

CaoTester2

Version 1.0.3

ユーザーズ ガイド

June 17, 2022

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0	2020-07-01	初版.
1.0.1	2020-09-10	レイアウトの読み込み, 保存, 終了処理を修正.
1.0.2	2021-04-14	空のファイルの値を取得した場合のログがエラーになってしまっていたのを修正. OnMessage イベント処理を修正. 画像データ取得時の処理を修正.
1.0.3	2022-06-17	Workspace ウィンドウのコントローラ名およびプロバイダ名のドロップダウンリストの検索機能を変更.

目次

1. はじめに.....	5
2. 画面説明.....	6
2.1. メイン画面.....	6
2.2. メニュー構成.....	9
2.2.1. File メニュー.....	9
2.2.2. View メニュー.....	9
2.2.3. Window メニュー.....	9
2.2.4. Window メニュー.....	10
2.3. Workspace ウィンドウ.....	11
2.3.1. Object タブ.....	11
2.3.2. Info タブ.....	13
2.4. Controller ウィンドウ.....	14
2.4.1. Object タブ.....	14
2.4.2. Property タブ.....	17
2.4.3. Info タブ.....	19
2.5. Variable ウィンドウ.....	20
2.5.1. Property タブ.....	20
2.5.2. Info タブ.....	23
2.6. Robot ウィンドウ.....	24
2.6.1. Object タブ.....	24
2.6.2. Property タブ.....	26
2.6.3. Operation タブ.....	29
2.6.4. Info タブ.....	31
2.7. File ウィンドウ.....	32
2.7.1. Object タブ.....	32
2.7.2. Value タブ.....	35
2.7.3. Property タブ.....	37
2.7.4. Operation タブ.....	41
2.7.5. Info タブ.....	42
2.8. Task ウィンドウ.....	43
2.8.1. Object タブ.....	43
2.8.2. Property タブ.....	45
2.8.3. Operation タブ.....	48

2.8.4. Info タブ	49
2.9. Extension ウィンドウ	50
2.9.1. Object タブ	50
2.9.2. Property タブ	52
2.9.3. Info タブ	54
2.10. Command ウィンドウ.....	55
2.10.1. Object タブ	55
2.10.2. Property タブ	58
2.10.3. Info タブ	60
2.11. Message ウィンドウ	61
2.12. Engine Status ウィンドウ	63
2.13. Engine ウィンドウ.....	66
2.14. オブジェクトウィンドウ.....	68
2.15. ログウィンドウ.....	71
2.16. Image ウィンドウ	72
2.17. Info タブ	75
3. 付録	76
付録 A. 書き込み値, 型情報の指定方法	76
付録 B. 使用できるデータ型	78
付録 C. License	79
付録 C.1. DockPanel Suite.....	79

1. はじめに

本書は ORiN2SDK で定義されている CAO のインターフェースを実装した総合テストツール CaoTester2 のユーザーズガイドです。

CaoTester2 では、プロバイダの変数の追加や削除、値の設定や取得、拡張コマンドの実行を行うことができます。

本書は、この CaoTester2 の機能について説明します。

2. 画面説明

2.1. メイン画面

CaoTester2 のメイン画面は以下のウィンドウで構成されています。

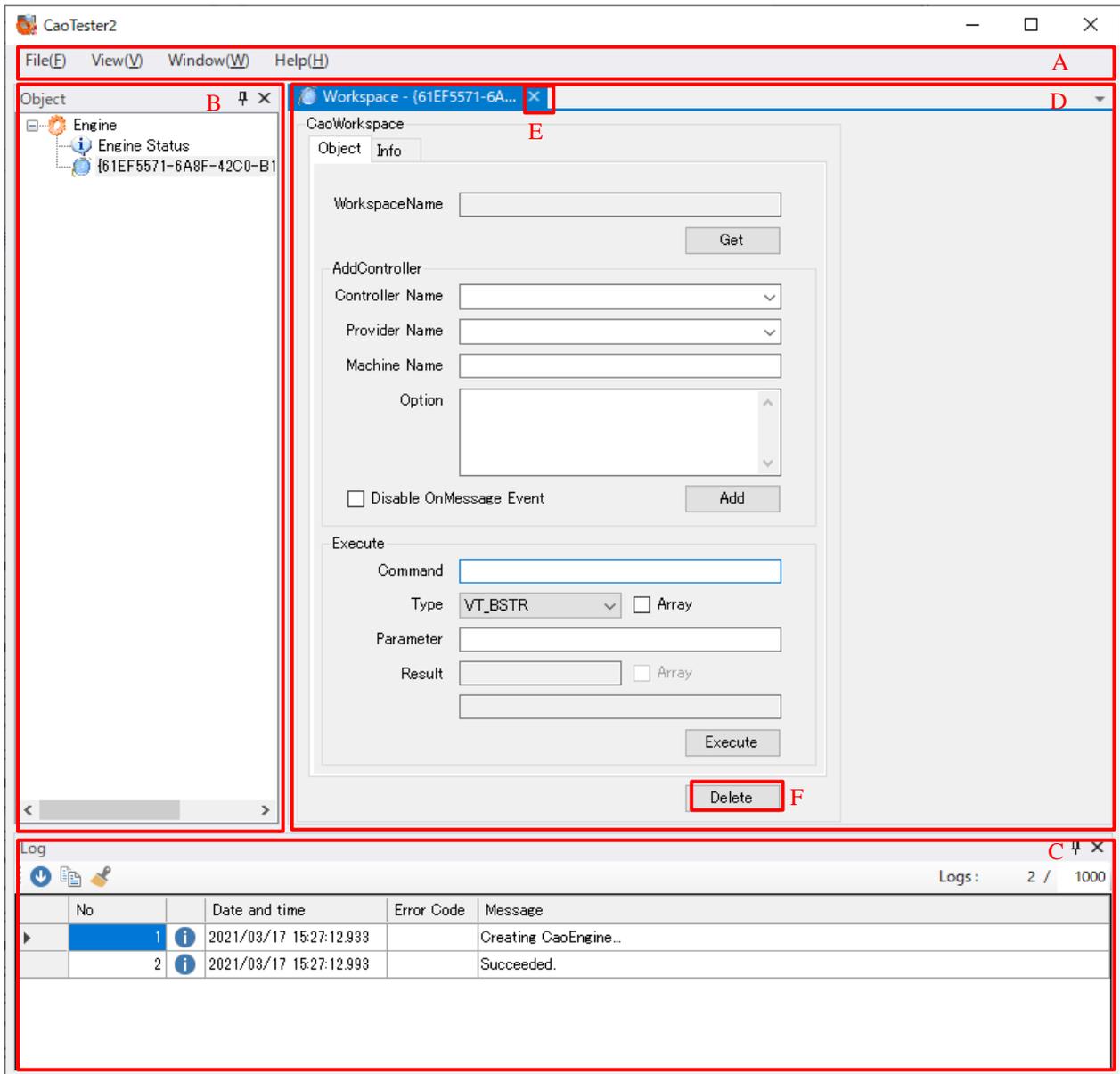


図 2-1 CaoTester2 メイン画面

A: メニューバー

各メニューが表示されます。詳細は 2.2 を参照してください。

B:オブジェクトウィンドウ

作成, 削除したオブジェクトがツリー状態で表示されます. 詳細は 2.14 を参照してください.

C:ログウィンドウ

行った操作やその結果がログとして記録されます. 詳細は 2.15 を参照してください.

D:アイテムウィンドウ

作成したオブジェクトに対して値の設定や取得等の操作を行うことができます.

E のアイテムウィンドウのクローズボタンはオブジェクトを破棄せずにウィンドウを非表示にします.

F の Delete ボタンはオブジェクトとウィンドウを破棄します.

アイテムウィンドウには表 2-1 のウィンドウが含まれます.

表 2-1 アイテムウィンドウ一覧

ウィンドウ名	参照先
Workspace ウィンドウ	2.3
Controller ウィンドウ	2.4
Variable ウィンドウ	2.5
Robot ウィンドウ	2.6
File ウィンドウ	2.7
Task ウィンドウ	2.8
Extension ウィンドウ	2.9
Command ウィンドウ	2.10
Message ウィンドウ	2.11
Engine Status ウィンドウ	2.12
Engine ウィンドウ	2.13

変数への値の書き込みや Execute の実行時には書き込む値の型情報を指定して書き込む値を明確にする必要があります. パラメータの型情報については付録 A を参照してください.

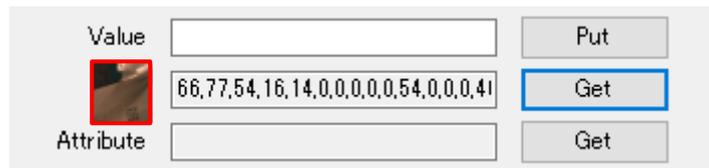


The image shows a dialog box titled "Execute". It contains the following fields and controls:

- Command: A text input field.
- Type: A dropdown menu currently showing "VT_BSTR".
- Array: A checkbox that is checked.
- Parameter: A text input field.
- Execute: A button to the right of the Parameter field.
- Result: A text input field.
- Array: A checkbox to the right of the Result field.

