット 2 共通ステータスが返却されます. |
| 7 | VT_ARRAY \| VT_I4 | | |
| | 0 | VT_I4 | 軸ステータスが返却されます. |
| | 1 | VT_I4 | 軸センサー入力ステータスが返却されます. |
| | 2 | VT_I4 | 軸関連エラーコードが返却されます. |
| | 3 | VT_I4 | エンコーダステータス（ リセット時）が返却されます. |
| | 4 | VT_I4 | 現在位置が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("UNIT_AXIS_STATUS2", Array("1:2:3", 1))
```

#### 2.3.2.54. CaoController::Execute("COORDINATE_DATA_RANGE2")コマンド

座標系定義データ範囲指定連続 2 の取得を実行します.

引数：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| 0 | VT_BSTR | 照会軸パターンを指定します. |
| 1 | VT_UI2 | 種別を指定します. |
| 2 | VT_UI2 | 照会先先頭座標系定義データ No. を指定します. |
| 3 | VT_UI2 | 照会レコード数を指定します. |

戻り値：

| VT_ARRAY \| VT_VARIANT | | | | |
|---|---|---|---|---|
| 0 | VT_ARRAY \| VT_I4 | | | |
| | 0 | VT_I4 | | X 座標オフセット量が返却されます. |
| | 1 | VT_I4 | | Y 座標オフセット量が返却されます. |
| | 2 | VT_I4 | | Z 座標オフセット量が返却されます. |
| | 3 | VT_I4 | | R 座標オフセット量が返却されます. |

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("COORDINATE_DATA_RANGE2", Array("1:2:3", 1, 2, 3))
```

#### 2.3.2.55. CaoController::Execute("MOVE_UNIT_ABSOLUIT ")コマンド

ユニット絶対座標指定移動を実行します.

引数：

| VT_ARRAY \| VT_VARIANT | | | |
|---|---|---|---|
| 0 | VT_BSTR | | 軸パターンを指定します. |
| 1 | VT_UI2 | | 加速度を指定します. |
| 2 | VT_UI2 | | 減速度を指定します. |
| 3 | VT_UI2 | | 速度を指定します. |
| 4 | VT_UI2 | | 位置決め動作種別を指定します. |
| 5 | VT_ARRAY \| VT_I4 | | |
| | n | VT_I4 | 絶対座標データを指定します. |

戻り値：なし

使用例：

```
controller.Execute("MOVE_UNIT_ABSOLUIT", Array("1:2:3", 2, 3, 4, 1, Array(10, 20)))
```

### 2.3.2.56. CaoController::Execute("MOVE_UNIT_RELATIVE ")コマンド

ユニット相対座標指定移動を実行します.

引数：

| | | | |
|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | |
| 0 | VT_BSTR | | 軸パターンを指定します. |
| 1 | VT_UI2 | | 加速度を指定します. |
| 2 | VT_UI2 | | 減速度を指定します. |
| 3 | VT_UI2 | | 速度を指定します. |
| 4 | VT_UI2 | | 位置決め動作種別を指定します. |
| 5 | VT_ARRAY \| VT_I4 | | |
| | n | VT_I4 | 相対座標データを指定します. |

戻り値：

使用例：

```
controller.Execute("MOVE_UNIT_RELATIVE", Array("1:2:3", 2, 3, 4, 1, Array(10, 20)))
```

### 2.3.2.57. CaoController::Execute("MOVE_UNIT_POSITION_NO")コマンド

ユニットポジション No.指定移動を実行します.

引数：

| | | |
|---|---|---|
| VT_ARRAY \| VT_VARIANT | | |
| 0 | VT_BSTR | 軸パターンを指定します. |
| 1 | VT_UI2 | 加速度を指定します. |
| 2 | VT_UI2 | 減速度を指定します. |
| 3 | VT_UI2 | 速度を指定します. |
| 4 | VT_UI2 | 位置決め動作種別を指定します. |
| 5 | VT_UI2 | ポジション No.を指定します. |

戻り値：なし

使用例：

```
Dim vArray As Variant
vArray = controller.Execute("MOVE_UNIT_POSITION_NO ", Array("1:2:3", 2, 3, 4, 5, 6))
```

#### 2.3.2.58. CaoController::Execute("MOVE_UNIT_POSITION_NO2")コマンド

ユニットポジション No.指定移動 2 を実行します.

