

Brother Protocol2 プロバイダ

Version 1.2.2

ユーザーズ ガイド

March 23, 2020

備考：

【改版履歴】

バージョン	日付	内容
1.0.0	2018-01-24	初版.
1.1.0	2018-02-28	ファイルデータ「工具データ」「マクロ変数」に関するメソッド、変数の追加
1.1.1	2018-04-23	対象バージョン明記 記述漏れ修正
1.2.0	2018-07-05	キープアライブ追加
1.2.1	2018-09-18	バージョン取得の不具合修正 受信データ解析の不具合修正
1.2.2	2020-03-23	複数接続時の不具合修正

【接続確認済み機器】

■ 【機器名】

機種名	ファームウェアバージョン	注意事項
SPEEDIO	WX01.08.060	

目次

1. はじめに.....	7
1.1. 本書が想定している環境とバージョン	8
1.2. 接続方法と注意点	8
1.3. 参考となる情報源	8
2. アプリケーション開発のための環境セットアップ	9
2.1. PC 環境のセットアップ	9
2.1.1. Brother Protocol2 プロバイダの自動インストール	9
2.1.2. Brother Protocol2 プロバイダの手動インストール	9
3. Brother Protocol2 プロバイダによるプログラミング	10
3.1. PLC にアクセスするサンプルプログラミング	10
3.1.1. サンプルプログラム	11
3.1.1.1. 接続	12
3.1.1.2. PLC アドレスのリード・ライト	13
3.1.1.3. 切断	14
4. コマンドリファレンス.....	15
4.1. コマンド一覧	15
4.2. メソッド・プロパティ	15
4.2.1. CaoWorkspace::AddController メソッド.....	15
4.2.2. CaoController::AddVariable メソッド.....	17
4.2.3. CaoController::GetVariableNames メソッド.....	18
4.2.4. CaoController::Execute メソッド.....	18
4.2.4.1. ReadFileManagementData.....	20
4.2.4.2. ReadSelectPRGInfo	21
4.2.4.3. ReadDate	21
4.2.4.4. ReadPLCData	22
4.2.4.5. WriteRelativePOSOffset	22
4.2.4.6. WriteRelativePOSOffsetAll	22
4.2.4.7. WriteToolOffset	23
4.2.4.8. WriteToolLife.....	24
4.2.4.9. WriteToolGroup	24
4.2.4.10. WriteDate	25

4.2.4.11. WriteOperatingtime	25
4.2.4.12. WritePLCData	26
4.2.4.13. ClearToolLife	26
4.2.4.14. ClearToolGroup	26
4.2.4.15. IOReference	27
4.2.4.16. IOModiy	27
4.2.4.17. RefreshPDSP	27
4.2.4.18. GetPDSP	27
4.2.4.19. GetMachinePOS	28
4.2.4.20. GetRelativePOS	28
4.2.4.21. GetAbsolutePOS	29
4.2.4.22. GetDistanceTarget	30
4.2.4.23. GetSpeed	30
4.2.4.24. GetToolNo	31
4.2.4.25. GetNextToolNo	31
4.2.4.26. GetMagagineNo	31
4.2.4.27. RefreshALARM	32
4.2.4.28. GetALARM	32
4.2.4.29. GetAlarmCount	32
4.2.4.30. GetCurrentAlarm	32
4.2.4.31. RefreshPANEL	33
4.2.4.32. GetPANEL	33
4.2.4.33. GetMode	33
4.2.4.34. RefreshWKCNTNTR	34
4.2.4.35. GetWKCNTNTR	34
4.2.4.36. GetNCWorkCounter	35
4.2.4.37. GetInteractiveWorkCounter	35
4.2.4.38. RefreshPRDC2	36
4.2.4.39. GetPRDC2	36
4.2.4.40. GetOpeTimeHistoryCount	36
4.2.4.41. GetOpeTimeHistory	37
4.2.4.42. RefreshHEAC	37
4.2.4.43. GetHEAC	38
4.2.4.44. SetHEACRange	38
4.2.4.45. GetHEACHistoryCount	39
4.2.4.46. GetHEACZHistory	39
4.2.4.47. GetHEACXHistory	40

4.2.4.48. GetHEACYHistory	41
4.2.4.49. RefreshTOLS	41
4.2.4.50. GetTOLS	42
4.2.4.51. GetToolData	42
4.2.4.52. RefreshMCRS	44
4.2.4.53. GetMCRS	44
4.2.4.54. GetMacroVariable.....	44
4.2.5. CaoVariable::get_Value プロパティ	45
4.2.6. CaoVariable::put_Value プロパティ	45
4.3. 変数一覧.....	45
4.3.1. コントローラクラス.....	45
4.3.1.1. @MAKER_NAME.....	47
4.3.1.2. @VERSION	47
4.3.1.3. PLC<??>	48
4.3.1.4. IO<??>	48
4.3.1.5. @FILE_DATA	49
4.3.1.6. @SELECT_PRG_INFO	49
4.3.1.7. @DATE	50
4.3.1.8. MACHINE_POS<??>	51
4.3.1.9. RELATIVE_POS<??>.....	52
4.3.1.10. ABSOLUTE_POS<??>.....	53
4.3.1.11. DISTANCE_TARGET<??>	54
4.3.1.12. SPEED<??>	55
4.3.1.13. TOOL_NO<??>	55
4.3.1.14. NEXT_TOOL_NO<??>.....	56
4.3.1.15. MAGAGINE_NO<??>	57
4.3.1.16. ALARM_COUNT<??>	57
4.3.1.17. ALARM<??>	58
4.3.1.18. MODE<??>	58
4.3.1.19. NC_WORK_COUNTER<??>	59
4.3.1.20. INTERACTIVE_WORK_COUNTER<??>.....	60
4.3.1.21. OPE_TIME_COUNT<??>	61
4.3.1.22. OPE_TIME_HISTORY<??>	62
4.3.1.23. HEAC_HISTORY_COUNT<??>.....	62
4.3.1.24. HEAC_Z_HISTORY<??>	63
4.3.1.25. HEAC_X_HISTORY<??>	64
4.3.1.26. HEAC_Y_HISTORY<??>	66

4.3.1.27. TOOL_DATA<??>	67
4.3.1.28. MACRO<??>.....	69
5. 付録	70
付録 A. エラーコード	70
付録 B. ファイルデータの読み込みの仕組み	70
付録 C. PLC 信号種類.....	72

1. はじめに

本書は、brother 社製のコンパクトマシニングセンタ (SPEEDIO 等) に対して NC 専用通信装置通信プロトコル方式 2 を使用してデータの読み込み、及び書き込みを行なう、Brother Protocol2 プロバイダのユーザーズガイドです。下図 1-1 が本プロバイダとデバイスの全体構成図になります。

以降 brother 製のコンパクトマシニングセンタをマシニングセンタと呼称します。

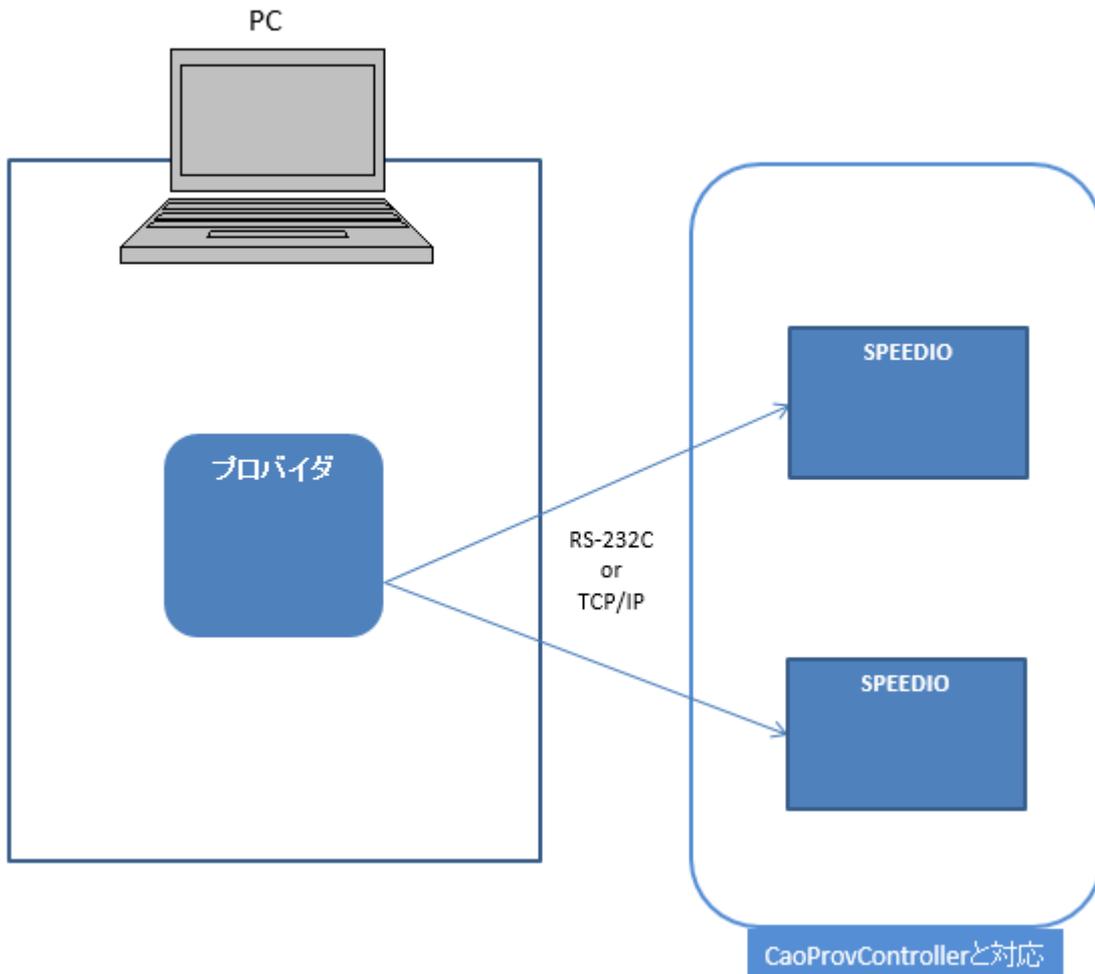


図 1-1 全体構成図

また、本プロバイダ及びデバイスそれぞれの対応を表した図が下図 1-2 となります。
(※一例です。全てを表しているわけではありません。)



図 1-2 Brother Protocol2 プロバイダと SPEEDIO のデータ対応図

1. 1. 本書が想定している環境とバージョン

クライアント PC が Windows 上で動作し、対象とするマシニングセンタのソフトウェアバージョン 5 以上である環境を想定しています。マシニングセンタのバージョンは「XXXX.YY.ZZZZ(例 WX01.08.060)」で表されますが、ここでのソフトウェアバージョンは「YY」の部分を指しています。PC の開発環境は、Component Object Model (COM, コンポーネント・オブジェクト・モデル) をサポートするプログラミング環境であれば開発が可能です。

1. 2. 接続方法と注意点

Brother Protocol2 プロバイダはシリアル通信 もしくは TCP/IP 通信でマシニングセンタと接続します。TCP/IP 通信で接続する場合、マシニングセンタの仕様上、前回通信から応答監視時間¹経過後に接続が切断されてしまうため、応答監視時間未満の周期で通信し続けるようにする必要があります。

1. 3. 参考となる情報源

本書のプログラミング事例は、すべて Visual Basic 6.0 で記載していますが、C++, Java, .NET, LabVIEW, Delphi などさまざまなプログラム言語で開発が可能です。使用方法に関しては、「ORiN 2 プログラミングガイド」を参照ください。

「ORiN 2 プログラミングガイド」は ORiN2 SDK インストールフォルダの以下のファイルに該当します。

- ・ ORiN2¥CAO¥Doc¥ORiN2_ProgrammersGuide_<lang>.pdf

※<lang>の部分は環境毎の言語文字列に置き換えてお読みください。

プロバイダを使ったアプリケーションを開発する上で必要となる ORiN2, COM/DCOM の基礎知識や技術に関して例を交えながら解説されています。

マシニングセンタの通信仕様に関しては、ブラザー工業株式会社より入手してください。

¹ マシニングセンタ内の設定項目

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 環境のセットアップ

2.1.1. Brother Protocol2 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、Brother マシニングセンタに接続するための動作環境（ラインタイム）の準備は完了です。

開発環境のセットアップは別途、Microsoft Visual Studio 6.0, 2003/2005/2008/2010, LabVIEW など Component Object Model (COM, コンポーネント・オブジェクト・モデル) をサポートする、プログラミング環境をご準備してください。

2.1.2. Brother Protocol2 プロバイダの手動インストール

インストーラを使用せずにセットアップを行う場合は手作業で下記レジストリ登録を行う必要があります。

表 2-1 Brother Protocol2 プロバイダ

ファイル名	CaoProvBrotherProtocol2.dll
ProgID	CaoProv.Brother.Protocol2
レジストリ登録	regsvr32 CaoProvBrotherProtocol2.dll
レジストリ登録の抹消	regsvr32 /u CaoProvBrotherProtocol2.dll

CAO エンジンが動作するには予め、PC 毎に正規の ORiN2 SDK ライセンスが1つ登録されていなくてはなりません。ORiN2 SDK ユーザーズガイド内にある「ライセンスの追加と削除」の節を参照してください。

3. Brother Protocol2 プロバイダによるプログラミング

Brother Protocol2 プロバイダでは、以下の手順で PC とマシニングセンタを接続することができます。

- CaoEngine の作成
- CaoWorkspace の作成
- CaoController の作成

マシニングセンタに接続した後は、CaoController の Execute メソッドを使用する、もしくは、CaoVariable オブジェクトを生成することで、マシニングセンタ内の情報にアクセスすることができます。