図 2-2 パラメータの型情報

変数の取得や Execute の実行結果が画像データとして認識できる場合、取得結果の左側にサムネイル画像が表示されます。サムネイル画像をクリックすると Image ウィンドウを表示することができます。詳細は 2.16 を参照してください。



The image shows a section of the interface with the following elements:

- Value: A text input field containing a small image thumbnail and the string "66,77,54,16,14,0,0,0,0,0,54,0,0,0,41".
- Put: A button to the right of the Value field.
- Attribute: A text input field below the Value field.
- Get: A button to the right of the Attribute field.

図 2-3 画像データ取得時のサムネイル表示

2.2. メニュー構成

CaoTester2 のメニュー項目について説明します。

2.2.1. File メニュー

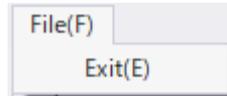


図 2-4 File メニュー

Exit

CaoTester2 を終了します。

2.2.2. View メニュー

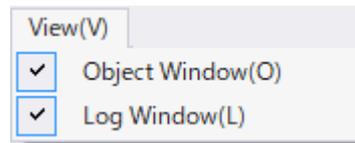


図 2-5 View メニュー

Object Window

Object Window の表示/非表示を切り替えます。

Log Window

Log Window の表示/非表示を切り替えます。

2.2.3. Window メニュー

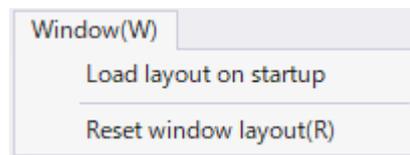


図 2-6 Window メニュー

Load layout on Startup

起動時に前回のウィンドウレイアウト情報の読み込みの On/Off を切り替えます。

Reset window layout

ウィンドウレイアウト情報をリセットします。

2.2.4. Window メニュー



図 2-7 Help メニュー

User's Guide

CaoTester2 のマニュアル(本ドキュメント)を開きます。

Version

CaoTester2 のバージョン情報を表示します。

2.3. Workspace ウィンドウ

Workspace ウィンドウではコントローラオブジェクトの作成や拡張コマンドの実行ができます。

2.3.1. Object タブ

Object タブではコントローラオブジェクトの作成や拡張コマンドの実行ができます。

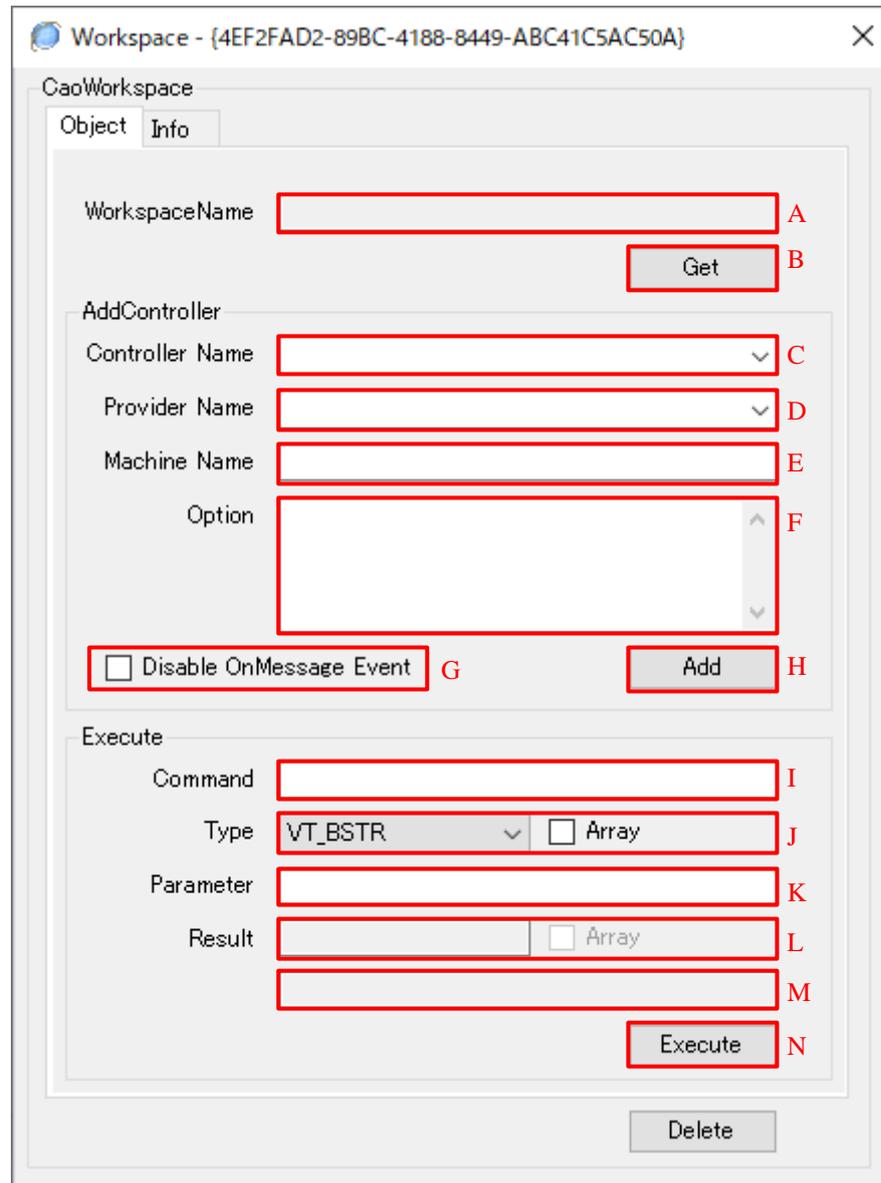


図 2-8 Workspace ウィンドウの Object タブ

A: ワークスペース名の取得の実行結果

B のワークスペース名の取得の実行結果を格納します。

B: ワークスペース名の取得

ワークスペース名を取得し実行結果を A に格納します。

C:コントローラ名

H の AddController 時のコントローラ名として使用します。

D:プロバイダ名

H の AddController 時のプロバイダ名として使用します。

E:マシン名

H の AddController 時のマシン名として使用します。

F:オプション

H の AddController 時のオプションとして使用します。

G:イベント許可フラグ

H の AddController 実行時のオプションに @EventEnable を付与します。