引数：

| VT_ARRAY \| VT_VARIANT | | |
|---|---|---|
| 0 | VT_BSTR | 軸パターンを指定します. |
| 1 | VT_UI2 | 加速度を指定します. |
| 2 | VT_UI2 | 減速度を指定します. |
| 3 | VT_UI2 | 速度を指定します. |
| 4 | VT_UI2 | 位置決め動作種別を指定します. |
| 5 | VT_UI2 | ポジション No.を指定します. |

戻り値：なし

使用例：

```
controller.Execute("MOVE_UNIT_POSITION_NO2", Array("1:2:3", 2, 3, 4, 1, 5))
```

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

CaoController から Variable オブジェクトを生成する際に，変数名を指定することによって，接続した SEL デバイスから読み書きするデータを決定します.

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
)
```

使用できる変数名，オプション，詳細に付いては表 2-4 を参照して下さい.

### 2.3.4. CaoController::get_VariableNames メソッド

表 2-4 の変数名リストを取得します.

### 2.3.5. CaoVariable::get_Value プロパティ

指定したオプションによって，SEL デバイスからデータを取得します.

### 2.3.6. CaoVariable::set_Value プロパティ

指定したオプションによって，SEL デバイスへデータを書き込みます.

## 2.4. 変数一覧

### 2.4.1. コントローラクラス

以下の表 2-4 にコントローラクラスの AddVariable にて使用できる変数一覧を記述します.

**表 2-4 コントローラクラス 変数一覧**

| 変数名 | 説明 | Value | | リンク |
|---|---|---|---|---|
| | | get | put | |
| @MAKER_NAME | メーカー名を取得します. | ○ | - | P.49 |
| @VERSION | バージョンを取得します. | ○ | - | P.49 |
| VERSION_CODE<??> | バージョンコードを取得します. | ○ | - | P.49 |
| @POSITION_DATA_NUM | 有効ポジションデータ数を取得します. | ○ | - | P.50 |
| POSITION_DATA<??> | 有効ポジションデータの取得と設定します. | ○ | ○ | P.50 |
| INPUT_PORT<??> | 入力ポートを取得します. | ○ | - | P.51 |
| OUTPUT_PORT<??> | 出力ポートを取得します. | ○ | - | P.51 |
| FLAG<??> | フラグを取得します. | ○ | - | P.52 |
| INTEGER_VARIABLE<??> | 整数変数の取得と設定します. | ○ | ○ | P.53 |
| REAL_VARIABLE<??> | 実数変数の取得と設定します. | ○ | ○ | P.54 |
| STRING_VARIABLE<??> | ストリング変数の取得と設定します. | ○ | ○ | P.54 |
| AXIS_STATUS<??> | 軸ステータスを取得します. | ○ | - | P.55 |
| PROGRAM_STATUS<??> | プログラムステータスを取得します. | ○ | - | P.55 |
| @SYSTEM_STATUS | システムステータスを取得します. | ○ | - | P.56 |
| ERROR_INFO<??> | エラー詳細情報を取得します. | ○ | - | P.57 |
| @EX2_POSITION_DATA_NUM | 有効ポジションデータ数 2 を取得します. | ○ | - | P.58 |
| EX2_POSITION_DATA<??> | 有効ポジションデータ 2 の取得と設定します. | ○ | ○ | P.58 |
| EX3_POSITION_DATA<??> | 有効ポジションデータ 3 の取得と設定します. | ○ | ○ | P.59 |
| COORDINATE_DATA_RANGE<??> | 座標系定義データ範囲指定連続を取得します. | ○ | - | P.60 |
| UNIT_AXIS_STATUS<??> | ユニット軸ステータスを取得します. | ○ | - | P.61 |
| CHECK_ZONE_DATA_RANGE<??> | 簡易干渉チェックゾーン定義データ範囲指定連続を取得します. | ○ | - | P.62 |