3.1. PLC にアクセスするサンプルプログラミング

ここでは例として PLC のグローバル OM のデータレジスタの値を読み書きするサンプルプログラムを示します。表 3-1 にサンプルプログラムの要件を、図 3-1 にサンプルプログラムの流れをそれぞれ記述しています。

表 3-1 サンプルプログラムの要件

要件	説明
接続先	TCP/IP で接続する
	接続先 IP アドレスは 192.168.1.2
	接続先ポート番号は 10000
処理内容	PLC の D0100 から値を読み込む。
	PLC の D1100 に取得した値 + 1 の値を書込む

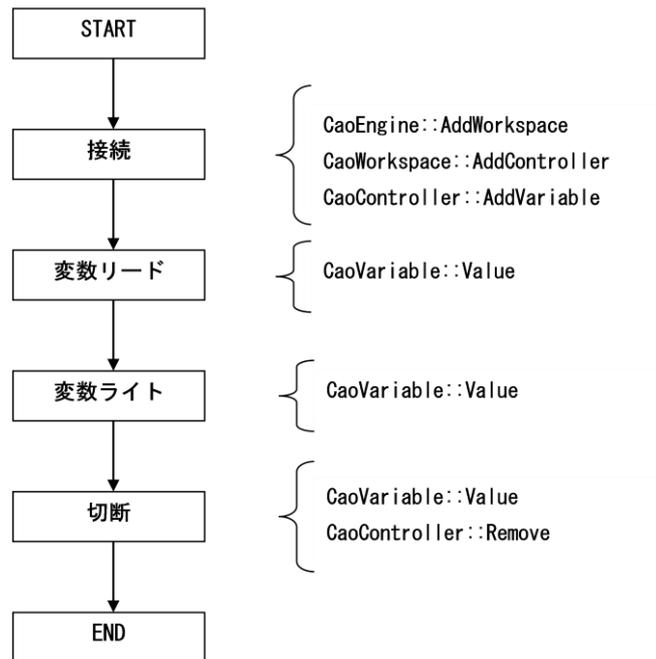


図 3-1 PLC アクセスの流れ

以降の節から具体的なコードを示します。

3.1.1. 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	READ_WRITE_PLC.vb
---------------	--------------------------

```

Sub Main
  ' オブジェクト
  Dim engine As caoEngine
  Dim workspace As caoWorkspace
  Dim ctrl As CaoController
  Dim varD0100 As CaoVariable
  Dim varD1100 As CaoVariable

  ' CaoEngine オブジェクトの生成
  Set engine = New caoEngine
  ' CaoWorkspace オブジェクトの生成
  Set workspace = engine.AddWorkspace("NewWrks", "")
  ' CaoController オブジェクトの生成
  Set ctrl = workspace.AddController("BrotherMC", _
    "CaoProv.Brother.Protocol2", _
    "", _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")

  ' D0100 にアクセスする CalVariable を CaoController に追加

```

```

Set varD0100 = ctrl.AddVariable("PLC_D0100", "KIND=D, NO=100")
' D1100 にアクセスする CaoVariable を CaoController に追加
Set varD1100 = ctrl.AddVariable("PLC_D1100", "KIND=D, NO=1100")

' D0100 から値を読み込む
Dim value As Integer
value = D0100.value

' D0100 から読込んだ値 + 1 を D1100 に書込む
varD1100.value = value + 1

' CaoController から CaoVariable をクリアする
Call ctrl.Variables.Clear
' CaoVariable の消去
Set varD0100 = Nothing
Set varD1100 = Nothing

' CaoWorkspace から CaoController を削除
Call workspace.Controllers.Remove(ctrl.Index)
' CaoController の消去
Set ctrl = Nothing

' CaoEngine から CaoWorkspace を削除
Call engine.Workspaces.Remove(workspace.Index)
' CaoWorkspace の消去
Set workspace = Nothing

' CaoEngine の消去
Set engine = Nothing
End Sub

```

3.1.1.1. 接続

マシニングセンタとの接続を行うには、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、CaoEngineオブジェクトとCaoWorkspaceオブジェクトとCaoControllerオブジェクトです。CaoWorkspaceオブジェクトは、CaoController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VB6でのコード例を示します。

```

Dim engine As caoEngine          ' CaoEngineオブジェクト用の変数
Dim workspace As caoWorkspace    ' CaoWorkspaceオブジェクト用の変数
Dim ctrl As CaoController        ' CaoControllerオブジェクト用の変数
Dim varD0100 As CaoVariable      ' CaoVariableオブジェクト用の変数
Dim varD1100 As CaoVariable      ' CaoVariableオブジェクト用の変数

```

- (2) CaoEngineオブジェクトを生成します。CaoEngine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

```
Set engine = New caoEngine
```

- (3) CaoWorkspaceオブジェクトを取得もしくは生成します。CaoEngine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

```
Set workspace = engine.AddWorkspace("NewWrks", "")
```

- (4) CaoControllerオブジェクトを生成します。CaoControllerオブジェクトを生成するには、使用するプロバイダ名と使用するためのパラメータを設定します。Brother Protocol2プロバイダでは、接続先のコントローラのIPアドレスをオプションで指定します。以下にコード例を示します。

CaoController オブジェクトの生成

```
Set ctrl = g_wrks.AddController("BrotherMC", _
                               "CaoProv.Brother.Protocol2", _
                               "", _
                               "CONN=ETH:192.168.1.2, TIMEOUT=3000:10000")
```

- (5) CaoVariableオブジェクトを生成します。接続したい変数のCaoVariableオブジェクトを生成します。以下にPLC(D0100)とPLC(D1100)にアクセスする変数オブジェクトを生成するコード例を示します。

```
Set varD0100 = ctrl.AddVariable("PLC_D0100", "KIND=D, NO=100")
Set varD1100 = ctrl.AddVariable("PLC_D1100", "KIND=D, NO=1100")
```

3.1.1.2. PLC アドレスのリード・ライト

接続した PLC アドレスの値を取得/設定するには、CaoVariable オブジェクトの Value プロパティを参照/設定します。PLC アドレスの値を取得/設定する場合は、32 ビット整数型変数を用意する必要があります。以下にコード例を示します。

```
Dim value As Integer
value = D0100.value           ' 値の取得
varD1100.value = value + 1   ' 値の設定
```

3.1.1.3. 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。以下にコード例を示します。

```
' CaoController から CaoVariable をクリアする
Call ctrl.Variables.Clear
' CaoVariable の消去
Set varD0100 = Nothing
Set varD1100 = Nothing
' CaoWorkspace から CaoController を削除
Call workspace.Controllers.Remove(ctrl.Index)
' CaoController の消去
Set ctrl = Nothing
' CaoEngine から CaoWorkspace を削除
Call engine.Workspaces.Remove(workspace.Index)
' CaoWorkspace の消去
Set workspace = Nothing
' CaoEngine の消去
Set engine = Nothing
```

4. コマンドリファレンス

4.1. コマンド一覧

表 4-1 コマンド一覧

カテゴリ	メソッド/プロパティ	機能	参照
CaoWorkspace	AddController	コントローラに接続	P. 15
CaoController	AddVariable	ユーザ・システム変数に接続	P. 17
	GetVariableNames	ユーザ・システム変数の一覧の取得	P. 18
	Execute	コントローラクラスに実装されているコマンドの実行	P. 18
CaoVariable	get_Value	値の取得	P. 45
	put_Value	値の設定	P. 45

4.2. メソッド・プロパティ

4.2.1. CaoWorkspace::AddController メソッド

Controller オブジェクトの生成時を行います。

各オプションには、デバイスに接続するための情報を指定します。

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. Brother. Protocol2", ' 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
)
```

使用例

```
Dim engine As caoEngine
Dim workspace As caoWorkspace
Dim ctrl As CaoController
Set engine = New caoEngine
Set workspace = engine.AddWorkspace("NewWrks", "")
```

' シリアル通信を使用してマシニングセンタと接続する場合

```
Set ctrl = workspace.AddController("Controller", _
    "CaoProv. Brother. Protocol2", _
```

```

"""
    , _
    "CONN=COM:2, Timeout=500")
' TCP/IP 通信を使用してマシンングセンタと接続する場合
Set ctrl = workspace.AddController("Controller", _
    "CaoProv. Brother. Protocol2", _
    """
    , _
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")

```

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 4-2 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>[:<ボーレート>[:<パリティ>:<データビット数>:<ストップビット数>]]	O ²	シリアル通信を使用してマシンングセンタと接続する場合指定してください。 PC 側で使用できるシリアルポート番号を指定します。	-	ボーレート: 9600 パリティ: NONE データビット数: 7bit ストップビット数: 1bit
CONN= ETH:<接続先 IP>[:<接続先ポート>]	O ²	TCP/IP 通信を使用してマシンングセンタと接続する場合指定してください。 接続先の IP アドレスおよびポート番号を指定します。	-	ポート:10000
Timeout=<送受信タイムアウト時間>	-	送受信タイムアウト時間 [ms] を設定します。	1 - 65535	500
KeepaliveTime=<キープアライブ待機時間>	₃	キープアライブ待機時間 [ms] を設定します。	1 - 4294967295	10000
KeepaliveInterval=<キープアライブ周期時間>	₃	キープアライブ周期時間 [ms] を設定します。	1 - 4294967295	2000

² Com, ETH のどちらか必須。

³ ETH の時のみ

4.2.2. CaoController::AddVariable メソッド

CaoController から CaoVariable オブジェクトを生成します。変数名には、表 4-4 に示す変数名のみ使用できます。

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  
)
```

使用例

```
Dim engine As caoEngine  
Dim workspace As caoWorkspace  
Dim ctrl As CaoController  
Set engine = New caoEngine  
Set workspace = engine.AddWorkspace("NewWrks", "")  
Set ctrl = workspace.AddController("Controller", _  
    "GaoProv.Brother.Protocol2", _  
    "", _  
    "CONN=ETH:192.168.1.2, TIMEOUT=3000")  
  
' PLC 信号 X3FF の変数を追加  
Dim varPLC As CaoVariable  
Set varPLC = ctrl.AddVariable("PLC_X_3FF", "KIND=X, NO=3FF")
```

4.2.3. CaoController::GetVariableNames メソッド

AddVariable メソッドで指定できる変数名とシステム変数名の一覧を取得します。

4.2.4. CaoController::Execute メソッド

Controller にてメソッドを実行します。

引数には、メソッドに対して必要な引数を指定します。

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
)
```

以下に、Execute で指定できるメソッド一覧を示します。使用例は各メソッドの詳細で記述しています。

表 4-3 CaoController::Execute メソッド一覧

メソッド名	説明	コマンド種別+機能
ReadFileManagementData	ファイル管理データを取得する。	REDFILE
ReadSelectPRGInfo	実行中プログラム情報を取得する。	REDPRGN
ReadDate	日付, 時刻を取得する。	REDDATE
ReadPLCData	PLC 信号データを取得する。	REDPLCD
WriteRelativePOSOffset	相対座標オフセット量のプリセットを行う。	WRTREL
WriteRelativePOSOffsetAll	相対座標オフセット量のプリセットを行う。(全軸)	WRTREL
WriteToolOffset	工具補正量を設定する。	WRTTOFS
WriteToolLife	工具の寿命を設定する。	WRTLLF
WriteToolGroup	工具グループに工具を設定する。	WRTTGRP
WriteDate	日付, 時刻を設定する。	WRDATE
WriteOperatingtime	稼働時間を設定する。	WRTOPTM

メソッド名	説明	コマンド種別+機能
WritePLCData	PLC 信号データを設定する.	WRTPLCD
ClearToolLife	工具の寿命を初期化する.	CLRTLLF
ClearToolGroup	工具グループの工具を削除する.	CLRTGRP
IOReference	外部入出力信号を参照する.	IOCREF
IOModiy	外部入出力信号を操作する.	IOCMOD
RefreshPDSP	「位置データ」ファイルデータを更新する.	LOD
GetPDSP	「位置データ」ファイルデータを取得する.	LOD
GetMachinePOS	機械座標位置を取得する.	LOD
GetRelativePOS	相対座標位置を取得する.	LOD
GetAbsolutePOS	絶対座標位置を取得する.	LOD
GetDistanceTarget	残移動を取得する.	LOD
GetSpeed	送り速度を取得する.	LOD
GetToolNo	主軸工具番号を取得する.	LOD
GetNextToolNo	次工具番号を取得する.	LOD
GetMagagineNo	マガジン番号を取得する.	LOD
RefreshALARM	「現在アラーム」ファイルデータを更新する.	LOD
GetALARM	「現在アラーム」ファイルデータを取得する.	LOD
GetAlarmCount	有効アラーム情報数を取得する.	LOD
GetCurrentAlarm	現在アラームを取得する.	LOD
RefreshPANEL	「操作パネルデータ」ファイルデータを更新する.	LOD
GetPANEL	「操作パネルデータ」ファイルデータを取得する.	LOD
GetMode	モードを取得する.	LOD
RefreshWKCNTNTR	「ワークカウンタ」ファイルデータを更新する.	LOD
GetWKCNTNTR	「ワークカウンタ」ファイルデータを取得する.	LOD

メソッド名	説明	コマンド種別+機能
GetNCWorkCounter	生産カウンタ (NC) を取得する.	LOD
GetInteractiveWorkCounter	生産カウンタ (対話) を取得する.	LOD
RefreshPRDC2	「生産データ 2(時間表示)」ファイルデータを更新する.	LOD
GetPRDC2	「生産データ 2(時間表示)」ファイルデータを取得する.	LOD
GetOpeTimeHistoryCount	有効稼働時間記録数を取得する.	LOD
GetOpeTimeHistory	稼働時間記録を取得する.	LOD
RefreshHEAC	「熱変位自動補正」ファイルデータを更新する.	LOD
GetHEAC	「熱変位自動補正」ファイルデータを取得する.	LOD
SetHEACRange	熱変位自動補正履歴のデータ範囲を設定する.	LOD
GetHEACHistoryCount	指定データ範囲内の有効熱変位自動補正履歴数を取得する.	LOD
GetHEACZHistory	熱変位自動補正履歴 (Z 軸) を取得する.	LOD
GetHEACXHistory	熱変位自動補正履歴 (X 軸) を取得する.	LOD
GetHEACYHistory	熱変位自動補正履歴 (Y 軸) を取得する.	LOD
RefreshTOLS	「工具データ」ファイルデータを更新する.	LOD
GetTOLS	「工具データ」ファイルデータを取得する.	LOD
GetToolData	工具データを取得する.	LOD
RefreshMCRS	「マクロ変数」ファイルデータを更新する.	LOD
GetMCRS	「マクロ変数」ファイルデータを取得する.	LOD
GetMacroVariable	マクロ変数 (#500~#999) を取得する.	LOD

4.2.4.1. ReadFileManagementData

ファイル管理データを取得します.

項目	型説明
----	-----

項目	型説明		
引数	なし		
戻り値	VT_ARRAY VT_VARIANT		
	0	VT_UI2	登録数が返却されます。
	1	VT_UI2	残り登録数が返却されます。
	2	VT_UI8	メモリ使用量が返却されます。(単位: Byte)
	3	VT_UI8	メモリ残容量が返却されます。(単位: Byte)

使用例