チェック状態の時に“@EventEnable=True”，を非チェック状態の時に“@EventEnable=False”を付与します。

@EventEnable=True の場合に OnMessage イベントを受信するようになります。

H:AddController の実行

C, D, E, F, G の値を使用して CaoWorkspace.AddController を実行します。

図 2-8 の例では以下ようになります。

```
AddContoller(“DS”, “CaoProv.DataStore”, “”, “”)
```

I:コマンド名

N の Execute 実行時のコマンド名として使用します。

J:パラメータの型情報

N の Execute 実行時のパラメータの型情報として使用します。

K:パラメータ文字列

N の Execute 実行時のパラメータ文字列として使用します。

L:Execute の実行結果の型情報

N の Execute 実行結果の型情報が格納されます。

M:Execute の実行結果

N の Execute 実行結果が格納されます。

N:Execute の実行

I, J, K の値を使用して Execute を実行します。

正常終了時の結果の型と値は L, M に格納されます。

2.3.2. Info タブ

Info タブではそのオブジェクトを作る時の情報を確認できます。詳細は 2.17 を参照してください。

2.4. Controller ウィンドウ

Controller ウィンドウでは変数、ロボット、ファイル、タスク、拡張ボード、コマンドオブジェクトの作成やメッセージの受信ができます。

2.4.1. Object タブ

Object タブでは変数、ロボット、ファイル、タスク、拡張ボード、コマンドオブジェクトの作成や拡張コマンドの実行、メッセージの取得ができます。

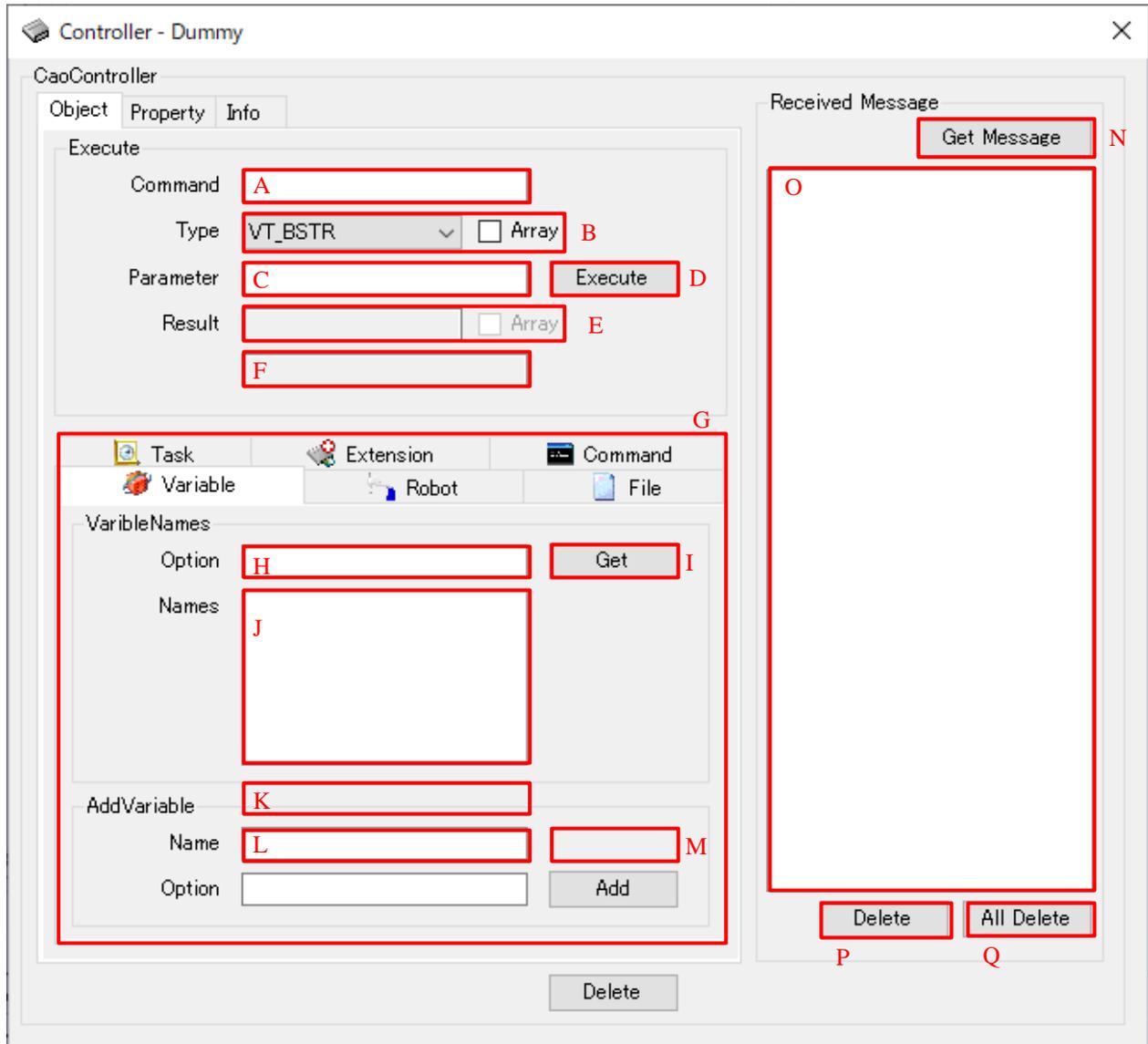


図 2-9 Controller ウィンドウの Object タブ

A: コマンド名

D の Execute 実行時のコマンド名として使用します。

B: パラメータの型情報

D の Execute 実行時のパラメータの型情報として使用します。

C:パラメータ文字列

D の Execute 実行時のパラメータ文字列として使用します。

D: Execute の実行

A, B, C の値を使用して Execute を実行します。

正常終了時の結果の型と値は E, F に格納されます。

E:Execute の実行結果の型情報

D の Execute 実行結果の型情報が格納されます。

F:Execute の実行結果

D の Execute 実行結果が格納されます。

G:オブジェクト作成グループ

オブジェクトの作成を行うグループです。

オブジェクト作成グループは変数, ロボット, ファイル, タスク, 拡張ボード, コマンドオブジェクトの全てが共通レイアウトです⁽¹⁾。

H:VariableNames のオプション⁽¹⁾

I の VariableNames 実行時のオプションとして使用します。

I:VariableNames の実行⁽¹⁾

VariableNames 実行結果が格納されます。

J:VariableNames の実行結果⁽¹⁾

I の VariableNames 実行結果が格納されます。

K:作成するオブジェクト名⁽¹⁾

M の AddVariable 実行時の変数名がして使用します。

L:作成するオブジェクトのオプション⁽¹⁾

M の AddVariable 実行時のオプションとして使用します。

¹ 代表として変数オブジェクトを作成する例として記述しています。

M:AddVariable の実行⁽¹⁾

K, L の値を使用して AddVariable を実行してオブジェクトを作成します。