| EX2_UNIT_AXIS_STATUS<??> | ユニット軸ステータス 2 を取得します. | ○ | - | P.63 |
| EX2_COORDINATE_DATA_RANGE<??> | 座標系定義データ範囲指定連続 2 を取得します. | ○ | - | P.64 |

※<??>は任意の文字

#### 2.4.1.1. @MAKER_NAME

メーカー名を取得します.

オプション：なし

get_value 時の戻り値のデータ構成：

| | |
|---|---|
| VT_BSTR | 固定値："IAI" |

#### 2.4.1.2. @VERSION

バージョンを取得します.

オプション：なし

get_value 時の戻り値のデータ構成：

| | |
|---|---|
| VT_BSTR | 現在の DLL のバージョンを表す"*.*.*"文字列 |

#### 2.4.1.3. VERSION_CODE<??>

バージョンコードを取得します.

オプション：

| | | |
|---|---|---|
| UNIT_TYPE | ユニット種別を指定します. | 0 - 3 |
| DEVICE_NO | デバイス No.を指定します. | 0 - |

get_value 時の戻り値のデータ構成：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| 0 | VT_UI2 | 機種コード |
| 1 | VT_UI2 | ユニットコード |
| 2 | VT_UI2 | バージョン No. |
| 3 | VT_UI2 | 時刻 |
| 4 | VT_UI2 | 時刻 |
| 5 | VT_UI2 | 時刻 |
| 6 | VT_UI2 | 時刻 |
| 7 | VT_UI2 | 時刻 |
| 8 | VT_UI2 | 時刻 |

#### 2.4.1.4. @POSITION_DATA_NUM

有効ポジションデータ数を取得します.

オプション：なし

get_value 時の戻り値のデータ構成：

| | |
|---|---|
| VT_UI2 | 有効ポジションデータ数 |

#### 2.4.1.5. POSITION_DATA<??>

有効ポジションデータの取得と設定します.

オプション：

| | | |
|---|---|---|
| POSITION_NO | 先頭ポジション No. を指定します. | 0 - |
| RECORD_NUM | レコード数を指定します. | 1 - |
| AXIS_NUM | 有効軸数を指定します. | 1 - 8 |

get_value 時の戻り値のデータ構成：

| | | | | | |
|---|---|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | | | |
| 0 | VT_ARRAY \| VT_VARIANT | | | | |
| | 0 | VT_UI2 | | | ポジション No. |
| | 1 | VT_BSTR | | | 軸パターン |
| | 2 | VT_UI2 | | | 加速度 |
| | 3 | VT_UI2 | | | 減速度 |
| | 4 | VT_UI2 | | | 速度 |
| | 5 | VT_ARRAY \| VT_I4 | | | |
| | | | n | VT_I4 | 位置データ |

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

#### 2.4.1.6. INPUT_PORT<??>

入力ポートを取得します.

オプション:

| | | |
|---|---|---|
| PORT_NO | 照会開始ポート No. を指定します. | 0 - |
| PORT_NUM | 照会ポート数を指定します. | 1 - |

get_value 時の戻り値のデータ構成:

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| n | VT_UI2 | 入力ポートデータ |

#### 2.4.1.7. OUTPUT_PORT<??>

出力ポートを取得します.

オプション:

| | | |
|---|---|---|
| PORT_NO | 照会開始ポート No. を指定します. | 0 - |
| PORT_NUM | 照会ポート数を指定します. | 1 - |

get_value 時の戻り値のデータ構成:

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| n | VT_UI2 | 出力ポートデータ |

#### 2.4.1.8. FLAG<??>

フラグを取得します.

オプション：

| | | |
|---|---|---|
| PROGRAM_NO | プログラム No. を指定します. | 0 - |
| START_NO | 開始変数 No. を指定します. | 0 - |