```
' ファイル管理データを取得
Dim vArray() As Variant
vArray = ctrl.Execute("ReadFileManagementData")
```

4.2.4.2. ReadSelectPRGInfo

実行中プログラム情報を取得します。

項目	型説明		
引数	なし		
戻り値	VT_ARRAY VT_BSTR		
	0	VT_BSTR	実行中プログラム番号が返却されます。
	1	VT_BSTR	メインプログラム番号が返却されます。
	2	VT_BSTR	実行中ブロック番号が返却されます。

使用例

```
' 実行中プログラム情報を取得
Dim strArray() As String
strArray = ctrl.Execute("ReadSelectPRGInfo")
```

4.2.4.3. ReadDate

日付, 時刻を取得します。

項目	型説明		
引数	なし		
戻り値	VT_ARRAY VT_UI2		
	0	VT_UI2	年(西暦)が返却されます。
	1	VT_UI2	月が返却されます。
	2	VT_UI2	日が返却されます。
	3	VT_UI2	時が返却されます。

項目	型説明		
	4	VT_UI2	分が返却されます。
	5	VT_UI2	秒が返却されます。

使用例

```
' 日付, 時刻を取得
Dim vArray As Variant
vArray = ctrl.Execute("ReadDate")
```

4.2.4.4. ReadPLCData

PLC 信号データを取得します。指定できる信号種類および、信号番号については、「付録 C. PLC 信号種類」を参照してください。

項目	型説明		
引数	VT_ARRAY VT_BSTR		
	0	VT_BSTR	信号種類を指定します。
	1	VT_BSTR	信号番号を指定します。
戻り値	VT_I4	設定値が返却されます。	

使用例

```
' 実行中プログラム情報を取得
Dim val As Integer
val = ctrl.Execute("ReadPLCData", Array("D", "100"))
```

4.2.4.5. WriteRelativePOSOffset

指定軸に対して相対座標位置のオフセット量をプリセット(再設定)します。

項目	型説明		
引数	VT_ARRAY VT_VARIANT		
	0	VT_I2	軸番号を指定します。
	1	VT_R8	指定軸番号のオフセット量を指定します。
戻り値	なし		

使用例

```
' 軸番号 1 のオフセット量に, 12.3 を設定
Call ctrl.Execute("WriteRelativePOSOffset", Array(1, 12.3))
```

4.2.4.6. WriteRelativePOSOffsetAll

全軸に対して相対座標位置のオフセット量をプリセット(再設定)します。

項目	型説明		
引数	VT_ARRAY VT_R8		
	0	VT_R8	軸番号 1 のオフセット量を指定します。
	1	VT_R8	軸番号 2 のオフセット量を指定します。
	2	VT_R8	軸番号 3 のオフセット量を指定します。
	3	VT_R8	軸番号 4 のオフセット量を指定します。
	4	VT_R8	軸番号 5 のオフセット量を指定します。
	5	VT_R8	軸番号 6 のオフセット量を指定します。
	6	VT_R8	軸番号 7 のオフセット量を指定します。
	7	VT_R8	軸番号 8 のオフセット量を指定します。
	8	VT_R8	軸番号 9 のオフセット量を指定します。
	9	VT_R8	軸番号 10 のオフセット量を指定します。
	10	VT_R8	軸番号 11 のオフセット量を指定します。
	11	VT_R8	軸番号 12 のオフセット量を指定します。
	12	VT_R8	軸番号 13 のオフセット量を指定します。
	13	VT_R8	軸番号 14 のオフセット量を指定します。
14	VT_R8	軸番号 15 のオフセット量を指定します。	
戻り値	なし		

使用例

軸番号 1~15 それぞれにオフセット量を設定

```
Call ctrl.Execute("WriteRelativePOSoffsetAll", Array(10, 2.2, 0, 12.2, 50, 0.5, 10.1, 0, 20.5, 1.12, 11.12, 0, 55, 0, 10))
```

4.2.4.7. WriteToolOffset

工具補正量を設定します。

項目	型説明		
引数	VT_ARRAY VT_VARIANT		
	0	VT_UI2	工具番号を指定します。(1 - 90)
	1	VT_UI2	補正種類を指定します。 0: 工具長オフセット 1: 工具長磨耗補正 2: 工具径補正 3: 工具径磨耗補正
2	VT_R8	補正量を指定します。	

項目	型説明
戻り値	なし

使用例

’ 工具番号 1 の工具長オフセットに 10 を設定

Call ctrl.Execute(“WriteToolOffset”, Array(1, 0, 10))

4.2.4.8. WriteToolLife

工具の寿命を設定します。

項目	型説明		
引数	VT_ARRAY VT_VARIANT		
	0	VT_UI2	工具番号を指定します。(1 - 90)
	1	VT_UI2	寿命種類を指定します。 0: 寿命単位 1: 初期寿命 2: 予告寿命 3: 寿命
	2	VT_UI4	設定値を指定します。 寿命種類に 0 を指定した場合は以下の寿命単位が設定可能 1: カウントしない 2: 時間(分) 3: 穴加工数(穴) 4: プログラム(回) 寿命種類に 1-3 を指定した場合は寿命値
戻り値	なし		

使用例

’ 工具番号 1 の寿命単位に時間(分)を設定

Call ctrl.Execute(“WriteToolLife”, Array(1, 0, 4))

4.2.4.9. WriteToolGroup

工具グループに工具を設定します。

項目	型説明		
引数	VT_ARRAY VT_UI2		
	0	VT_UI2	グループ番号を指定します。(1 - 30)
	1	VT_UI2	使用順を指定します。(1 - 30)

項目	型説明		
	2	VT_UI2	工具番号を指定します。(1 - 99)
戻り値	なし		

使用例

' 工具グループ 1 の使用順 10 に工具番号 2 を設定

Call ctrl.Execute("WriteToolGroup", Array(1, 10, 2))

4.2.4.10. WriteDate

日付, 時刻を設定します。

項目	型説明		
引数	VT_ARRAY VT_UI2		
	0	VT_UI2	年(西暦)を指定します。
	1	VT_UI2	月を指定します。
	2	VT_UI2	日を指定します。
	3	VT_UI2	時を指定します。
	4	VT_UI2	分を指定します。
5	VT_UI2	秒を指定します。	
戻り値	なし		

使用例

' NC の日付, 時刻に 2018 年 4 月 26 日 15 時 51 分 00 秒を設定

Call ctrl.Execute("WriteDate", Array(2018, 4, 26, 15, 51, 00))

4.2.4.11. WriteOperatingtime

稼働時間を設定します。

項目	型説明		
引数	VT_ARRAY VT_VARIANT		
	0	VT_UI4	時を指定します。
	1	VT_UI2	分を指定します。
2	VT_UI2	秒を指定します。	
戻り値	なし		

使用例

' 稼働時間に 112 時間 51 分 11 秒を設定

Call ctrl.Execute("WriteOperatingtime", Array(112, 51, 11))

4.2.4.12. WritePLCData

PLC 信号データを設定します。指定できる信号種類および、信号番号については、「付録 C. PLC 信号種類」を参照してください。

項目	型説明		
引数	VT_ARRAY VT_VARIANT		
	0	VT_BSTR	信号種類を指定します。
	1	VT_BSTR	信号番号を指定します。
	2	VT_I4	設定値を指定します。
戻り値	なし		

使用例

```
' 信号 D, 番号 1100 の信号データに 1 を設定
Call ctrl.Execute("WritePLCData", Array("D", "1100", 1))
```

4.2.4.13. ClearToolLife

工具の寿命を初期化します。

項目	型説明	
引数	VT_UI2	工具番号を指定します。(1 - 99)
戻り値	なし	

使用例

```
' 工具番号 99 の寿命を初期化
Call ctrl.Execute("ClearToolLife", 99)
```

4.2.4.14. ClearToolGroup

工具グループの工具を削除します。

項目	型説明		
引数	VT_ARRAY VT_UI2		
	0	VT_UI2	グループ番号を指定します。(1-30)
	1	VT_UI2	使用順を指定します。(1-30)
戻り値	なし		

使用例

```
' 工具グループ 1 の使用順 30 の工具を削除
Call ctrl.Execute("ClearToolGroup", Array(1, 30))
```

4.2.4.15. IOReference

外部入出力信号を参照します。指定できる信号名については

項目	型説明	
引数	VT_BSTR	信号名を指定します。
戻り値	VT_BOOL	信号値が返却されます。

使用例

```
' PR01 の値を取得
Dim val As Boolean
val = ctrl.Execute("IOReference", "PR01")
```

4.2.4.16. IOModiy

外部入出力信号を操作(設定)します。

項目	型説明		
引数	VT_ARRAY VT_VARIANT		
	0	VT_BSTR	信号名を指定します。
	1	VT_BOOL	設定値を指定します。
戻り値	なし		

使用例

```
' PR02 の値を ON に設定
Call ctrl.Execute("IOModify", Array("PR02", True))
```

4.2.4.17. RefreshPDSP

プロバイダが保持している「位置データ」ファイルデータを更新します。

項目	型説明
引数	なし
戻り値	なし

使用例

```
' プロバイダが保持している「位置データ」ファイルデータを更新
Call ctrl.Execute("RefreshPDSP")
```

4.2.4.18. GetPDSP

「位置データ」ファイルデータを取得します。

項目	型説明
----	-----

引数	VT_BOOL	データリフレッシュ実行有無を指定します。
戻り値	VT_BSTR	「位置データ」ファイルデータが返却されます。

使用例

```

' 「位置データ」ファイルデータ更新し、「位置データ」ファイルデータを取得
Dim strPDSP As String
strPDSP = ctrl.Execute("GetPDSP", True)

```

4.2.4.19. GetMachinePOS

「位置データ」ファイルデータ内の機械座標位置を取得します。

項目	型説明		
引数	VT_BOOL	データリフレッシュ実行有無を指定します。	
戻り値	VT_ARRAY VT_R8		
	0	VT_R8	機械座標(X軸)が返却されます。
	1	VT_R8	機械座標(Y軸)が返却されます。
	2	VT_R8	機械座標(Z軸)が返却されます。
	3	VT_R8	機械座標(4軸)が返却されます。
	4	VT_R8	機械座標(5軸)が返却されます。
	5	VT_R8	機械座標(6軸)が返却されます。
	6	VT_R8	機械座標(7軸)が返却されます。
	7	VT_R8	機械座標(8軸)が返却されます。
	8	VT_R8	機械座標(P1軸)が返却されます。
	9	VT_R8	機械座標(P2軸)が返却されます。
	10	VT_R8	機械座標(P3軸)が返却されます。
11	VT_R8	機械座標(P4軸)が返却されます。	

使用例

```

' 「位置データ」ファイルデータ更新後、機械座標位置を取得
Dim dblMachinePos() As Double
dblMachinePos = ctrl.Execute("GetMachinePOS", True)

```

4.2.4.20. GetRelativePOS

「位置データ」ファイルデータ内の相対座標位置を取得します。

項目	型説明	
引数	VT_BOOL	データリフレッシュ実行有無を指定します。
戻り値	VT_ARRAY VT_R8	

項目	型説明		
	0	VT_R8	相対座標 (X 軸) が返却されます.
	1	VT_R8	相対座標 (Y 軸) が返却されます.
	2	VT_R8	相対座標 (Z 軸) が返却されます.
	3	VT_R8	相対座標 (4 軸) が返却されます.
	4	VT_R8	相対座標 (5 軸) が返却されます.
	5	VT_R8	相対座標 (6 軸) が返却されます.
	6	VT_R8	相対座標 (7 軸) が返却されます.
	7	VT_R8	相対座標 (8 軸) が返却されます.
	8	VT_R8	相対座標 (P1 軸) が返却されます.
	9	VT_R8	相対座標 (P2 軸) が返却されます.
	10	VT_R8	相対座標 (P3 軸) が返却されます.
	11	VT_R8	相対座標 (P4 軸) が返却されます.

使用例

「位置データ」ファイルデータ更新後、相対座標位置を取得