N:メッセージの取得

エンジン内のメッセージキューからメッセージを取得します。

AddController 時の OnMessage イベント機能を無効にしている場合のみ取得できます。

OnMessage イベント機能が有効の状態では、O にメッセージが自動的に追加されていきます。

O:受信メッセージ一覧

取得したメッセージ一覧を表示します。

メッセージをダブルクリックすることで Message 内容を確認できます。詳細は 2.11 を参照してください。

P:メッセージの削除

選択したメッセージを削除します。

Q:受信メッセージ全削除

受信したメッセージを全削除します。

2.4.2. Property タブ

Property タブではコントローラオブジェクトのプロパティ情報の設定、取得ができます。

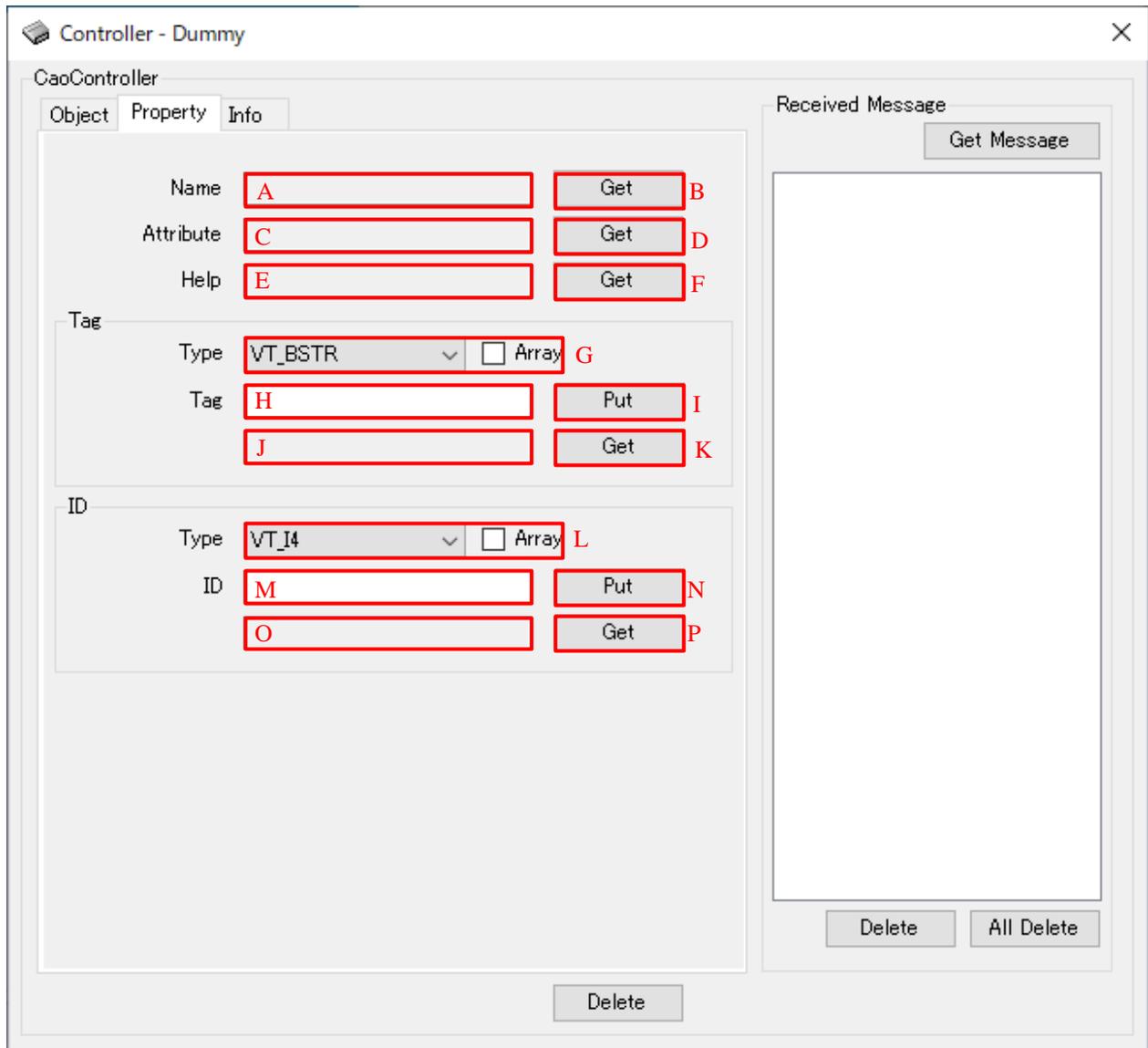


図 2-10 Controller ウィンドウの Property タブ

A:コントローラ名の取得の実行結果

B のコントローラ名の取得の実行結果を格納します。

B:コントローラ名の取得

コントローラ名を取得し実行結果を A に格納します。

C:属性の取得結果

D の属性の取得結果を格納します。

D:属性の取得

属性を取得し実行結果を C に格納します。

E:ヘルプの取得結果

F のヘルプの取得の実行結果を格納します。

F:ヘルプの取得

ヘルプを取得し実行結果を E に格納します。

G:タグの型情報

I の put_Tag 実行時のパラメータの型情報として使用します。

K の get_Tag 実行結果の型情報として格納されます。

H:タグデータ文字列

I の put_Tag 実行時のタグデータとして使用します。

I:タグの設定

put_Tag を実行します。

J:タグの取得結果

K の get_Tag の実行結果が格納されます。

K:タグの取得

get_Tag を実行し J に実行結果が格納されます。

型情報は I に格納されます。

L:ID の型情報

N の put_ID 実行時のパラメータの型情報として使用します。

P の get_ID 実行結果の型情報として格納されます。

M:ID 文字列

N の put_ID 実行時のタグデータとして使用します。

N:ID の設定

put_ID を実行します。

O:ID の取得結果

P の get_ID の実行結果が格納されます。

P:ID の取得

get_ID を実行し O に実行結果が格納されます。

型情報は L に格納されます。

2.4.3. Info タブ

Info タブではそのオブジェクトを作る時の情報を確認できます。詳細は 2.17 を参照してください。

2.5. Variable ウィンドウ

Variable ウィンドウでは変数オブジェクトの値の設定、取得ができます。

2.5.1. Property タブ

Property タブでは変数オブジェクトの値の設定、取得、プロパティ情報の設定、取得ができます。

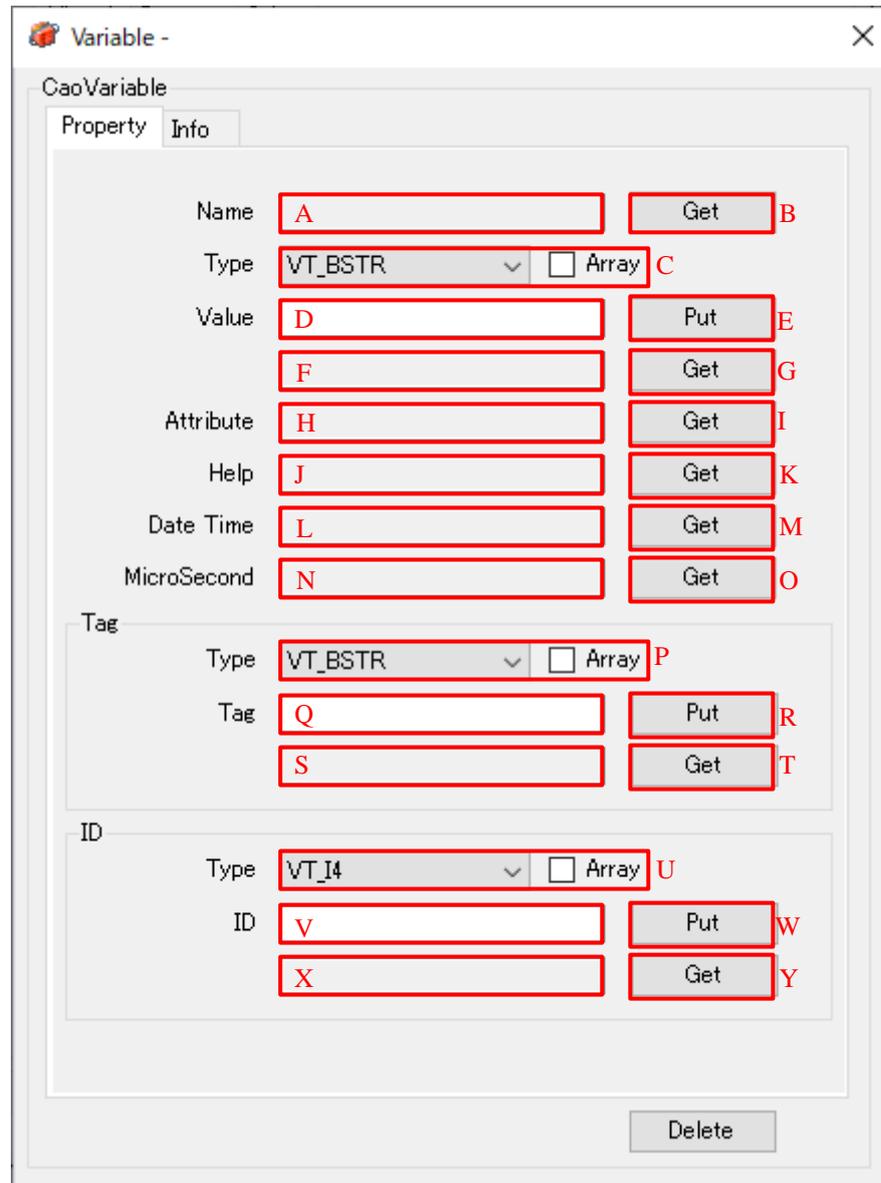


図 2-11 Variable ウィンドウの Property タブ

A: 変数名の取得の実行結果

B の変数名の取得の実行結果を格納します。

B: 変数名の取得

変数名を取得し実行結果を A に格納します。

C:変数値の型情報

E の put_Value 実行時のパラメータの型情報として使用します。

G の get_Value 実行結果の型情報として格納されます。

D:変数値データ文字列

E の put_Value 実行時のタグデータとして使用します。

E:変数値の設定

put_Value を実行します。

F:変数値の取得結果

G の get_Value の実行結果の結果文字列が格納されます。