| DATA_NUM | フラグ数を指定します. | 1 - |
| ARRAY | 配列有無を指定します.<br>(true かつ フラグ数=1 の場合, VT_UI2 で返却) | true / false<br>(true かつ 変数データ数=1 の場合,整数変数データを VT_UI2 で返却) |

get_value 時の戻り値のデータ構成：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| n | VT_UI2 | フラグデータ |

#### 2.4.1.9. INTEGER_VARIABLE<??>

整数変数の取得と設定します.

オプション：

| | | |
|---|---|---|
| PROGRAM_NO | プログラム No. を指定します. | 0 - |
| START_NO | 開始変数 No. を指定します. | 0 - |
| DATA_NUM | 変数データ数を指定します. | 1 - |
| ARRAY | 配列有無を指定します.<br>(true かつ フラグ数=1 の場合, VT_UI2 で返却) | true / false<br>(true かつ 変数データ数=1 の場合,整数変数データを VT_UI2 で返却) |

get_value 時の戻り値のデータ構成：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| n | VT_UI2 | 整数変数データ |

put_value 時の設定値のデータ構成

get_value の戻り値のデータ構成と同じ

#### 2.4.1.10. REAL_VARIABLE<??>

実数変数の取得と設定します.

オプション：

| | | |
|---|---|---|
| PROGRAM_NO | プログラム No. を指定します. | 0 - |
| START_NO | 開始変数 No. を指定します. | 0 - |
| DATA_NUM | 変数データ数を指定します. | 1 - |
| ARRAY | 配列有無を指定します.<br>(true かつ フラグ数=1 の場合, VT_UI2 で返却) | true / false<br>(true かつ 変数データ数=1 の場合,整数変数データを VT_UI2 で返却) |

get_value 時の戻り値のデータ構成：

| VT_ARRAY \| VT_R8 | | |
|---|---|---|
| n | VT_R8 | 実数変数データ |

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

#### 2.4.1.11. STRING_VARIABLE<??>

ストリング変数の取得と設定します.

オプション：

| | | |
|---|---|---|
| PROGRAM_NO | プログラム No. を指定します. | 0 - |
| START_NO | 開始変数 No. を指定します. | 0 - |
| DATA_NUM | 変数データ数を指定します. | 1 - |

get_value 時の戻り値のデータ構成：

| | |
|---|---|
| VT_BSTR | ストリング変数データ |

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

#### 2.4.1.12. AXIS_STATUS<??>

軸ステータスを取得します.

オプション：

| | | |
|---|---|---|
| AXIS_PATTERN | 照会軸パターンを指定します. | 1 - 8 |

get_value 時の戻り値のデータ構成：

| | | | |
|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | |
| 0 | VT_ARRAY \| VT_I4 | | |
| | 0 | VT_I4 | 軸ステータス |
| | 1 | VT_I4 | 軸センサー入力ステータス |
| | 2 | VT_I4 | 軸関連エラーコード |
| | 3 | VT_I4 | エンコーダステータス<br>（リセット時） |
| | 4 | VT_I4 | 現在位置 |

#### 2.4.1.13. PROGRAM_STATUS<??>

プログラムステータスを取得します.

オプション：

| | | |
|---|---|---|
| PROGRAM_NO | プログラム No. を指定します. | 0 - |

get_value 時の戻り値のデータ構成：

| | | |
|---|---|---|
| VT_ARRAY \| VT_UI2 | | |
| 0 | VT_UI2 | ステータス |
| 1 | VT_UI2 | 実行中プログラムステップ No. |
| 2 | VT_UI2 | プログラム依存エラーコード |