```
Dim dblRelativePos() As Double
```

```
dblRelativePos = ctrl.Execute("GetRelativePos", True)
```

4.2.4.21. GetAbsolutePOS

「位置データ」ファイルデータ内の絶対座標位置を取得します。

項目	型説明		
引数	VT_BOOL	データリフレッシュ実行有無を指定します.	
戻り値	VT_ARRAY VT_R8		
	0	VT_R8	絶対座標 (X 軸) が返却されます.
	1	VT_R8	絶対座標 (Y 軸) が返却されます.
	2	VT_R8	絶対座標 (Z 軸) が返却されます.
	3	VT_R8	絶対座標 (4 軸) が返却されます.
	4	VT_R8	絶対座標 (5 軸) が返却されます.
	5	VT_R8	絶対座標 (6 軸) が返却されます.
	6	VT_R8	絶対座標 (7 軸) が返却されます.
	7	VT_R8	絶対座標 (8 軸) が返却されます.
	8	VT_R8	絶対座標 (P1 軸) が返却されます.
9	VT_R8	絶対座標 (P2 軸) が返却されます.	

項目	型説明		
	10	VT_R8	絶対座標 (P3 軸) が返却されます。
	11	VT_R8	絶対座標 (P4 軸) が返却されます。

使用例

```

' 「位置データ」ファイルデータ更新後、絶対座標位置を取得
Dim dblAbsolutePos() As Double
dblAbsolutePos = ctrl.Execute("GetAbsolutePos", True)

```

4.2.4.22. GetDistanceTarget

「位置データ」ファイルデータ内の残移動を取得します。

項目	型説明		
引数	VT_BOOL		データリフレッシュ実行有無を指定します。
戻り値	VT_ARRAY VT_R8		
	0	VT_R8	残移動 (X 軸) が返却されます。
	1	VT_R8	残移動 (Y 軸) が返却されます。
	2	VT_R8	残移動 (Z 軸) が返却されます。
	3	VT_R8	残移動 (4 軸) が返却されます。
	4	VT_R8	残移動 (5 軸) が返却されます。
	5	VT_R8	残移動 (6 軸) が返却されます。
	6	VT_R8	残移動 (7 軸) が返却されます。
	7	VT_R8	残移動 (8 軸) が返却されます。
	8	VT_R8	残移動 (P1 軸) が返却されます。
	9	VT_R8	残移動 (P2 軸) が返却されます。
	10	VT_R8	残移動 (P3 軸) が返却されます。
11	VT_R8	残移動 (P4 軸) が返却されます。	

使用例

```

' 「位置データ」ファイルデータ更新後、残移動位置を取得
Dim dblDistance() As Double
dblDistance = ctrl.Execute("GetDistanceTarget", True)

```

4.2.4.23. GetSpeed

「位置データ」ファイルデータ内の送り速度を取得します。

項目	型説明		
引数	VT_BOOL		データリフレッシュ実行有無を指定します。

項目	型説明	
戻り値	VT_R8	送り速度が返却されます。

使用例

```
' 「位置データ」ファイルデータ更新後、速度を取得
Dim dblSpeed As Double
dblSpeed = ctrl.Execute("GetSpeed", True)
```

4.2.4.24. GetToolNo

「位置データ」ファイルデータ内の主軸工具番号を取得します。

項目	型説明	
引数	VT_BOOL	データリフレッシュ実行有無を指定します。
戻り値	VT_UI1	主軸工具番号が返却されます。

使用例

```
' 「位置データ」ファイルデータ更新後、主軸工具番号を取得
Dim bytToolNo As Byte
bytToolNo = ctrl.Execute("GetToolNo", True)
```

4.2.4.25. GetNextToolNo

「位置データ」ファイルデータ内の次工具番号を取得します。

項目	型説明	
引数	VT_BOOL	データリフレッシュ実行有無を指定します。
戻り値	VT_UI1	次工具番号が返却されます。

使用例

```
' 「位置データ」ファイルデータ更新後、次主軸工具番号を取得
Dim bytNextToolNo As Byte
bytNextToolNo = ctrl.Execute("GetNextToolNo", True)
```

4.2.4.26. GetMagazineNo

「位置データ」ファイルデータ内のマガジン番号を取得します。

項目	型説明	
引数	VT_BOOL	データリフレッシュ実行有無を指定します。
戻り値	VT_R8	マガジン番号が返却されます。

使用例

```
' 「位置データ」ファイルデータ更新後、マガジン番号を取得
Dim bytMagazineNo As Byte
```

```
bytMagagineNo = ctrl.Execute("GetMagagineNo", True)
```

4.2.4.27. RefreshALARM

「現在アラーム」ファイルデータを更新します。

項目	型説明
引数	なし
戻り値	なし

使用例

```
' プロバイダが保持している「現在アラーム」ファイルデータを更新
Call ctrl.Execute("RefreshALARM")
```

4.2.4.28. GetALARM

「現在アラーム」ファイルデータを取得します。

項目	型説明	
引数	VT_BOOL	データリフレッシュ実行有無を指定します。
戻り値	VT_BSTR	「現在アラーム」ファイルデータが返却されます。

使用例

```
' 「アラーム」ファイルデータ更新後、アラームファイル情報を取得
Dim strALARM As String
strALARM = ctrl.Execute("GetALARM", True)
```

4.2.4.29. GetAlarmCount

「現在アラーム」ファイルデータ内の有効アラーム情報数を取得します。

項目	型説明	
引数	VT_BOOL	データリフレッシュ実行有無を指定します。
戻り値	VT_UI1	有効アラーム情報数が返却されます。

使用例

```
' 「現在アラーム」ファイルデータ更新後、有効アラーム情報数を取得
Dim bytAlarmCount As Byte
bytAlarmCount = ctrl.Execute("GetAlarmCount", True)
```

4.2.4.30. GetCurrentAlarm

「現在アラーム」ファイルデータ内の現在アラーム情報を取得します。

項目	型説明
----	-----

項目	型説明		
引数	VT_BOOL	データリフレッシュ実行有無を指定します。	
戻り値	VT_ARRAY VT_VARIANT		
	n ⁴	VT_ARRAY VT_UI2	
		VT_UI2	発生個所が返却されます。
		VT_UI2	番号が返却されます。
	VT_UI2	補助番号が返却されます。	

使用例

```
' 「現在アラーム」ファイルデータ更新後、現在アラーム情報を取得
Dim vntCurrentAlarm As Variant
vntCurrentAlarm = ctrl.Execute("GetCurrentAlarm", True)
```

4.2.4.31. RefreshPANEL

「操作パネルデータ」ファイルデータを更新します。

項目	型説明
引数	なし
戻り値	なし

使用例

```
' プロバイダが保持している「操作パネルデータ」ファイルデータを更新
Call ctrl.Execute("RefreshPANEL")
```

4.2.4.32. GetPANEL

「操作パネルデータ」ファイルデータを取得します。

項目	型説明
引数	VT_BOOL
戻り値	VT_BSTR

使用例

```
' 「操作パネル」ファイルデータ更新後、操作パネル情報を取得
Dim strPANEL As String
strPANEL = ctrl.Execute("GetPANEL", True)
```

4.2.4.33. GetMode

「操作パネルデータ」ファイルデータ内のモードを取得します。

⁴ n=35

項目	型説明	
引数	VT_BOOL	データリフレッシュ実行有無を指定します。
戻り値	VT_UI1	モードが返却されます。 0:手動 1:MDI 運転 2:メモリ運転 3:プログラム編集 4:MDI 手動 5:運転中編集

使用例

```
' 「操作パネル」ファイルデータ更新後、モードを取得
Dim bytMode As Byte
bytMode = ctrl.Execute("GetMode", True)
```

4.2.4.34. RefreshWKCNTNTR

「ワークカウンタ」ファイルデータを更新します。

項目	型説明	
引数	なし	
戻り値	なし	

使用例

```
' プロバイダが保持している「ワークカウンタ」ファイルデータを更新
Call ctrl.Execute("RefreshWKCNTNTR")
```

4.2.4.35. GetWKCNTNTR

「ワークカウンタ」ファイルデータを取得します。

項目	型説明	
引数	VT_BOOL	データリフレッシュ実行有無を指定します。
戻り値	VT_BSTR	「ワークカウンタ」ファイルデータが返却されます。

使用例

```
' 「ワークカウンタ」ファイルデータ更新後、「ワークカウンタ」ファイルデータを取得
Dim strWKCNTNTR As String
strWKCNTNTR = ctrl.Execute("GetWKCNTNTR", True)
```

4. 2. 4. 36. GetNCWorkCounter

「ワークカウンタ」ファイルデータ内の生産カウンタ (NC) を取得します。

項目	型説明		
引数	VT_BOOL	データリフレッシュ実行有無を指定します。	
戻り値	VT_ARRAY VT_I4		
	0	VT_I4	カウント数(カウンタ 1)が返却されます。
	1	VT_I4	現在値(カウンタ 1)が返却されます。
	2	VT_I4	終了値(カウンタ 1)が返却されます。
	3	VT_I4	終了予告値(カウンタ 1)が返却されます。
	:		
	12	VT_I4	カウント数(カウンタ 4)が返却されます。
	13	VT_I4	現在値(カウンタ 4)が返却されます。
	14	VT_I4	終了値(カウンタ 4)が返却されます。
	15	VT_I4	終了予告値(カウンタ 4)が返却されます。

使用例

```
' ワークカウンタ」ファイルデータ更新後、生産カウンタ (NC) を取得
Dim lCounter() As Long
lCounter = ctrl.Execute("GetNCWorkCounter", True)
```

4. 2. 4. 37. GetInteractiveWorkCounter

「ワークカウンタ」ファイルデータ内の生産カウンタ (対話) を取得します。

項目	型説明		
引数	VT_BOOL	データリフレッシュ実行有無を指定します。	
戻り値	VT_ARRAY VT_I4		
	0	VT_I4	カウント数(カウンタ 1)が返却されます。
	1	VT_I4	現在値(カウンタ 1)が返却されます。
	2	VT_I4	終了値(カウンタ 1)が返却されます。
	3	VT_I4	終了予告値(カウンタ 1)が返却されます。
	:		
	12	VT_I4	カウント数(カウンタ 4)が返却されます。
	13	VT_I4	現在値(カウンタ 4)が返却されます。
	14	VT_I4	終了値(カウンタ 4)が返却されます。

項目	型説明		
	15	VT_I4	終了予告値(カウンタ 4)が返却されます。

使用例

```
' ワークカウンタ」ファイルデータ更新後、生産カウンタ(対話)を取得
Dim lCounter() As Long
lCounter = ctrl.Execute("GetInteractiveWorkCounter", True)
```

4.2.4.38. RefreshPRDC2

「生産データ 2(時間表示)」ファイルデータを更新します。

項目	型説明
引数	なし
戻り値	なし

使用例

```
' プロバイダが保持している「生産データ 2(時間表示)」ファイルデータを更新
Call ctrl.Execute("RefreshPRDC2")
```

4.2.4.39. GetPRDC2

「生産データ 2(時間表示)」ファイルデータを取得します。

項目	型説明	
引数	VT_BOOL	データリフレッシュ実行有無を指定します。
戻り値	VT_BSTR	「生産データ 2(時間表示)」ファイルデータが返却されます。

使用例

```
' 「生産データ 2(時間表示)」ファイルデータ更新後、
' 「生産データ 2(時間表示)」ファイルデータを取得
Dim strPRDC2 As String
strPRDC2 = ctrl.Execute("GetPRDC2", True)
```

4.2.4.40. GetOpeTimeHistoryCount

「生産データ 2(時間表示)」ファイルデータ内の有効稼働時間記録数を取得します。

項目	型説明	
引数	VT_BOOL	データリフレッシュ実行有無を指定します。
戻り値	VT_UI1	有効稼働時間記録数が返却されます。

使用例

’ 「生産データ 2(時間表示)」ファイルデータ更新後、有効稼働時間記録数を取得

Dim bytCount As Byte

bytCount = ctrl.Execute("GetOpeTimeHistoryCount", True)

4.2.4.41. GetOpeTimeHistory

「生産データ 2(時間表示)」ファイルデータ内の稼働時間記録を取得します。

項目	型説明		
引数	VT_BOOL	データリフレッシュ実行有無を指定します。	
戻り値	VT_ARRAY VT_VARIANT		
	0	VT_BSTR	日付(稼働時間記録 1)が返却されます。
	1	VT_BSTR	稼働時間(稼働時間記録 1)が返却されます。
	2	VT_I4	運転回数(稼働時間記録 1)が返却されます。
	3	VT_I4	稼働率(稼働時間記録 1)が返却されます。
	⋮		
	76	VT_BSTR	日付(稼働時間記録 20)が返却されます。
	77	VT_BSTR	稼働時間(稼働時間記録 20)が返却されます。
	78	VT_I4	運転回数(稼働時間記録 20)が返却されます。
	79	VT_I4	稼働率(稼働時間記録 20)が返却されます。

使用例

’ 「生産データ 2(時間表示)」ファイルデータ更新後、稼働時間記録を取得

Dim vntHistory As Variant

vntHistory = ctrl.Execute("GetOpeTimeHistory", True)

4.2.4.42. RefreshHEAC

「熱変位自動補正」ファイルデータを更新します。ファイルデータは現在の単位系のものが自動的に対象となります。

項目	型説明
引数	なし
戻り値	なし

使用例

’ プロバイダが保持している「熱変位自動補正」ファイルデータを更新

Call ctrl.Execute("RefreshHEAC")

4.2.4.43. GetHEAC

「熱変位自動補正」ファイルデータを取得します。

項目	型説明	
引数	VT_BOOL	データリフレッシュ実行有無を指定します。
戻り値	VT_BSTR	「熱変位自動補正」ファイルデータが返却されます。

使用例