G:変数値の取得

get_Value を実行し F に実行結果が格納されます。

型情報は C に格納されます。

H:属性の取得結果

I の属性の取得の実行結果を格納します。

I:属性の取得

属性を取得し実行結果を H に格納します。

J:ヘルプの取得結果

K のヘルプの取得の実行結果を格納します。

K:ヘルプの取得

ヘルプを取得し実行結果を J に格納します。

L:タイムスタンプ(日時)の取得結果

M のタイムスタンプ(日時)の取得の実行結果を格納します。

M: タイムスタンプ(日時)の取得

タイムスタンプ(日時)を取得し実行結果を L に格納します。

N:タイムスタンプ(マイクロ秒)の取得結果

O のタイムスタンプ(マイクロ秒)の取得の実行結果を格納します。

O: タイムスタンプ(マイクロ秒)の取得

タイムスタンプ(マイクロ秒)を取得し実行結果を N に格納します。

P: タグの型情報

R の put_Tag 実行時のパラメータの型情報として使用します。

T の get_Tag 実行結果の型情報として格納されます。

Q: タグデータ文字列

R の put_Tag 実行時のタグデータとして使用します。

R: タグの設定

put_Tag を実行します。

S: タグの取得結果

T の get_Tag の実行結果が格納されます。

T: タグの取得

get_Tag を実行し S に実行結果が格納されます。

型情報は P に格納されます。

U: ID の型情報

W の put_ID 実行時のパラメータの型情報として使用します。

Y の get_ID 実行結果の型情報として格納されます。

V: ID 文字列

W の put_ID 実行時のタグデータとして使用します。

W: ID の設定

put_ID を実行します。

X: ID の取得結果

Y の get_ID の実行結果が格納されます。

Y: ID の取得

get_ID を実行し X に実行結果が格納されます。

型情報は U に格納されます。

2.5.2. Info タブ

Info タブではそのオブジェクトを作る時の情報を確認できます。詳細は 2.17 を参照ください。

2.6. Robot ウィンドウ

Robot ウィンドウでは変数オブジェクト作成やロボット変数の値の設定, 取得, ロボット操作, 拡張コマンドの実行ができます。

2.6.1. Object タブ

Object タブでは変数オブジェクトの作成や拡張コマンドの実行ができます。

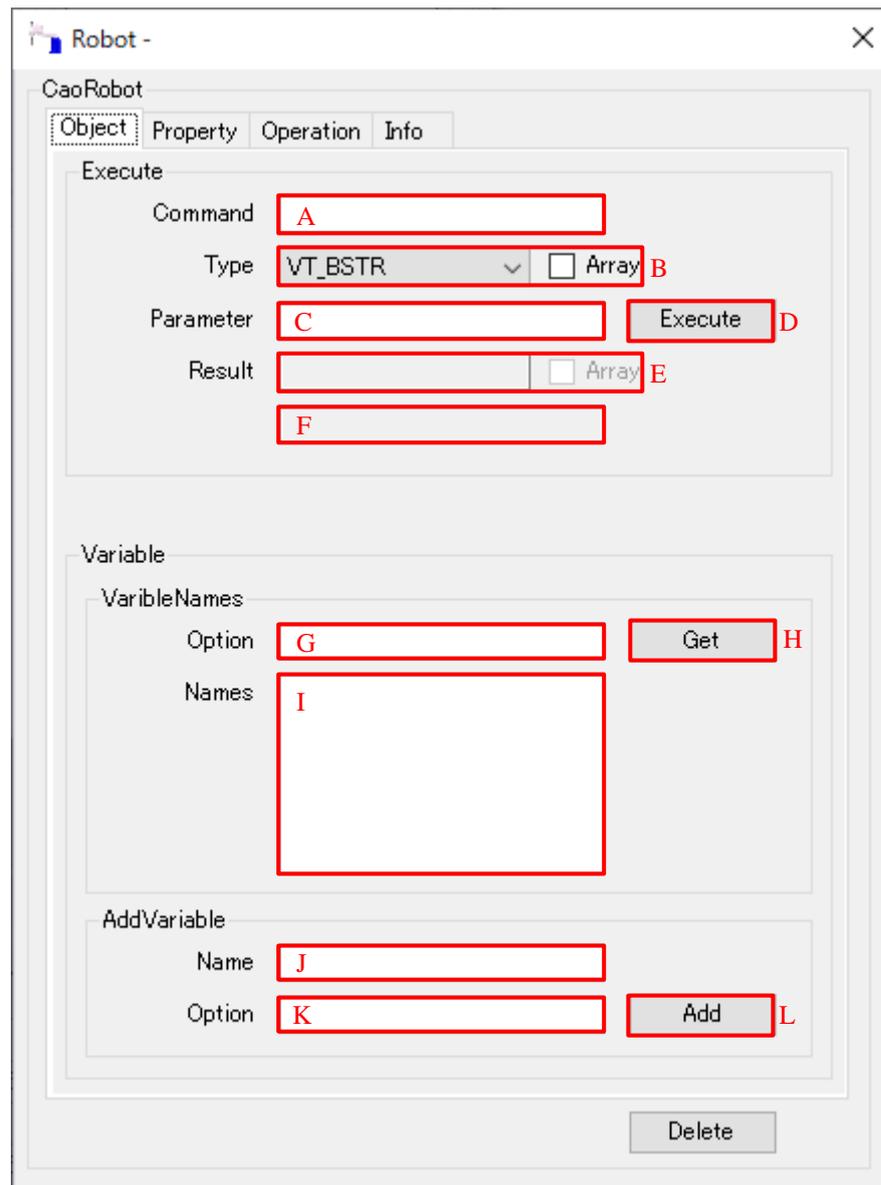


図 2-12 Robot ウィンドウの Object タブ

A: コマンド名

D の Execute 実行時のコマンド名として使用します。

B: パラメータの型情報

D の Execute 実行時のパラメータの型情報として使用します。

C:パラメータ文字列

D の Execute 実行時のパラメータ文字列として使用します。

D:Execute の実行

A, B, C の値を使用して Execute を実行します。

正常終了時の結果の型と値は E, F に格納されます。

E:Execute の実行結果の型情報

D の Execute 実行結果の型情報が格納されます。

F:Execute の実行結果

D の Execute 実行結果が格納されます。

G:VariableNames のオプション

H の VariableNames 実行時のオプションとして使用します。

H:VariableNames の実行

VariableNames を実行します。

I に実行結果が格納されます。

I:VariableNames の実行結果

H の VariableNames 実行結果が格納されます。

J:作成するオブジェクト名

L の AddVariable 実行時の変数名がして使用します。