| 3 | VT_UI2 | エラー発生ステップ No. |

#### 2.4.1.14. @SYSTEM_STATUS

システムステータスを取得します.

オプション：なし

get_value 時の戻り値のデータ構成：

| VT_ARRAY \| VT_UI2 | | |
|---|---|---|
| 0 | VT_UI2 | システムモード |
| 1 | VT_UI2 | 最重レベルシステム<br>エラーNo. |
| 2 | VT_UI2 | 最新システム<br>エラーNo. |
| 3 | VT_UI2 | システムステータス<br>バイト 1 |
| 4 | VT_UI2 | システムステータス<br>バイト 2 |
| 5 | VT_UI2 | システムステータス<br>バイト 3 |
| 6 | VT_UI2 | システムステータス<br>バイト 4 |

#### 2.4.1.15. ERROR_INFO<??>

エラー詳細情報を取得します.

オプション：

| | | |
|---|---|---|
| TYPE1 | 種別 1 を指定します. | 0 - |
| TYPE2 | 種別 2 を指定します. | 0 - |
| RECORD_NO | レコード No. を指定します. | 0 - |

get_value 時の戻り値のデータ構成：

| VT_ARRAY \| VT_VARIANT | | |
|---|---|---|
| 0 | VT_UI4 | エラーNo. |
| 1 | VT_UI4 | 詳細情報 1 |
| 2 | VT_UI4 | 詳細情報 2 |
| 3 | VT_UI4 | 詳細情報 3 |
| 4 | VT_I4 | 詳細情報 4 |
| 5 | VT_UI4 | 詳細情報 5 |
| 6 | VT_UI4 | 詳細情報 6 |
| 7 | VT_UI4 | 詳細情報 7 |
| 8 | VT_UI4 | 詳細情報 8 |
| 9 | VT_BSTR | メッセージ文字列 |

#### 2.4.1.16. @EX2_POSITION_DATA_NUM

有効ポジションデータ数 2 を取得します.

オプション：なし

get_value 時の戻り値のデータ構成：

| | |
|---|---|
| VT_UI2 | 有効ポジションデータ数 |

#### 2.4.1.17. EX2_POSITION_DATA<??>

有効ポジションデータ 2 の取得と設定します.

オプション：

| | | |
|---|---|---|
| POSITION_NO | 先頭ポジション No. を指定します. | 0 - |
| RECORD_NUM | レコード数を指定します. | 1 - |
| AXIS_NUM | 有効軸数を指定します. | 1 - 8 |

get_value 時の戻り値のデータ構成：

| | | | |
|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | |
| 0 | VT_ARRAY \| VT_VARIANT | | |
| | 0 | VT_UI2 | ポジション No. |
| | 1 | VT_UI2 | 軸パターン |
| | 2 | VT_UI2 | 加速度 |
| | 3 | VT_UI2 | 減速度 |
| | 4 | VT_UI2 | 速度 |
| | 5 | VT_ARRAY \| VT_I4 | |
| | | n VT_I4 | 位置データ |

put_value 時の設定値のデータ構成

get_value の戻り値のデータ構成と同じ

### 2.4.1.18. EX3_POSITION_DATA<??>

有効ポジションデータ 3 の取得と設定します.

オプション：

| | | |
|---|---|---|
| OUTPUT_OPERATION | レコードフォーマット拡張指定種別を指定します. | 0 - 1<br><br>0=無効<br>1=有効 |
| POSITION_NO | 先頭ポジション No. を指定します. | 0 - |
| RECORD_NUM | レコード数を指定します. | 1 - |
| AXIS_NUM | 有効軸数を指定します. | 1 - 8 |

get_value 時の戻り値のデータ構成：

| VT_ARRAY \| VT_VARIANT | | | |
|---|---|---|---|
| 0 | VT_ARRAY \| VT_VARIANT | | |
| | 0 | VT_UI2 | ポジション No. |
| | 1 | VT_UI2 | 軸パターン |
| | 2 | VT_UI2 | 加速度 |
| | 3 | VT_UI2 | 減速度 |
| | 4 | VT_UI2 | 速度 |
| | 5 | VT_ARRAY \| VT_I4 | |
| | | n VT_I4 | 位置データ |
| | 6 | VT_UI2 | 拡張データ |
| | 7 | VT_UI4 | 出力ファンクションコード |