```
' 「生産データ 2(時間表示)」ファイルデータ更新後、「熱変位自動補正」ファイルデータを取得
Dim strHEAC As String
strHEAC = ctrl.Execute("GetHEAC", True)
```

4.2.4.44. SetHEACRange

「熱変位自動補正」ファイルデータ内の熱変位自動補正履歴のデータ範囲を設定します。「GetHEACHistoryCount」、「GetHEACZHistory」、「GetHEACXHistory」、「GetHEACYHistory」メソッドで対象となる熱変位自動補正履歴データは、このメソッドで設定した範囲のデータです。このメソッドでデータ範囲を指定しない場合、取得したファイルデータ内の全データが対象となります。

また、「熱変位自動補正」ファイルデータが未取得の場合のみ、ファイルデータを更新します。

項目	型説明		
引数	VT_ARRAY VT_I2		
	0	VT_I2	データ範囲開始位置を指定します。(1 - 1080)
	1	VT_I2	データ数を指定します。(1 - 1080)
戻り値	なし		

使用例

```
' 熱変位自動補正履歴のデータ範囲を 10 件目から 20 件に設定
```

```
Call ctrl.Execute("SetHEACRange", Array(1, 20))
```

```
' 熱変位自動補正履歴(Z軸)を 10 件目から 20 件取得
```

```
Dim vntHEACZHist As Variant
```

```
vntHEACZHist = ctrl.Execute("GetHEACZHistory", False)
```

```
' 熱変位自動補正履歴(X軸)を 10 件目から 20 件取得
```

```
Dim vntHEACXHist As Variant
```

```
vntHEACXHist = ctrl.Execute("GetHEACXHistory", False)
```

```
' 熱変位自動補正履歴(Y軸)を 10 件目から 20 件取得
```

```
Dim vntHEACYHist As Variant
```

```
vntHEACYHist = ctrl.Execute("GetHEACYHistory", False)
```

4.2.4.45. GetHEACHistoryCount

「熱変位自動補正」ファイルデータの熱変位自動補正履歴の中で、「SetHEACRange」メソッドで指定した範囲のデータの有効熱変位自動補正履歴数を取得します。

項目	型説明	
引数	VT_BOOL	データリフレッシュ実行有無を指定します。
戻り値	VT_UI2	有効熱変位自動補正履歴数が返却されます。

使用例

「熱変位自動補正」ファイルデータ更新後、指定データ範囲内の有効熱変位自動補正履歴数を取得

```
Dim vValue As Variant
```

```
vValue = caoController.Execute("GetHEACHistoryCount", true)
```

4.2.4.46. GetHEACZHistory

「熱変位自動補正」ファイルデータ内の熱変位自動補正履歴 (Z 軸) を取得します。取得するデータの範囲は「SetHEACRange」メソッドで指定します。

項目	型説明		
引数	VT_BOOL	データリフレッシュ実行有無を指定します。	
戻り値	VT_ARRAY VT_VARIANT		
	0	VT_BSTR	日付 (履歴 m^5) が返却されます。
	1	VT_BSTR	時刻 (履歴 m) が返却されます。
	2	VT_I4	計算回数 (履歴 m) が返却されます。
	3	VT_R8	平均速度 (履歴 m) が返却されます。
	4	VT_R4	ボールねじ変位量 1 (履歴 m) が返却されます。
	5	VT_R4	ボールねじ変位量 2 (履歴 m) が返却されます。
	6	VT_UI4	主軸回転量累積値 (履歴 m) が返却されます。
	7	VT_R4	主軸変位量 (履歴 m) が返却されます。
	⋮		
	$n \times 8 - 8$	VT_BSTR	日付 (履歴 $m+n^6-1$) が返却されます。
	$n \times 8 - 7$	VT_BSTR	時刻 (履歴 $m+n-1$) が返却されます。
$n \times 8 - 6$	VT_I4	計算回数 (履歴 $m+n-1$) が返却されます。	

⁵ m = 「SetHEACRange」メソッドで指定したデータ範囲開始位置

⁶ n = 「SetHEACRange」メソッドで指定したデータ数

項目	型説明		
	n×8-5	VT_R8	平均速度(履歴 m+n-1)が返却されます。
	n×8-4	VT_R4	ボールねじ変位量 1(履歴 m+n-1)が返却されます。
	n×8-3	VT_R4	ボールねじ変位量 2(履歴 m+n-1)が返却されます。
	n×8-2	VT_UI4	主軸回転量累積値(履歴 m+n-1)が返却されます。
	n×8-1	VT_R4	主軸変位量(履歴 m+n-1)が返却されます。

使用例

SetHEACRange を参照

4.2.4.47. GetHEACXHistory

「熱変位自動補正」ファイルデータ内の熱変位自動補正履歴(X軸)を取得します。取得するデータの範囲は「SetHEACRange」メソッドで指定します。

項目	型説明			
引数	VT_BOOL		データリフレッシュ実行有無を指定します。	
戻り値	VT_ARRAY VT_VARIANT			
	0	VT_BSTR	日付(履歴 m ⁷)が返却されます。	
	1	VT_BSTR	時刻(履歴 m)が返却されます。	
	2	VT_I4	計算回数(履歴 m)が返却されます。	
	3	VT_R8	平均速度(履歴 m)が返却されます。	
	4	VT_R4	ボールねじ変位量 1(履歴 m)が返却されます。	
	5	VT_R4	ボールねじ変位量 2(履歴 m)が返却されます。	
	6	VT_UI4	主軸回転量累積値(履歴 m)が返却されます。	
	7	VT_R4	主軸変位量(履歴 m)が返却されます。	
		⋮		
	n×8-8	VT_BSTR	日付(履歴 m+n ⁸ -1)が返却されます。	
	n×8-7	VT_BSTR	時刻(履歴 m+n-1)が返却されます。	
	n×8-6	VT_I4	計算回数(履歴 m+n-1)が返却されます。	
	n×8-5	VT_R8	平均速度(履歴 m+n-1)が返却されます。	
	n×8-4	VT_R4	ボールねじ変位量 1(履歴 m+n-1)が返却されます。	
n×8-3	VT_R4	ボールねじ変位量 2(履歴 m+n-1)が返却されます。		
n×8-2	VT_UI4	主軸回転量累積値(履歴 m+n-1)が返却されます。		
n×8-1	VT_R4	主軸変位量(履歴 m+n-1)が返却されます。		

⁷ m = 「SetHEACRange」メソッドで指定したデータ範囲開始位置

⁸ n = 「SetHEACRange」メソッドで指定したデータ数

使用例

SetHEACRange を参照

4.2.4.48. GetHEACYHistory

「熱変位自動補正」ファイルデータ内の熱変位自動補正履歴 (Y 軸) を取得します。取得するデータの範囲は「SetHEACRange」メソッドで指定します。

項目	型説明		
引数	VT_BOOL		データリフレッシュ実行有無を指定します。
戻り値	VT_ARRAY VT_VARIANT		
	0	VT_BSTR	日付 (履歴 m^9) が返却されます。
	1	VT_BSTR	時刻 (履歴 m) が返却されます。
	2	VT_I4	計算回数 (履歴 m) が返却されます。
	3	VT_R8	平均速度 (履歴 m) が返却されます。
	4	VT_R4	ボールねじ変位量 1 (履歴 m) が返却されます。
	5	VT_R4	ボールねじ変位量 2 (履歴 m) が返却されます。
	6	VT_UI4	主軸回転量累積値 (履歴 m) が返却されます。
	7	VT_R4	主軸変位量 (履歴 m) が返却されます。
	⋮		
	$n \times 8 - 8$	VT_BSTR	日付 (履歴 $m+n^{10}-1$) が返却されます。
	$n \times 8 - 7$	VT_BSTR	時刻 (履歴 $m+n-1$) が返却されます。
	$n \times 8 - 6$	VT_I4	計算回数 (履歴 $m+n-1$) が返却されます。
	$n \times 8 - 5$	VT_R8	平均速度 (履歴 $m+n-1$) が返却されます。
	$n \times 8 - 4$	VT_R4	ボールねじ変位量 1 (履歴 $m+n-1$) が返却されます。
$n \times 8 - 3$	VT_R4	ボールねじ変位量 2 (履歴 $m+n-1$) が返却されます。	
$n \times 8 - 2$	VT_UI4	主軸回転量累積値 (履歴 $m+n-1$) が返却されます。	
$n \times 8 - 1$	VT_R4	主軸変位量 (履歴 $m+n-1$) が返却されます。	

使用例

SetHEACRange を参照

4.2.4.49. RefreshTOLS

「工具データ」ファイルデータを更新します。

⁹ m = 「SetHEACRange」メソッドで指定したデータ範囲開始位置

¹⁰ n = 「SetHEACRange」メソッドで指定したデータ数

項目	型説明		
引数	VT_ARRAY VT_UI1		
	0	VT_UI1	単位系を指定します。(0:現在の単位系, 1:メトリック, 2:インチ)
	1	VT_UI1	データ番号を指定します。(0:カレント面, 1~10:データ番号)
戻り値	なし		

使用例

```
' プロバイダが保持している「工具データ」ファイルデータを更新
```

```
Call ctrl.Execute("RefreshTOLS", Array(1, 1))
```

4.2.4.50. GetTOLS

「工具データ」ファイルデータを取得します。

項目	型説明		
引数	VT_ARRAY VT_VARIANT		
	0	VT_BOOL	データリフレッシュ実行有無を指定します。
	1	VT_UI1	単位系を指定します。(0:現在の単位系, 1:メトリック, 2:インチ)
	2	VT_UI1	データ番号を指定します。(0:カレント面, 1~10:データ番号)
戻り値	VT_BSTR	「工具データ」ファイルデータが返却されます。	

使用例

```
' 「工具データ」ファイルデータをメトリック,
```

```
' データ番号1のデータで更新後、「工具データ」ファイルデータを取得
```

```
Dim strTOLS As String
```

```
strTOLS = ctrl.Execute("GetTOLS", Array(True, 1, 1))
```

4.2.4.51. GetToolData

「工具データ」ファイルデータ内の工具データを取得します。

項目	型説明		
引数	VT_ARRAY VT_VARIANT		
	0	VT_BOOL	データリフレッシュ実行有無を指定します。
	1	VT_UI1	単位系を指定します。(0:現在の単位系, 1:メトリック, 2:インチ)
	2	VT_UI1	データ番号を指定します。(0:カレント面, 1~10:データ番号)
戻り値	VT_ARRAY VT_VARIANT		
	0	VT_BSTR	工具名(工具番号1)が返却されます。
	1	VT_R8	工具長オフセット(工具番号1)が返却されます。
	2	VT_R8	工具長摩耗補正(工具番号1)が返却されます。

項目	型説明		
	3	VT_R8	工具径オフセット(工具番号 1)が返却されます。
	4	VT_R8	工具径摩耗補正(工具番号 1)が返却されます。
	5	VT_I4	初期寿命(工具番号 1)が返却されます。
	6	VT_I4	予告寿命(工具番号 1)が返却されます。
	7	VT_I4	寿命(工具番号 1)が返却されます。
	8	VT_R8	周速(工具番号 1)が返却されます。
	9	VT_R4	回転送り(工具番号 1)が返却されます。
	10	VT_R4	S 指定値(工具番号 1)が返却されます。
	11	VT_R8	F 指定値(工具番号 1)が返却されます。
	12	VT_I4	最高回転数(工具番号 1)が返却されます。
	13	VT_BOOL	ツール洗浄(工具番号 1)が返却されます。
	14	VT_BOOL	CTS(工具番号 1)が返却されます。
	:		
	1470	VT_BSTR	工具名(工具番号 99)が返却されます。
	1471	VT_R8	工具長オフセット(工具番号 99)が返却されます。
	1472	VT_R8	工具長摩耗補正(工具番号 99)が返却されます。
	1473	VT_R8	工具径オフセット(工具番号 99)が返却されます。
	1474	VT_R8	工具径摩耗補正(工具番号 99)が返却されます。
	1475	VT_I4	初期寿命(工具番号 99)が返却されます。
	1476	VT_I4	予告寿命(工具番号 99)が返却されます。
	1477	VT_I4	寿命(工具番号 99)が返却されます。
	1478	VT_R8	周速(工具番号 99)が返却されます。
	1479	VT_R4	回転送り(工具番号 99)が返却されます。
	1480	VT_R4	S 指定値(工具番号 99)が返却されます。
	1481	VT_R8	F 指定値(工具番号 99)が返却されます。
	1482	VT_I4	最高回転数(工具番号 99)が返却されます。
	1483	VT_BOOL	ツール洗浄(工具番号 99)が返却されます。
	1484	VT_BOOL	CTS(工具番号 99)が返却されます。

使用例

```
Dim engine As caoEngine
Dim workspace As caoWorkspace
' 「工具データ」ファイルデータをメトリック、
' データ番号 1 のデータで更新後、工具データを取得
Dim vntToolData As Variant
```

```
vntToolData = ctrl.Execute("GetToolData", Array(True, 1, 1))
```

4.2.4.52. RefreshMCRS

「マクロ変数」ファイルデータを更新します。

項目	型説明		
引数	VT_ARRAY VT_UI1		
	0	VT_UI1	単位系を指定します。(0:現在の単位系, 1:メトリック, 2:インチ)
	1	VT_UI1	データ番号を指定します。(0:カレント面, 1~10:データ番号)
戻り値	なし		

使用例