K:作成するオブジェクトのオプション

L の AddVariable 実行時のオプションとして使用します。

L:AddVariable の実行

J, K の値を使用して AddVariable を実行してオブジェクトを作成します。

2.6.2. Property タブ

Property タブではロボットオブジェクトのプロパティ情報を設定、取得ができます。

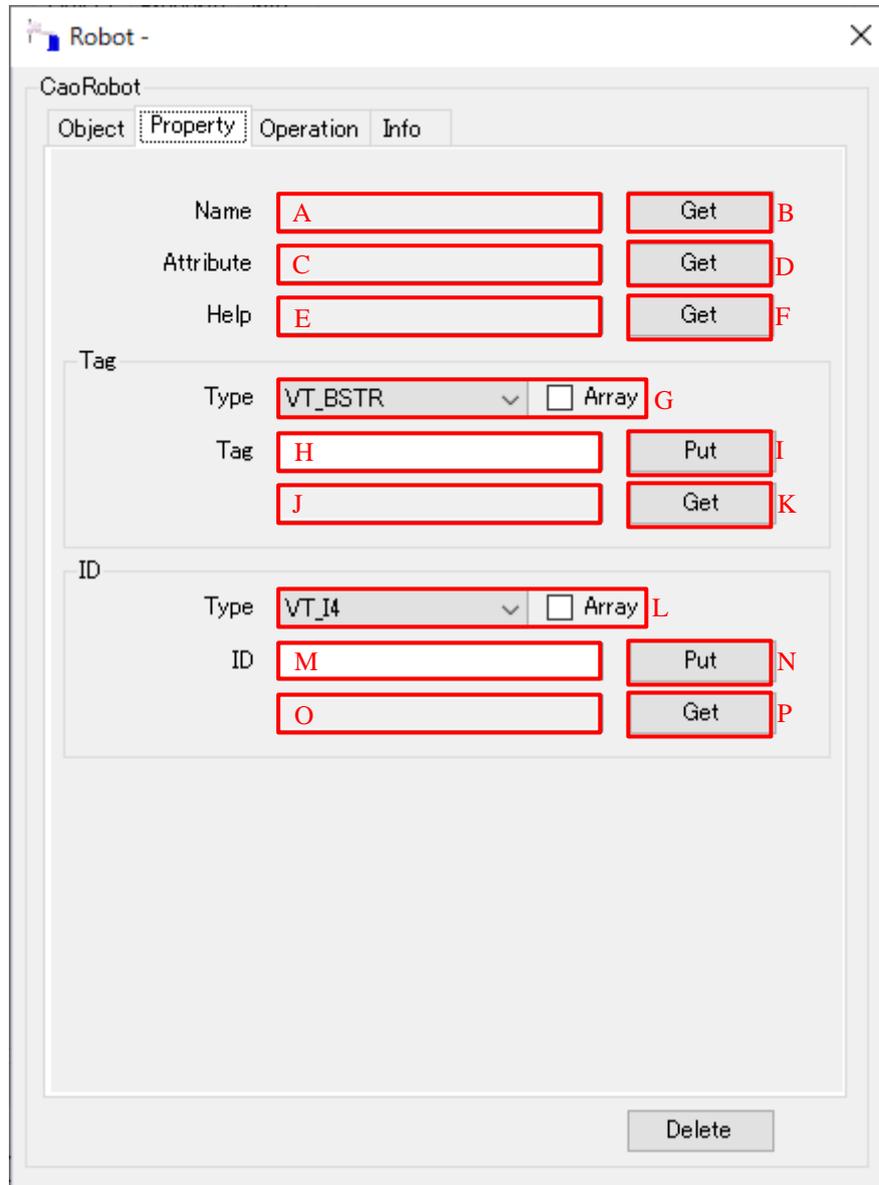


図 2-13 Robot ウィンドウの Property タブ

A: ロボット名の取得の実行結果

B のロボット名の取得の実行結果を格納します。

B: ロボット名の取得

ロボット名を取得し実行結果を A に格納します。

C: 属性の取得結果

D の属性の取得の実行結果を格納します。

D:属性の取得

属性を取得し実行結果を C に格納します。

E:ヘルプの取得結果

F のヘルプの取得の実行結果を格納します。

F:ヘルプの取得

ヘルプを取得し実行結果を E に格納します。

G:タグの型情報

I の put_Tag 実行時のパラメータの型情報として使用します。

K の get_Tag 実行結果の型情報として格納されます。

H:タグデータ文字列

I の put_Tag 実行時のタグデータとして使用します。

I:タグの設定

put_Tag を実行します。

J:タグの取得結果

K の get_Tag の実行結果が格納されます。

K:タグの取得

get_Tag を実行し J に実行結果が格納されます。

型情報は G に格納されます。

L:ID の型情報

N の put_ID 実行時のパラメータの型情報として使用します。

P の get_ID 実行結果の型情報として格納されます。

M:ID 文字列

N の put_ID 実行時のタグデータとして使用します。

N:ID の設定

put_ID を実行します。

O:ID の取得結果

P の get_ID の実行結果が格納されます。

P:ID の取得

get_ID を実行し O に実行結果が格納されます。

型情報は L に格納されます。

2.6.3. Operation タブ

Operation タブではロボットの速度設定や操作ができます。

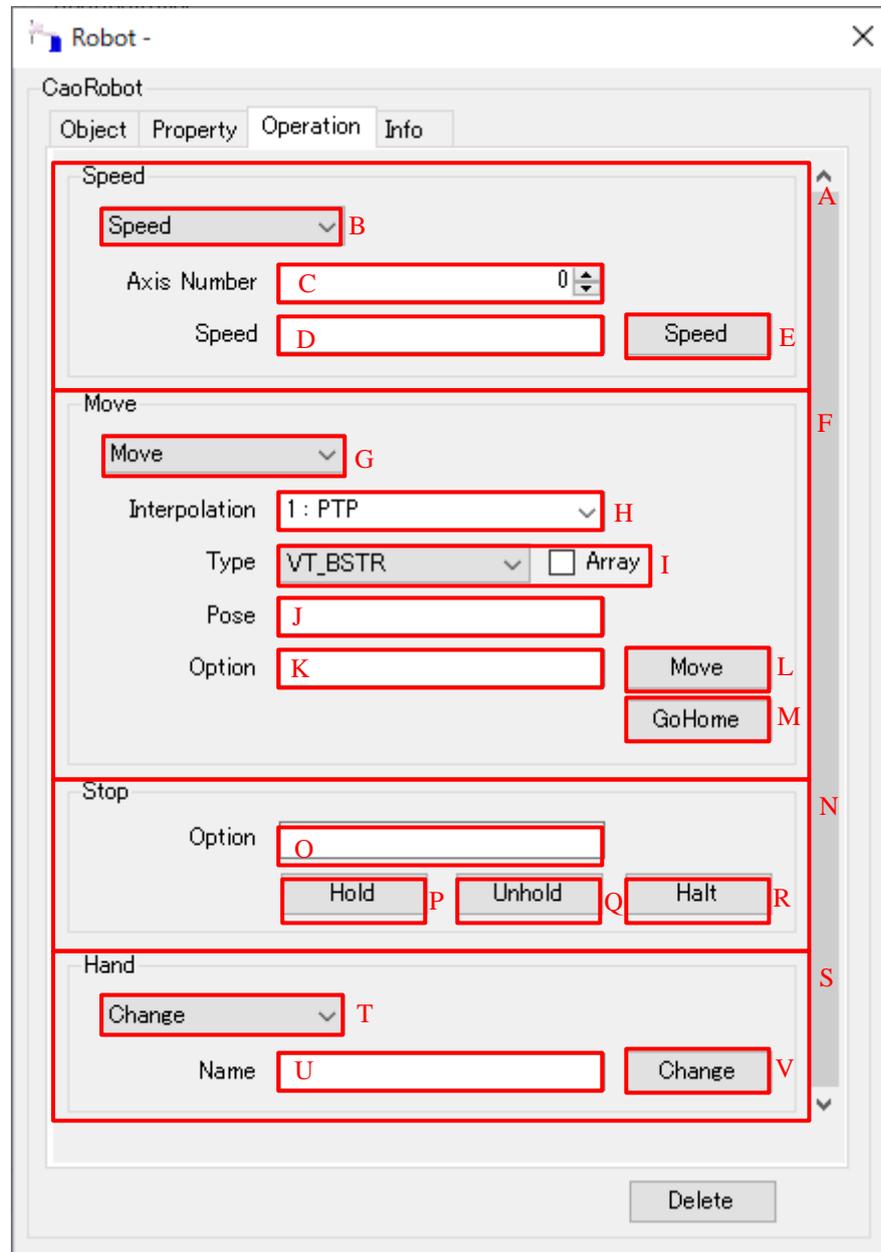


図 2-14 Robot ウィンドウの Operation タブ

A:ロボット速度設定グループ

ロボットの速度設定を行うグループです。

B の種別で設定方法を決定します。

B:速度設定種別

速度設定方法を決めます。

Accelerate, Speed を選択することが可能です⁽²⁾.

C:軸番号

E の Speed 実行時の第一引数として使用します.

D:速度

E の Speed 実行時の第二引数として使用します.

E:Speed の実行

C, D の値を使用して Speed を実行します.

F:ロボット移動グループ

ロボットの移動を行うグループです.

G の種別で設定方法を決定します.

G:ロボット移動種別

移動方法を決めます.

Drive, Move, Rotate を選択することが可能です⁽³⁾.

H:補完指定

L の Move 実行時の第一引数として使用します.

I:ポーズ列の型情報

L の Move 実行時の第二引数のポーズ列の型情報として使用します.

J:ポーズ列

L の Move 実行時の第二引数のポーズ列として使用します.

K:動作オプション

L の Move 実行時の第三引数として使用します.

L:Move の実行

H, I, J, K の値を使用して Move を実行します.

M:GoHome の実行

² 代表で Speed の説明を行います.

³ 代表で Move の説明を行います.