| | 8 | VT_UI4 | 出力ポート・フラグ No. |
| | 9 | VT_UI4 | ファンクションパラメータ 1 |
| | 10 | VT_UI4 | ファンクションパラメータ 2 |

put_value 時の設定値のデータ構成：

get_value の戻り値のデータ構成と同じ

#### 2.4.1.19. COORDINATE_DATA_RANGE<??>

座標系定義データ範囲指定連続を取得します.

オプション :

| | | |
|---|---|---|
| TYPE | 種別を指定します. | 0 - 1<br>0=ワーク座標系定義データ<br>1=ツール座標系定義データ |
| DATA_NO | 照会先先頭座標系定義データ No. を指定します. | 0 - |
| RECORD_NUM | 照会レコード数を指定します. | 1 - |

get_value 時の戻り値のデータ構成 :

| | | | | |
|---|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | | |
| 0 | VT_ARRAY \| VT_I4 | | | |
| | 0 | VT_I4 | X 座標オフセット量 | |
| | 1 | VT_I4 | Y 座標オフセット量 | |
| | 2 | VT_I4 | Z 座標オフセット量 | |
| | 3 | VT_I4 | R 座標オフセット量 | |

### 2.4.1.20. UNIT_AXIS_STATUS<??>

ユニット軸ステータスを取得します.

オプション：

| | | |
|---|---|---|
| AXIS_PATTERN | 照会軸パターンを指定します. | 1 - 8 |
| TYPE | 種別を指定します. | 0 - |

get_value 時の戻り値のデータ構成：

| | | | |
|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | |
| 0 | VT_UI2 | | ワーク座標系選択 No. |
| 1 | VT_UI2 | | ツール座標系選択 No. |
| 2 | VT_UI2 | | 軸共通ステータス |
| 3 | VT_ARRAY \| VT_I4 | | |
| | 0 | VT_I4 | 軸ステータス |
| | 1 | VT_I4 | 軸センサー入力ステータス |
| | 2 | VT_I4 | 軸関連エラーコード |
| | 3 | VT_I4 | エンコーダステータス<br>（ リセット時） |
| | 4 | VT_I4 | 現在位置 |

#### 2.4.1.21. CHECK_ZONE_DATA_RANGE<??>

簡易干渉チェックゾーン定義データ範囲指定連続を取得します.

オプション:

| | | |
|---|---|---|
| DATA_NO | 照会先頭簡易干渉チェックゾーン定義データ No. を指定します. | 0 - |
| RECORD_NUM | 照会レコード数を指定します. | 1 - |
| AXIS_NUM | 軸数を指定します. | 1 - 4 |

get_value 時の戻り値のデータ構成:

| | | | | |
|---|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | | |
| 0 | VT_ARRAY \| VT_VARIANT | | | |
| | 0 | VT_UI4 | | 簡易干渉チェックゾーン<br>定義座標有効軸パターン |
| | 1 | VT_ARRAY \| VT_I4 | | |
| | | n | VT_I4 | 簡易干渉チェックゾーン<br>定義座標 1 |
| | 2 | VT_ARRAY \| VT_I4 | | |
| | | n | VT_I4 | 簡易干渉チェックゾーン<br>定義座標 2 |
| | 3 | VT_UI2 | | 侵入時出力<br>物理的出力<br>ポート No.<br>or<br>グローバル<br>フラグ No. |
| | 4 | VT_UI2 | | 侵入時エラー種別指定 |

### 2.4.1.22. EX2_UNIT_AXIS_STATUS<??>

ユニット軸ステータス 2 を取得します.

オプション :

| | | |
|---|---|---|
| AXIS_PATTERN | 照会軸パターンを指定します. | 1 - 8 |
| TYPE | 種別を指定します. | 0 - |

get_value 時の戻り値のデータ構成 :

| | | | | |
|---|---|---|---|---|
| VT_ARRAY \| VT_VARIANT | | | | |
| 0 | VT_UI2 | | | ワーク座標系選択 No. |
| 1 | VT_UI2 | | | ツール座標系選択 No. |
| 2 | VT_UI2 | | | 軸共通ステータス |