```
' プロバイダが保持している「マクロ変数」ファイルデータを、  
' メトリック、データ番号1のデータで更新
```

```
Call ctrl.Execute("RefreshMCRS ", Array(1, 1))
```

4.2.4.53. GetMCRS

「マクロ変数」ファイルデータを取得します。

項目	型説明		
引数	VT_ARRAY VT_VARIANT		
	0	VT_BOOL	データリフレッシュ実行有無を指定します。
	1	VT_UI1	単位系を指定します。(0:現在の単位系, 1:メトリック, 2:インチ)
	2	VT_UI1	データ番号を指定します。(0:カレント面, 1~10:データ番号)
戻り値	VT_BSTR	「マクロ変数」ファイルデータが返却されます。	

使用例

```
' 「マクロ変数」ファイルデータをメトリック、  
' データ番号1のデータで更新後、「マクロ変数」ファイルデータを取得
```

```
Dim strMCRS As String  
strMCRS = ctrl.Execute("GetMCRS", Array(True, 1, 1))
```

4.2.4.54. GetMacroVariable

「マクロ変数」ファイルデータ内のマクロ変数(#500~#999)を取得します。

項目	型説明		
引数	VT_ARRAY VT_VARIANT		
	0	VT_BOOL	データリフレッシュ実行有無を指定します。
	1	VT_UI1	単位系を指定します。(0:現在の単位系, 1:メトリック, 2:インチ)
	2	VT_UI1	データ番号を指定します。(0:カレント面, 1~10:データ番号)

項目	型説明		
戻り値	VT_ARRAY VT_R8		
	0	VT_R8	マクロ変数(#500)が返却されます。
	1	VT_R8	マクロ変数(#501)が返却されます。
	⋮		
	498	VT_R8	マクロ変数(#998)が返却されます。
	499	VT_R8	マクロ変数(#999)が返却されます。

使用例

```

' 「マクロ変数」ファイルデータを、
' メトリック、データ番号1のデータで更新後、マクロ変数(#500~#999)を取得
Dim vntMacroVariable As Variant
vntMacroVariable = ctrl.Execute("GetMacroVariable ", Array(True, 1, 1))

```

4.2.5. CaoVariable::get_Value プロパティ

オブジェクトに対応している変数の値を取得します。

変数の実装状況およびデータ型は「表 4-4 コントローラクラス 変数一覧」を参照してください。

4.2.6. CaoVariable::put_Value プロパティ

オブジェクトに対応している変数に値を設定します。

変数の実装状況およびデータ型は「表 4-4 コントローラクラス 変数一覧」を参照してください。

4.3. 変数一覧

4.3.1. コントローラクラス

CaoController クラスで使用できる変数一覧を以下に示します。

表 4-4 コントローラクラス 変数一覧

変数名	データ型	説明	Value	
			get	put
@MAKER_NAME	VT_BSTR	メーカー名の取得	○	-
@VERSION	VT_BSTR	DLL のバージョンの取得	○	-
PLC<??>	VT_I4	PLC 信号データの取得と設定	○	○
IO<??>	VT_BOOL	外部入出力信号の取得と設定	○	○

変数名	データ型	説明	Value	
			get	put
@FILE_DATA	VT_ARRAY VT_VARIANT	ファイル管理データの取得	○	-
@SELECT_PRG_INFO	VT_ARRAY VT_BSTR	実行中プログラム情報の取得	○	-
@DATE	VT_ARRAY VT_UI2	日付, 時刻の取得と設定	○	○
MACHINE_POS<??>	VT_ARRAY VT_R8	機械座標位置の取得	○	-
RELATIVE_POS<??>	VT_ARRAY VT_R8	相対座標位置の取得	○	-
ABSOLUTE_POS<??>	VT_ARRAY VT_R8	絶対座標位置の取得	○	-
DISTANCE_TARGET<??>	VT_ARRAY VT_R8	残移動の取得	○	-
SPEED<??>	VT_R8	送り速度の取得	○	-
TOOL_NO<??>	VT_UI1	主軸工具番号の取得	○	-
NEXT_TOOL_NO<??>	VT_UI1	次工具番号の取得	○	-
MAGAGINE_NO<??>	VT_UI1	マガジン番号の取得	○	-
ALARM_COUNT<??>	VT_UI1	有効アラーム数の取得	○	-
ALARM<??>	VT_ARRAY VT_VARIANT	現在アラーム情報の取得	○	-
MODE<??>	VT_UI1	モードの取得	○	-
NC_WORK_COUNTER<??>	VT_ARRAY VT_I4	生産カウンタ (NC) の取得	○	-
INTERACTIVE_WORK_COUNTER<??>	VT_ARRAY VT_I4	生産カウンタ (対話) の取得	○	-
OPE_TIME_COUNT<??>	VT_UI1	有効稼働時間記録数の取得	○	-
OPE_TIME_HISTORY<??>	VT_ARRAY VT_VARIANT	稼働時間記録の取得	○	-
HEAC_HISTORY_COUNT<??>	VT_I2	有効熱変位自動補正履歴数の取得	○	-
HEAC_Z_HISTORY<??>	VT_ARRAY VT_VARIANT	熱変位自動補正履歴 (Z 軸) の取得	○	-

変数名	データ型	説明	Value	
			get	put
HEAC_X_HISTORY<??>	VT_ARRAY VT_VARIANT	熱変位自動補正履歴(X 軸)の取得	○	-
HEAC_Y_HISTORY<??>	VT_ARRAY VT_VARIANT	熱変位自動補正履歴(Y 軸)の取得	○	-
TOOL_DATA<??>	VT_ARRAY VT_VARIANT	工具データの取得	○	-
MACRO<??>	VT_ARRAY VT_R8	マクロ変数 (#500~#999) の取得	○	-

4.3.1.1. @MAKER_NAME

メーカー名の取得をします。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("@MAKER_NAME")
' 値取得
Dim strVal As String
strVal = var.value
```

● データ型

項目	型説明	
戻り値	VT_BSTR	メーカー名を取得します。

4.3.1.2. @VERSION

DLL のバージョンの取得をします。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("@VERSION")
' 値取得
Dim value As String
value = var.value
```

● データ型

項目	型説明	
戻り値	VT_BSTR	DLL のバージョンを取得します。

4.3.1.3. PLC<??>

PLC 信号データの取得, 設定をします.

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("@VERSIONE")
' 値取得
Dim value As String
value = var.value
```

● オプション

オプション	必須	説明	値範囲	デフォルト値
KIND= <PLC 信号種類を指定>	○	PLC 信号種類を指定します.	-	-
NO= <PLC 信号番号を指定>	○	PLC 信号番号を指定します.	-	-

● データ型

型説明	
VT_I4	PLC 信号データを取得します.

● 対応コマンド

コマンド種別+機能	get	REDPLCD
	put	WRTPLCD

4.3.1.4. IO<??>

外部入出力信号の取得, 設定をします.

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("IO_PRO2", "SIGNAL=PRO2")
' 値取得
Dim value As Boolean
value = var.value
' 値設定
If (value = True) Then
    var.value = False
Else
    var.value = True
End If
```

● オプション

オプション	必須	説明	値範囲	デフォルト値
SIGNAL= <外部入出力信号名を指定>	○	外部入出力信号名を指定します。	-	-

● データ型

型説明	
VT_BOOL	外部入出力信号を取得します。

● 対応コマンド

コマンド種別+機能	get	IOCREF
	put	IOCMOD

4.3.1.5. @FILE_DATA

ファイル管理データの取得をします。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("@FILE_DATA")
' 値取得
Dim value As Variant
value = var.value
```

● データ型

型説明			
VT_ARRAY VT_VARIANT			
0	VT_UI2	登録数が返却されます。	
1	VT_UI2	残り登録数が返却されます。	
2	VT_UI8	メモリ使用量が返却されます。(単位: Byte)	
3	VT_UI8	メモリ残容量が返却されます。(単位: Byte)	

● 対応コマンド

コマンド種別+機能	REDFILE
-----------	---------

4.3.1.6. @SELECT_PRG_INFO

実行中プログラム情報の取得をします。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("@SELECT_PRG_INFO")
' 値取得
Dim value As Variant
value = var.value
```

● データ型

型説明			
VT_ARRAY VT_BSTR			
0	VT_BSTR	実行中プログラム番号が返却されます。	
1	VT_BSTR	メインプログラム番号が返却されます。	
2	VT_BSTR	実行中ブロック番号が返却されます。	

● 対応コマンド

コマンド種別+機能	REDPRGN
-----------	---------

4.3.1.7. @DATE

日付, 時刻の取得, 設定をします。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("@DATE")
' 値取得
Dim value As Variant
value = var.value
' 値設定
var.Value = Array(2018, 04, 27, 11, 55, 00)
```

● データ型

型説明			
VT_ARRAY VT_UI2			
0	VT_UI2	年(西暦)を指定します。	
1	VT_UI2	月を指定します。	
2	VT_UI2	日を指定します。	
3	VT_UI2	時を指定します。	

型説明			
	4	VT_UI2	分を指定します.
	5	VT_UI2	秒を指定します.
なし			

- 対応コマンド

コマンド種別+機能	REDDATE
	WRDATE

4.3.1.8. MACHINE_POS<??>

「位置データ」ファイルデータ内の機械座標位置を取得します.

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("MACHINE_POS", "REFRESH=True")
' 値取得
Dim values() As Double
values = var.value
```

- オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= <ファイルデータ更新有無 を指定>	○	ファイルデータ更新処理の有無 を指定します.	True/False	-

- データ型

型説明			
VT_ARRAY VT_R8			
	0	VT_R8	機械座標 (X 軸) が返却されます.
	1	VT_R8	機械座標 (Y 軸) が返却されます.
	2	VT_R8	機械座標 (Z 軸) が返却されます.
	3	VT_R8	機械座標 (4 軸) が返却されます.
	4	VT_R8	機械座標 (5 軸) が返却されます.
	5	VT_R8	機械座標 (6 軸) が返却されます.
	6	VT_R8	機械座標 (7 軸) が返却されます.
	7	VT_R8	機械座標 (8 軸) が返却されます.

型説明			
8	VT_R8	機械座標 (P1 軸) が返却されます。	
9	VT_R8	機械座標 (P2 軸) が返却されます。	
10	VT_R8	機械座標 (P3 軸) が返却されます。	
11	VT_R8	機械座標 (P4 軸) が返却されます。	

- 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.9. RELATIVE_POS<??>

「位置データ」ファイルデータ内の相対座標位置を取得します。

使用例

```
' 変数追加
```

```
Dim var As CaoVariable
```

```
Set var = ctrl.AddVariable("RELATIVE_POS", "REFRESH=True")
```

```
' 値取得
```

```
Dim values() As Double
```

```
values = var.value
```

- オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

- データ型

型説明			
VT_ARRAY VT_R8			
0	VT_R8	相対座標 (X 軸) が返却されます。	
1	VT_R8	相対座標 (Y 軸) が返却されます。	
2	VT_R8	相対座標 (Z 軸) が返却されます。	
3	VT_R8	相対座標 (4 軸) が返却されます。	
4	VT_R8	相対座標 (5 軸) が返却されます。	
5	VT_R8	相対座標 (6 軸) が返却されます。	
6	VT_R8	相対座標 (7 軸) が返却されます。	
7	VT_R8	相対座標 (8 軸) が返却されます。	

型説明			
8	VT_R8	相対座標 (P1 軸) が返却されます。	
9	VT_R8	相対座標 (P2 軸) が返却されます。	
10	VT_R8	相対座標 (P3 軸) が返却されます。	
11	VT_R8	相対座標 (P4 軸) が返却されます。	

- 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.10. ABSOLUTE_POS<??>

「位置データ」ファイルデータ内の絶対座標位置を取得します。

使用例

' 変数追加

```
Dim var As CaoVariable
```

```
Set var = ctrl.AddVariable("ABSOLUTE_POS", "REFRESH=True")
```

' 値取得

```
Dim values() As Double
```

```
values = var.value
```

- オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

- データ型

型説明			
VT_ARRAY VT_R8			
0	VT_R8	絶対座標 (X 軸) が返却されます。	
1	VT_R8	絶対座標 (Y 軸) が返却されます。	
2	VT_R8	絶対座標 (Z 軸) が返却されます。	
3	VT_R8	絶対座標 (4 軸) が返却されます。	
4	VT_R8	絶対座標 (5 軸) が返却されます。	
5	VT_R8	絶対座標 (6 軸) が返却されます。	
6	VT_R8	絶対座標 (7 軸) が返却されます。	
7	VT_R8	絶対座標 (8 軸) が返却されます。	

型説明			
8	VT_R8	絶対座標 (P1 軸) が返却されます。	
9	VT_R8	絶対座標 (P2 軸) が返却されます。	
10	VT_R8	絶対座標 (P3 軸) が返却されます。	
11	VT_R8	絶対座標 (P4 軸) が返却されます。	

- 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.11. DISTANCE_TARGET<??>

「位置データ」ファイルデータ内の移動残を取得します。

使用例

' 変数追加