GoHome を実行します。

N:ロボット停止グループ

ロボットの停止を行うグループです。

O:停止オプション

P, Q, R のオプションとして使用します。

P:Hold の実行

O の値を使用して Hold を実行します。

Q:Unhold の実行

O の値を使用して Unhold を実行します。

R: Halt の実行

O の値を使用して Halt を実行します。

S:ロボットハンド操作グループ

ロボットのハンド操作を行うグループです。

T の種別で設定方法を決定します。

T:ロボットハンド操作種別

ハンド操作方法を決めます。

Change, Chuck / Unchuck を選択することが可能です⁽⁴⁾。

U:ハンド名

V の Change 実行時の第一引数として使用します。

V:Change の実行

U の値を使用して Change を実行します。

2.6.4. Info タブ

Info タブではそのオブジェクトを作る時の情報を確認できます。詳細は 2.17 を参照ください。

⁴ 代表で Change の説明を行います。

2.7. File ウィンドウ

File ウィンドウではファイルオブジェクト作成やファイル変数の作成, ファイルの値の設定, 取得, ファイル操作, 拡張コマンドの実行ができます。

2.7.1. Object タブ

Object タブでは変数, ファイルオブジェクトの作成や拡張コマンドの実行ができます。

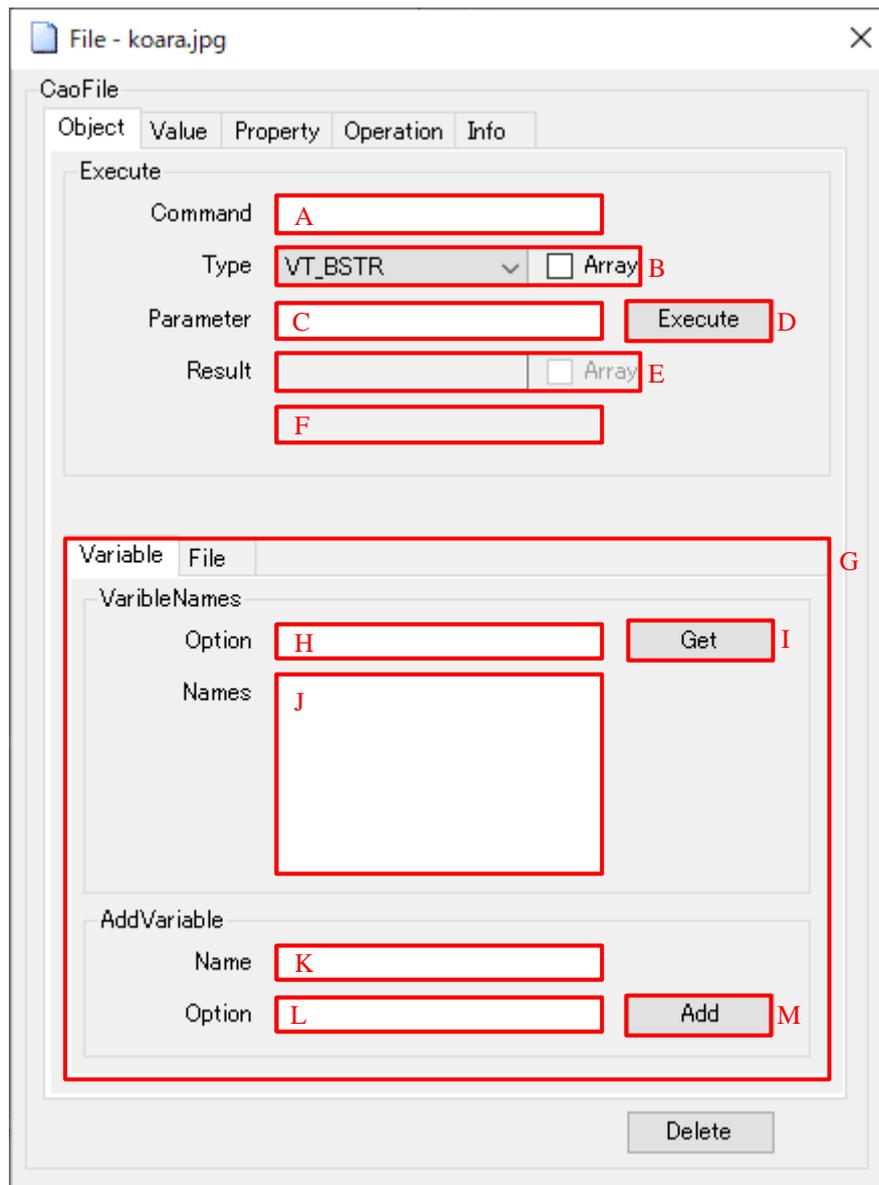


図 2-15 File ウィンドウの File タブ

A:コマンド名

D の Execute 実行時のコマンド名として使用します。

B:パラメータの型情報

D の Execute 実行時のパラメータの型情報として使用します。

C:パラメータ文字列

D の Execute 実行時のパラメータ文字列として使用します。

D:Execute の実行

A, B, C の値を使用して Execute を実行します。

正常終了時の結果の型と値は E, F に格納されます。

E:Execute の実行結果型情報

D の Execute 実行結果の型情報が格納されます。

F:Execute の実行結果

D の Execute 実行結果が格納されます。

G:オブジェクト作成グループ

オブジェクトの作成を行うグループです。

オブジェクト作成グループは変数, ファイルオブジェクトの全てが共通レイアウトです⁽⁵⁾。

H:VariableNames のオプション⁽¹⁾

I の VariableNames 実行時のオプションとして使用します。

I:VariableNames の実行⁽¹⁾

VariableNames 実行結果が格納されます。

J:VariableNames の実行結果⁽¹⁾

I の VariableNames 実行結果が格納されます。

K:作成するオブジェクト名⁽¹⁾

M の AddVariable 実行時の変数名がして使用します。

L:作成するオブジェクトのオプション⁽¹⁾

M の AddVariable 実行時のオプションとして使用します。

M:AddVariable の実行⁽¹⁾

⁵ 代表として変数オブジェクトを作成する例として記述しています。

K, L の値を使用して `AddVariable` を実行してオブジェクトを作成します.

2.7.2. Value タブ

Value タブではファイルオブジェクトの値を設定、取得ができます。

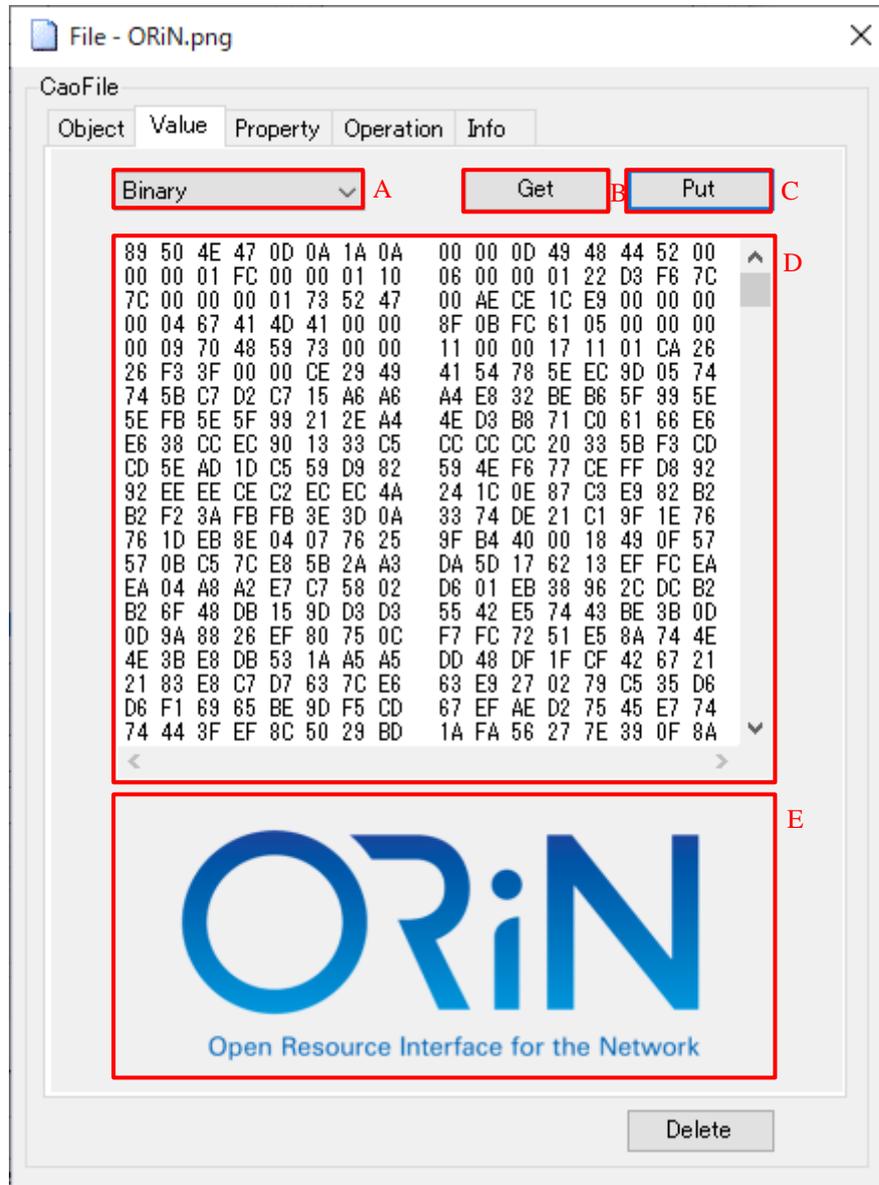


図 2-16 File ウィンドウの Value タブ

A:オープンモード

B のファイルの内容の取得の実行結果を格納します。

C のファイルの内容の設定にオープンモードとして使用します。

B:ファイルの内容の取得

get_Value を実行し結果を A, C に格納します。

画像データの場合は E にサムネイルを表示します。

サムネイル画像をクリックすることで Image ウィンドウを表示できます。詳細は 2.16 を参照してください。

C:ファイルの内容の設定

A, D の内容を使用して put_Value を実行します.

D:ファイルの内容

B の get_Value を実行した結果が格納されます.

C の put_Value 実行時に使用します.

E:ヘルプの取得結果

B の get_Value を実行した結果が画像ファイルの場合にサムネイル画像として表示します.

サムネイル画像をクリックすることで Image ウィンドウを表示できます. 詳細は 2.16 を参照してください.