| 3 | VT_ARRAY \| VT_I4 | | | |
| | 0 | VT_I4 | | 軸ステータス |
| | 1 | VT_I4 | | 軸センサー入力ステータス |
| | 2 | VT_I4 | | 軸関連エラーコード |
| | 3 | VT_I4 | | エンコーダステータス<br>（ リセット時） |
| | 4 | VT_I4 | | 現在位置 |

#### 2.4.1.23. EX2_COORDINATE_DATA_RANGE<??>

座標系定義データ範囲指定連続 2 を取得します.

オプション：

| | | |
|---|---|---|
| AXIS_PATTERN | 照会軸パターンを指定します. | 1 - |
| TYPE | 種別を指定します. | 0 - 1<br>0=ワーク座標系定義データ<br>1=ツール座標系定義データ |
| DATA_NO | 照会先先頭座標系定義データ No. を指定します. | 0 - |
| RECORD_NUM | 照会レコード数を指定します. | 1 - |

get_value 時の戻り値のデータ構成：

| VT_ARRAY \| VT_VARIANT | | | |
|---|---|---|---|
| 0 | VT_BSTR | | ユニット軸パターン |
| 1 | VT_UI2 | | ワーク座標系選択 No.<br>(ユニット 1) |
| 2 | VT_UI2 | | ツール座標系選択 No.<br>(ユニット 1) |
| 3 | VT_UI2 | | ユニット 1<br>共通ステータス |
| 4 | VT_UI2 | | ワーク座標系選択 No.<br>(ユニット 2) |
| 5 | VT_UI2 | | ツール座標系選択 No.<br>(ユニット 2) |
| 6 | VT_UI2 | | ユニット 2<br>共通ステータス |
| 7 | VT_ARRAY \| VT_I4 | | |
| | 0 | VT_I4 | 軸ステータス |
| | 1 | VT_I4 | 軸センサー入力ステータス |
| | 2 | VT_I4 | 軸関連エラーコード |
| | 3 | VT_I4 | エンコーダステータス<br>( リセット時) |
| | 4 | VT_I4 | 現在位置 |

## 2.5. エラーコード

本プロバイダには，以下の独自エラーコードが存在します．（表 2-5 参照）
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 2-5 独自エラーコード表**

| エラー名 | エラー番号 | 説明 |
|---|---|---|
| E_AXIS_DUPLICATION | 0x80110001 | 軸番号指定に重複があります． |
| E_OUT_OF_DATA_RANGE | 0x80110002 | 指定されたデータに値範囲外のものが存在します． |
| E_PACKET_CHECKSUM | 0x80110003 | 通信によるパケット異常が発生しました． |
| E_MEMORY_AREA_ALLOCATION | 0x80110004 | 内部のメモリ領域確保が失敗しました． |
| ― | 0x80100*** | デバイスでエラーが発生しました．<br>***は 3 桁のエラーコードです．<br>対象機種のマニュアルのエラーコードを参照してください． |

# 3. サンプルプログラム

以下に SEL デバイスへ書き込みをする簡単なサンプルを示します.

前提条件：
・RC8 の PAC スクリプトを対象としたものとする.
・SEL デバイスの COM ポートは「1」とする.

**List 3-1 Sample.pcs**

```
Sub Main

        ' オブジェクト
        Dim caoCtrl as Object
        Dim caoVal as Object

        ' コントローラーオブジェクト作成
        caoCtrl = cao.AddController("SEL", "CaoProv.IAI.SEL", "",
"conn=com:1,station=99,timeout=500")

        ' サーボON
        caoCtrl.Execute "SERVO", Array("1", 1)

        ' 原点復帰
        caoCtrl.Execute "ORIGIN", Array("1", 100, 100)
        ' 原点復帰が完了するまで待ち
        delay 10000

        ' 絶対値移動
        caoCtrl.Execute "MOVE_ABSOLUITE", Array("1", 100, 100 ,100 , ARRAY(300000))
        ' 絶対値移動が完了するまで待ち
        delay 10000

End Sub
```