```
Dim var As CaoVariable
```

```
Set var = ctrl.AddVariable("DISTANCE_TARGET", "REFRESH=True")
```

' 値取得

```
Dim values() As Double
```

```
values = var.value
```

- オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

- データ型

型説明			
VT_ARRAY VT_R8			
0	VT_R8	移動残 (X 軸) が返却されます。	
1	VT_R8	移動残 (Y 軸) が返却されます。	
2	VT_R8	移動残 (Z 軸) が返却されます。	
3	VT_R8	移動残 (4 軸) が返却されます。	
4	VT_R8	移動残 (5 軸) が返却されます。	
5	VT_R8	移動残 (6 軸) が返却されます。	
6	VT_R8	移動残 (7 軸) が返却されます。	
7	VT_R8	移動残 (8 軸) が返却されます。	

型説明			
8	VT_R8		移動残 (P1 軸) が返却されます。
9	VT_R8		移動残 (P2 軸) が返却されます。
10	VT_R8		移動残 (P3 軸) が返却されます。
11	VT_R8		移動残 (P4 軸) が返却されます。

- 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.12. SPEED<??>

「位置データ」ファイルデータ内の送り速度を取得します。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("SPEED", "REFRESH=True")
' 値取得
Dim value As Double
value = var.value
```

- オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

- データ型

型説明	
VT_R8	送り速度が返却されます。

- 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.13. TOOL_NO<??>

「位置データ」ファイルデータ内の主軸工具番号を取得します。

使用例

```
Dim engine As caoEngine
' 変数追加
```

```
Dim var As CaoVariable
Set var = ctrl.AddVariable("TOOL_NO", "REFRESH=True")
' 値取得
Dim value As Byte
value = var.value
```

- オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= 〈ファイルデータ更新有無を 指定〉	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

- データ型

型説明	
VT_UI1	主轴工具番号が返却されます。

- 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.14. NEXT_TOOL_NO<??>

「位置データ」ファイルデータ内の次工具番号を取得します。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("NEXT_TOOL_NO", "REFRESH=True")
' 値取得
Dim value As Byte
value = var.value
```

- オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= 〈ファイルデータ更新有無を 指定〉	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

- データ型

型説明	
VT_UI1	次工具番号が返却されます。

4.3.1.15. MAGAGINE_NO<??>

「位置データ」ファイルデータ内のマガジン番号を取得します。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("MAGAGINE_NO", "REFRESH=True")
' 値取得
Dim value As Byte
value = var.value
```

● オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

● データ型

型説明	
VT_UI1	マガジン番号が返却されます。

● 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.16. ALARM_COUNT<??>

「現在アラーム」ファイルデータ内の有効アラーム数を取得します。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("ALARM_COUNT ", "REFRESH=True")
' 値取得
Dim value As Byte
value = var.value
```

● オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

● データ型

型説明	
VT_UI1	有効アラーム数が返却されます。

4.3.1.17. ALARM<??>

「現在アラーム」ファイルデータ内の現在アラーム情報を取得します。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("ALARM", "REFRESH=True")
' 値取得
Dim value As Variant
value = var.value
```

● オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

● データ型

型説明		
VT_ARRAY VT_VARIANT		
	n ¹¹	VT_ARRAY VT_UI2
		VT_UI2 発生個所が返却されます。
		VT_UI2 番号が返却されます。
		VT_UI2 補助番号が返却されます。

● 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.18. MODE<??>

「操作パネルデータ」ファイルデータ内のモードを取得します。

使用例

```
Dim engine As caoEngine
```

¹¹ n=35

' 変数追加

```
Dim var As CaoVariable
Set var = ctrl.AddVariable("MODE", "REFRESH=True")
```

' 値取得

```
Dim value As Byte
value = var.value
```

● オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= 〈ファイルデータ更新有無を 指定〉	○	ファイルデータ更新処理の有無を 指定します.	0 - 1	-

● データ型

型説明	
VT_UI1	モードが返却されます. 0:手動 1:MDI 運転 2:メモリ運転 3:プログラム編集 4:MDI 手動 5:運転中編集

● 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.19. NC_WORK_COUNTER<??>

「ワークカウンタ」ファイルデータ内の生産カウンタ (NC) を取得します.

使用例

' 変数追加

```
Dim var As CaoVariable
Set var = ctrl.AddVariable("NC_WORK_COUNTER", "REFRESH=True")
```

' 値取得

```
Dim values() As Long
values = var.value
```

● オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH=	○	ファイルデータ更新処理の有無を	0 - 1	-

オプション	必須	説明	値範囲	デフォルト値
<ファイルデータ更新有無を指定>		指定します.		

- データ型

型説明			
VT_ARRAY VT_I4			
0	VT_I4	カウント数(カウンタ 1)が返却されます.	
1	VT_I4	現在値(カウンタ 1)が返却されます.	
2	VT_I4	終了値(カウンタ 1)が返却されます.	
3	VT_I4	終了予告値(カウンタ 1)が返却されます.	
⋮			
12	VT_I4	カウント数(カウンタ 4)が返却されます.	
13	VT_I4	現在値(カウンタ 4)が返却されます.	
14	VT_I4	終了値(カウンタ 4)が返却されます.	
15	VT_I4	終了予告値(カウンタ 4)が返却されます.	

- 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.20. INTERACTIVE_WORK_COUNTER<??>

「ワークカウンタ」ファイルデータ内の生産カウンタ(対話)を取得します.

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("INTERACTIVE_WORK_COUNTER", "REFRESH=True")
' 値取得
Dim values() As Long
values = var.value
```

- オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= <ファイルデータ更新有無を指定>	○	ファイルデータ更新処理の有無を指定します.	0 - 1	-

● データ型

型説明			
VT_ARRAY VT_I4			
0	VT_I4	カウント数(カウンタ 1)が返却されます。	
1	VT_I4	現在値(カウンタ 1)が返却されます。	
2	VT_I4	終了値(カウンタ 1)が返却されます。	
3	VT_I4	終了予告値(カウンタ 1)が返却されます。	
⋮			
12	VT_I4	カウント数(カウンタ 4)が返却されます。	
13	VT_I4	現在値(カウンタ 4)が返却されます。	
14	VT_I4	終了値(カウンタ 4)が返却されます。	
15	VT_I4	終了予告値(カウンタ 4)が返却されます。	

● 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.21. OPE_TIME_COUNT<??>

「生産データ 2(時間表示)」ファイルデータ内の有効稼働時間記録数を取得します。

使用例

```

' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("OPE_TIME_COUNT", "REFRESH=True")
' 値取得
Dim value As Byte
value = var.value

```

● オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

● データ型

型説明	
VT_UI1	有効稼働時間記録数が返却されます。

● 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.22. OPE_TIME_HISTORY<??>

「生産データ 2(時間表示)」ファイルデータ内の稼働時間記録を取得します。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("OPE_TIME_HISTORY", "REFRESH=True")
' 値取得
Dim value As Variant
value = var.value
```

● オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

● データ型

型説明			
VT_ARRAY VT_VARIANT			
0	VT_BSTR	日付(稼働時間記録 1)が返却されます。	
1	VT_BSTR	稼働時間(稼働時間記録 1)が返却されます。	
2	VT_I4	運転回数(稼働時間記録 1)が返却されます。	
3	VT_I4	稼働率(稼働時間記録 1)が返却されます。	
⋮			
76	VT_BSTR	日付(稼働時間記録 20)が返却されます。	
77	VT_BSTR	稼働時間(稼働時間記録 20)が返却されます。	
78	VT_I4	運転回数(稼働時間記録 20)が返却されます。	
79	VT_I4	稼働率(稼働時間記録 20)が返却されます。	

● 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.23. HEAC_HISTORY_COUNT<??>

「熱変位自動補正」ファイルデータの熱変位自動補正履歴の中で、「SetHEACRange」メソッドで指定した範囲のデータの有効熱変位自動補正履歴数を取得します。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("HEAC_HISTORY_COUNT", "REFRESH=True")
' 値取得
Dim value As Variant
value = var.value
```

● オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

● データ型

型説明	
VT_UI2	有効熱変位自動補正履歴数が返却されます。

● 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.24. HEAC_Z_HISTORY<??>

「熱変位自動補正」ファイルデータ内の熱変位自動補正履歴 (Z 軸) を取得します。取得するデータの範囲は「SetHEACRange」メソッドで指定します。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("HEAC_Z_HISTORY", "REFRESH=True")
' 値取得
Dim value As Variant
value = var.value
```

● オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

● データ型

型説明		
VT_ARRAY VT_VARIANT		
0	VT_BSTR	日付 (履歴 m^{12}) が返却されます。
1	VT_BSTR	時刻 (履歴 m) が返却されます。
2	VT_I4	計算回数 (履歴 m) が返却されます。
3	VT_R8	平均速度 (履歴 m) が返却されます。
4	VT_R4	ボールねじ変位量 1 (履歴 m) が返却されます。
5	VT_R4	ボールねじ変位量 2 (履歴 m) が返却されます。
6	VT_UI4	主軸回転量累積値 (履歴 m) が返却されます。
7	VT_R4	主軸変位量 (履歴 m) が返却されます。
⋮		
$n \times 8 - 8$	VT_BSTR	日付 (履歴 $m+n^{13}-1$) が返却されます。
$n \times 8 - 7$	VT_BSTR	時刻 (履歴 $m+n-1$) が返却されます。
$n \times 8 - 6$	VT_I4	計算回数 (履歴 $m+n-1$) が返却されます。
$n \times 8 - 5$	VT_R8	平均速度 (履歴 $m+n-1$) が返却されます。
$n \times 8 - 4$	VT_R4	ボールねじ変位量 1 (履歴 $m+n-1$) が返却されます。
$n \times 8 - 3$	VT_R4	ボールねじ変位量 2 (履歴 $m+n-1$) が返却されます。
$n \times 8 - 2$	VT_UI4	主軸回転量累積値 (履歴 $m+n-1$) が返却されます。
$n \times 8 - 1$	VT_R4	主軸変位量 (履歴 $m+n-1$) が返却されます。

● 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.25. HEAC_X_HISTORY<??>

「熱変位自動補正」ファイルデータ内の熱変位自動補正履歴 (X 軸) を取得します。取得するデータの範囲は「SetHEACRange」メソッドで指定します。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("HEAC_X_HISTORY", "REFRESH=True")
' 値取得
```

¹² m = 「SetHEACRange」メソッドで指定したデータ範囲開始位置

¹³ n = 「SetHEACRange」メソッドで指定したデータ数

Dim value As Variant
value = var.value

● オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

● データ型

型説明		
VT_ARRAY VT_VARIANT		
0	VT_BSTR	日付 (履歴 m^{14}) が返却されます。
1	VT_BSTR	時刻 (履歴 m) が返却されます。
2	VT_I4	計算回数 (履歴 m) が返却されます。
3	VT_R8	平均速度 (履歴 m) が返却されます。
4	VT_R4	ボールねじ変位量 1 (履歴 m) が返却されます。
5	VT_R4	ボールねじ変位量 2 (履歴 m) が返却されます。
6	VT_UI4	主軸回転量累積値 (履歴 m) が返却されます。
7	VT_R4	主軸変位量 (履歴 m) が返却されます。
⋮		
$n \times 8 - 8$	VT_BSTR	日付 (履歴 $m+n^{15}-1$) が返却されます。
$n \times 8 - 7$	VT_BSTR	時刻 (履歴 $m+n-1$) が返却されます。
$n \times 8 - 6$	VT_I4	計算回数 (履歴 $m+n-1$) が返却されます。
$n \times 8 - 5$	VT_R8	平均速度 (履歴 $m+n-1$) が返却されます。
$n \times 8 - 4$	VT_R4	ボールねじ変位量 1 (履歴 $m+n-1$) が返却されます。
$n \times 8 - 3$	VT_R4	ボールねじ変位量 2 (履歴 $m+n-1$) が返却されます。
$n \times 8 - 2$	VT_UI4	主軸回転量累積値 (履歴 $m+n-1$) が返却されます。
$n \times 8 - 1$	VT_R4	主軸変位量 (履歴 $m+n-1$) が返却されます。

● 対応コマンド

コマンド種別+機能	LOD

¹⁴ m = 「SetHEACRange」メソッドで指定したデータ範囲開始位置

¹⁵ n = 「SetHEACRange」メソッドで指定したデータ数

4.3.1.26. HEAC_Y_HISTORY<??>

「熱変位自動補正」ファイルデータ内の熱変位自動補正履歴 (Y 軸) を取得します。取得するデータの範囲は「SetHEACRange」メソッドで指定します。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("HEAC_Y_HISTORY", "REFRESH=True")
' 値取得
Dim value As Variant
value = var.value
```

● オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= <ファイルデータ更新有無を 指定>	○	ファイルデータ更新処理の有無を 指定します。	0 - 1	-

● データ型

型説明			
VT_ARRAY VT_VARIANT			
0	VT_BSTR	日付 (履歴 m ¹⁶) が返却されます。	
1	VT_BSTR	時刻 (履歴 m) が返却されます。	
2	VT_I4	計算回数 (履歴 m) が返却されます。	
3	VT_R8	平均速度 (履歴 m) が返却されます。	
4	VT_R4	ボールねじ変位量 1 (履歴 m) が返却されます。	
5	VT_R4	ボールねじ変位量 2 (履歴 m) が返却されます。	
6	VT_UI4	主軸回転量累積値 (履歴 m) が返却されます。	
7	VT_R4	主軸変位量 (履歴 m) が返却されます。	
⋮			
n × 8 - 8	VT_BSTR	日付 (履歴 m+n ¹⁷ -1) が返却されます。	
n × 8 - 7	VT_BSTR	時刻 (履歴 m+n-1) が返却されます。	
n × 8 - 6	VT_I4	計算回数 (履歴 m+n-1) が返却されます。	
n × 8 - 5	VT_R8	平均速度 (履歴 m+n-1) が返却されます。	
n × 8 - 4	VT_R4	ボールねじ変位量 1 (履歴 m+n-1) が返却されます。	

¹⁶ m = 「SetHEACRange」メソッドで指定したデータ範囲開始位置

¹⁷ n = 「SetHEACRange」メソッドで指定したデータ数

型説明			
n × 8-3	VT_R4	ボールねじ変位量 2 (履歴 m+n-1) が返却されます。	
n × 8-2	VT_UI4	主軸回転量累積値 (履歴 m+n-1) が返却されます。	
n × 8-1	VT_R4	主軸変位量 (履歴 m+n-1) が返却されます。	

- 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.27. TOOL_DATA<??>

「工具データ」ファイルデータ内の工具データを取得します。

使用例

```

' 変数追加
Dim var As GaoVariable
Set var = ctrl.AddVariable("TOOL_DATA_M_1", "REFRESH=True, UNIT=1, NO=1")
' 値取得
Dim value As Variant
value = var.value

```

- オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= 〈ファイルデータ更新有無を指定〉	○	ファイルデータ更新処理の有無を指定します。	0 - 1	-
UNIT= 〈単位系を指定〉	-	ファイルデータの単位系を指定します。(0:現在の単位系, 1:メトリック, 2:インチ)	0 - 2	0
DATA_NO= 〈データ番号を指定〉	-	ファイルデータのデータ番号を指定します。(0:カレント面, 1~10:データ番号)	0 - 10	0

- データ型

型説明			
VT_ARRAY VT_VARIANT			
0	VT_BSTR	工具名 (工具番号 1) が返却されます。	
1	VT_R8	工具長オフセット (工具番号 1) が返却されます。	
2	VT_R8	工具長摩耗補正 (工具番号 1) が返却されます。	

型説明		
3	VT_R8	工具径オフセット(工具番号 1)が返却されます。
4	VT_R8	工具径摩耗補正(工具番号 1)が返却されます。
5	VT_I4	初期寿命(工具番号 1)が返却されます。
6	VT_I4	予告寿命(工具番号 1)が返却されます。
7	VT_I4	寿命(工具番号 1)が返却されます。
8	VT_R8	周速(工具番号 1)が返却されます。
9	VT_R4	回転送り(工具番号 1)が返却されます。
10	VT_R4	S 指定値(工具番号 1)が返却されます。
11	VT_R8	F 指定値(工具番号 1)が返却されます。
12	VT_I4	最高回転数(工具番号 1)が返却されます。
13	VT_BOOL	ツール洗浄(工具番号 1)が返却されます。
14	VT_BOOL	GTS(工具番号 1)が返却されます。
⋮		
1470	VT_BSTR	工具名(工具番号 99)が返却されます。
1471	VT_R8	工具長オフセット(工具番号 99)が返却されます。
1472	VT_R8	工具長摩耗補正(工具番号 99)が返却されます。
1473	VT_R8	工具径オフセット(工具番号 99)が返却されます。
1474	VT_R8	工具径摩耗補正(工具番号 99)が返却されます。
1475	VT_I4	初期寿命(工具番号 99)が返却されます。
1476	VT_I4	予告寿命(工具番号 99)が返却されます。
1477	VT_I4	寿命(工具番号 99)が返却されます。
1478	VT_R8	周速(工具番号 99)が返却されます。
1479	VT_R4	回転送り(工具番号 99)が返却されます。
1480	VT_R4	S 指定値(工具番号 99)が返却されます。
1481	VT_R8	F 指定値(工具番号 99)が返却されます。
1482	VT_I4	最高回転数(工具番号 99)が返却されます。
1483	VT_BOOL	ツール洗浄(工具番号 99)が返却されます。
1484	VT_BOOL	GTS(工具番号 99)が返却されます。

- 対応コマンド

コマンド種別+機能	LOD
-----------	-----

4.3.1.28. MACRO<??>

「マクロ変数」ファイルデータ内のマクロ変数(#500~#999)を取得します。

使用例

```
' 変数追加
Dim var As CaoVariable
Set var = ctrl.AddVariable("MACRO_M_1", "REFRESH=True, UNIT=1, NO=1")
' 値取得
Dim value As Variant
value = var.value
```

● オプション

オプション	必須	説明	値範囲	デフォルト値
REFRESH= 〈ファイルデータ更新有無を指定〉	○	ファイルデータ更新処理の有無を指定します。	0 - 1	-
UNIT= 〈単位系を指定〉	-	ファイルデータの単位系を指定します。(0:現在の単位系, 1:メトリック, 2:インチ)	0 - 2	0
DATA_NO= 〈データ番号を指定〉	-	ファイルデータのデータ番号を指定します。(0:カレント面, 1~10:データ番号)	0 - 10	0

● データ型

型説明			
VT_ARRAY VT_R8			
0	VT_R8	マクロ変数(#500)が返却されます。	
1	VT_R8	マクロ変数(#501)が返却されます。	
⋮			
498	VT_R8	マクロ変数(#998)が返却されます。	
499	VT_R8	マクロ変数(#999)が返却されます。	

● 対応コマンド

コマンド種別+機能	LOD
-----------	-----

5. 付録

付録A. エラーコード

本プロバイダには、独自のエラーコードが存在します。詳細は以下の表 5-1 独自エラーコード表をご参照ください。

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ご参照ください。

表 5-1 独自エラーコード表

エラー番号	説明
0x80110001	指定した信号種類が存在しません。
0x80110002	指定した数値の桁数、文字列の文字数が多すぎます。または、指定した信号番号が信号種類に適応した表記ではありません。
0x80110003	受信したメッセージのチェックサムが不正です。
0x80110004	ファイルデータが未取得の状態データの取得を行いました。
0x80110005	対応していないバージョンのマシニングセンタに接続しようとしてしました。

また、本プロバイダは、Protocol2 で通信を行う機器 (SPEEDIO 等) からの完了コードを“0x8010FF**”でマスクして返します。

完了コードについては、brother 社の「操作説明書(データ)」をご参照ください。

付録B. ファイルデータの読み込みの仕組み

NC 専用通信装置通信プロトコル方式 2 では、対象デバイスの保持しているファイルデータを取得するコマンドがあります。Brother.Protocol2 プロバイダではコマンド種別+機能が「LOD」となっているメソッド、変数が対応しています。

表 5-2 ファイルデータ - メソッド、変数対応表

ファイルデータ	メソッド	変数
PDSP	RefreshPDSP	MACHINE_POS<??>
	GetPDSP	RELATIVE_POS<??>
	GetMachinePOS	ABSOLUTE_POS<??>
	GetRelativePOS	DISTANCE_TARGET<??>
	GetAbsolutePOS	SPEED<??>
	GetDistanceTarget	TOOL_NO<??>

ファイルデータ	メソッド	変数
	GetSpeed GetToolNo GetNextToolNo GetMagagineNo	NEXT_TOOL_NO<??> MAGAGINE_NO<??>
ALARM	RefreshALARM GetALARM GetAlarmCount GetCurrentAlarm	ALARM_COUNT<??> ALARM<??>
PANEL	RefreshPANEL GetPANEL GetMode	MODE<??>MODE<??>
WKCNT	RefreshWKCNT GetWKCNT GetNCWorkCounter GetInteractiveWorkCounter	NC_WORK_COUNTER<??> INTERACTIVE_WORK_COUNTER<??>
PRDC2	RefreshPRDC2 GetPRDC2 GetOpeTimeHistoryCount GetOpeTimeHistory	OPE_TIME_COUNT<??> OPE_TIME_HISTORY<??>
HEAC	GetHEAC SetHEACRange GetHEACHistoryCount GetHEACZHistory GetHEACXHistory GetHEACYHistory	HEAC_HISTORY_COUNT<??> HEAC_Z_HISTORY<??> HEAC_X_HISTORY<??> HEAC_Y_HISTORY<??>
TOLS	RefreshTOLS GetTOLS GetToolData	TOOL_DATA<??>
MCRS	RefreshMCRS GetMCRS GetMacroVariable	MACRO<??>

上記メソッド、変数では、Brother.Protocol2 プロバイダを使用している PC のメモリ上に対象のファイルデータを保持し、そこから値を取得します。一度も取得していないファイルデータは PC のメモリ上に保持されていないため、値を取得しようとするとエラーとなります。

PCのメモリ上に保持されるデータは、データを更新するメソッド(Refresh**)を使用することで更新することができます。また、ファイルデータの一部、もしくは全部を取得するメソッドの引数、変数のオプションで取得時のファイルデータの更新の有無を指定できます。

データの更新はファイルデータ単位で行うため、同一ファイルデータ内の別データの取得によって更新した場合でも、そこで更新したデータを元に値を出力します。

付録C. PLC 信号種類

- 読み込み対象の PLC 信号種類

信号種類	意味	信号範囲
X	外部入力	000 - 3FF (16 進数で指定)
Y	外部出力	000 - 3FF (16 進数で指定)
BX	内部入力(ビット)	000 - 3FF (16 進数で指定)
BY	内部出力(ビット)	000 - 3FF (16 進数で指定)
BDX	内部入力(ワード)	000 - 255 (10 進数で指定)
BDY	内部出力(ワード)	000 - 255 (10 進数で指定)
M	内部リレー	0000 - 8191 (10 進数で指定)
D	データレジスタ	0000 - 2047 (10 進数で指定)
LM0	ローカルリレー高速1	0000 - 1023 (10 進数で指定)
LM1	ローカルリレー高速2	0000 - 1023 (10 進数で指定)
LM2	ローカルリレー低速1	0000 - 1023 (10 進数で指定)
LM3	ローカルリレー低速2	0000 - 1023 (10 進数で指定)
LD0	ローカルデータレジスタ高速1	0000 - 2047 (10 進数で指定)
LD1	ローカルデータレジスタ高速2	0000 - 2047 (10 進数で指定)
LD2	ローカルデータレジスタ低速1	0000 - 2047 (10 進数で指定)
LD3	ローカルデータレジスタ低速2	0000 - 2047 (10 進数で指定)
LTV0	ローカルタイマー-現在値高速1	000 - 255 (10 進数で指定)
LTV1	ローカルタイマー-現在値高速2	000 - 255 (10 進数で指定)
LTV2	ローカルタイマー-現在値低速1	000 - 255 (10 進数で指定)
LTV3	ローカルタイマー-現在値低速2	000 - 255 (10 進数で指定)
LTP0	ローカルタイマー-設定値高速1	000 - 255 (10 進数で指定)
LTP1	ローカルタイマー-設定値高速2	000 - 255 (10 進数で指定)
LTP2	ローカルタイマー-設定値低速1	000 - 255 (10 進数で指定)
LTP3	ローカルタイマー-設定値低速2	000 - 255 (10 進数で指定)
LCV0	ローカルカウンタ-現在値高速1	000 - 255 (10 進数で指定)
LCV1	ローカルカウンタ-現在値高速2	000 - 255 (10 進数で指定)

信号種類	意味	信号範囲
LCV2	ローカルカウンタ-現在値低速1	000 - 255 (10 進数で指定)
LCV3	ローカルカウンタ-現在値低速2	000 - 255 (10 進数で指定)
LCP0	ローカルカウンタ-設定値高速1	000 - 255 (10 進数で指定)
LCP1	ローカルカウンタ-設定値高速2	000 - 255 (10 進数で指定)
LCP2	ローカルカウンタ-設定値低速1	000 - 255 (10 進数で指定)
LCP3	ローカルカウンタ-設定値低速2	000 - 255 (10 進数で指定)

● 書込み可能な PLC 信号種類

信号種類	意味	信号範囲
Y	外部出力	000 - 3FF (16 進数で指定)
M	内部リレー	0000 - 8191 (10 進数で指定)
D	データレジスタ	0000 - 2047 (10 進数で指定)
LM0	ローカルリレー-高速1	0000 - 1023 (10 進数で指定)
LM1	ローカルリレー-高速2	0000 - 1023 (10 進数で指定)
LM2	ローカルリレー-低速1	0000 - 1023 (10 進数で指定)
LM3	ローカルリレー-低速2	0000 - 1023 (10 進数で指定)
LD0	ローカルデータレジスタ-高速1	0000 - 2047 (10 進数で指定)
LD1	ローカルデータレジスタ-高速2	0000 - 2047 (10 進数で指定)
LD2	ローカルデータレジスタ-低速1	0000 - 2047 (10 進数で指定)
LD3	ローカルデータレジスタ-低速2	0000 - 2047 (10 進数で指定)
LTV0	ローカルタイマ-現在値高速1	000 - 255 (10 進数で指定)
LTV1	ローカルタイマ-現在値高速2	000 - 255 (10 進数で指定)
LTV2	ローカルタイマ-現在値低速1	000 - 255 (10 進数で指定)
LTV3	ローカルタイマ-現在値低速2	000 - 255 (10 進数で指定)
LTP0	ローカルタイマ-設定値高速1	000 - 255 (10 進数で指定)
LTP1	ローカルタイマ-設定値高速2	000 - 255 (10 進数で指定)
LTP2	ローカルタイマ-設定値低速1	000 - 255 (10 進数で指定)
LTP3	ローカルタイマ-設定値低速2	000 - 255 (10 進数で指定)
LCV0	ローカルカウンタ-現在値高速1	000 - 255 (10 進数で指定)
LCV1	ローカルカウンタ-現在値高速2	000 - 255 (10 進数で指定)
LCV2	ローカルカウンタ-現在値低速1	000 - 255 (10 進数で指定)
LCV3	ローカルカウンタ-現在値低速2	000 - 255 (10 進数で指定)
LCP0	ローカルカウンタ-設定値高速1	000 - 255 (10 進数で指定)
LCP1	ローカルカウンタ-設定値高速2	000 - 255 (10 進数で指定)
LCP2	ローカルカウンタ-設定値低速1	000 - 255 (10 進数で指定)

信号種類	意味	信号範囲
LCP3	ローカルカウンタ-設定値低速2	000 - 255 (10 進数で指定)