2.7.3. Property タブ

Property タブではファイルオブジェクトのプロパティ情報を設定、取得ができます。

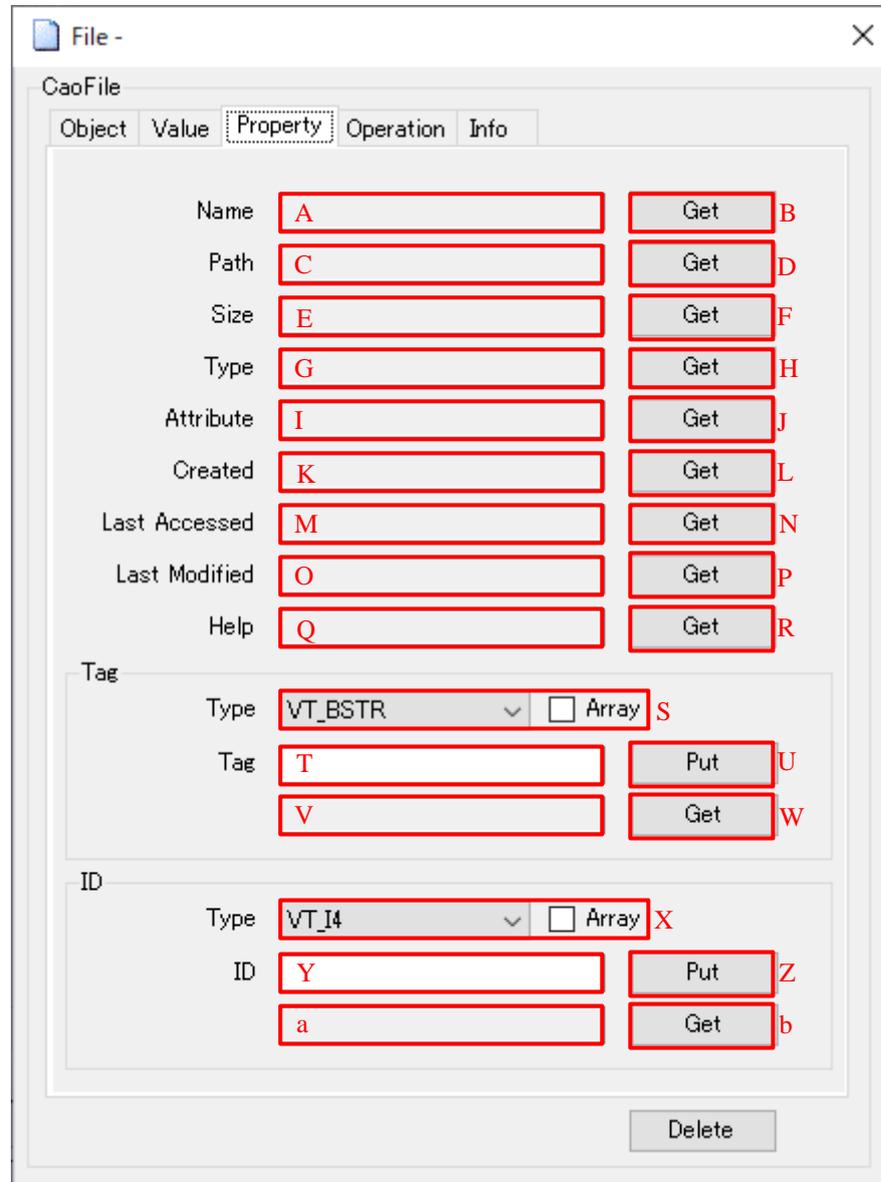


図 2-17 File ウィンドウの Property タブ

A: ファイル名の取得の実行結果

B のファイル名の取得の実行結果を格納します。

B: ファイル名の取得

ファイル名を取得し実行結果を A に格納します。

C: ファイルのパスの取得の実行結果

D のファイルのパスの取得の実行結果を格納します。

D:ファイルのパスの取得

ファイルのパスを取得し実行結果を C に格納します。

E:ファイルサイズの取得の実行結果

F のファイルサイズの取得の実行結果を格納します。

F:ファイルサイズの取得の実行結果

E のファイルサイズの取得の実行結果を格納します。

G:ファイルのタイプの取得の実行結果

H のファイルのタイプの取得の実行結果を格納します。

H:ファイルのタイプの取得

ファイルのタイプを取得し実行結果を G に格納します。

I:属性の取得結果

J の属性の取得の実行結果を格納します。

J:属性の取得

属性を取得し実行結果を I に格納します。

K:作成日時の取得結果

L の作成日時の取得の実行結果を格納します。

L:作成日時の取得

作成日時を取得し実行結果を K に格納します。

M:最終アクセス日時の取得結果

N の最終アクセス日時の取得の実行結果を格納します。

N:最終アクセス日時の取得

最終アクセス日時を取得し実行結果を M に格納します。

O:最終変更日時の取得結果

P の最終変更日時の取得の実行結果を格納します。

P:最終変更日時の取得

最終変更日時を取得し実行結果を O に格納します。

Q:ヘルプの取得結果

R のヘルプの取得の実行結果を格納します。

R:ヘルプの取得

ヘルプを取得し実行結果を Q に格納します。

S:タグの型情報

U の put_Tag 実行時のパラメータの型情報として使用します。

W の get_Tag 実行結果の型情報として格納されます。

T:タグデータ文字列

U の put_Tag 実行時のタグデータとして使用します。

U:タグの設定

put_Tag を実行します。

V:タグの取得結果

W の get_Tag の実行結果が格納されます。

W:タグの取得

get_Tag を実行し V に実行結果が格納されます。

型情報は S に格納されます。

X:ID の型情報

Z の put_ID 実行時のパラメータの型情報として使用します。

b の get_ID 実行結果の型情報として格納されます。

Y:ID 文字列

Z の put_ID 実行時のタグデータとして使用します。

Z:ID の設定

put_ID を実行します。

a:ID の取得結果

b の get_ID の実行結果が格納されます。

b:ID の取得

get_ID を実行し a に実行結果が格納されます。

型情報は X に格納されます。

2.7.4. Operation タブ

Operation タブではファイルの移動やコピー、削除、タスクの生成ができます。

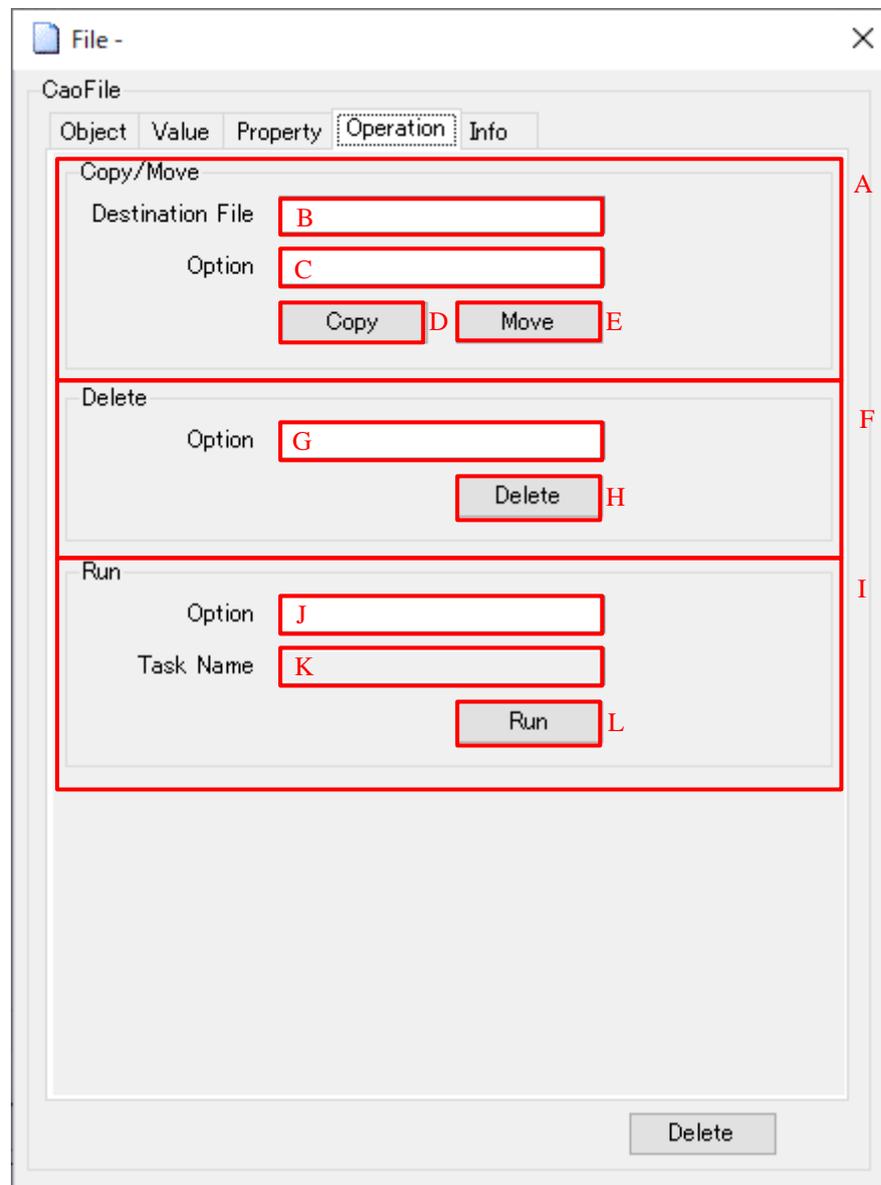


図 2-18 File ウィンドウの Operation タブ

A: ファイルのコピー、移動グループ

ファイルのコピー、移動を行うグループです。

B: コピー、移動先

D, E の Copy, Move 実行時の第一引数として使用します。

C: コピー移動先オプション

D, E の Copy, Move 実行時の第二引数として使用します。

D:ファイルのコピー

A, B の値を使用して Copy を実行します.

E:ファイルの移動

A, B の値を使用して Move を実行します.

F:ファイルの削除グループ

ファイルの削除を行うグループです.

G:ファイルの削除オプション

H の Delete 実行時の第一引数として使用します.

H:ファイルの削除

G の値を使用して Delete を実行します.

I:タスクの生成グループ

タスクの生成を行うグループです.

J:タスクの生成オプション

L の Run 実行時の第一引数として使用します.

K:タスクの取得結果

L の Run の実行結果を格納します.

L:タスクの生成

J の値を使用して Run を実行し結果を K に格納します.

2.7.5. Info タブ

Info タブではそのオブジェクトを作る時の情報を確認できます. 詳細は 2.17 を参照ください.

2.8. Task ウィンドウ

Task ウィンドウでは変数オブジェクト作成, タスク操作, 拡張コマンドの実行を行うことができます。

2.8.1. Object タブ

Object タブでは変数オブジェクトの作成や拡張コマンドの実行ができます。

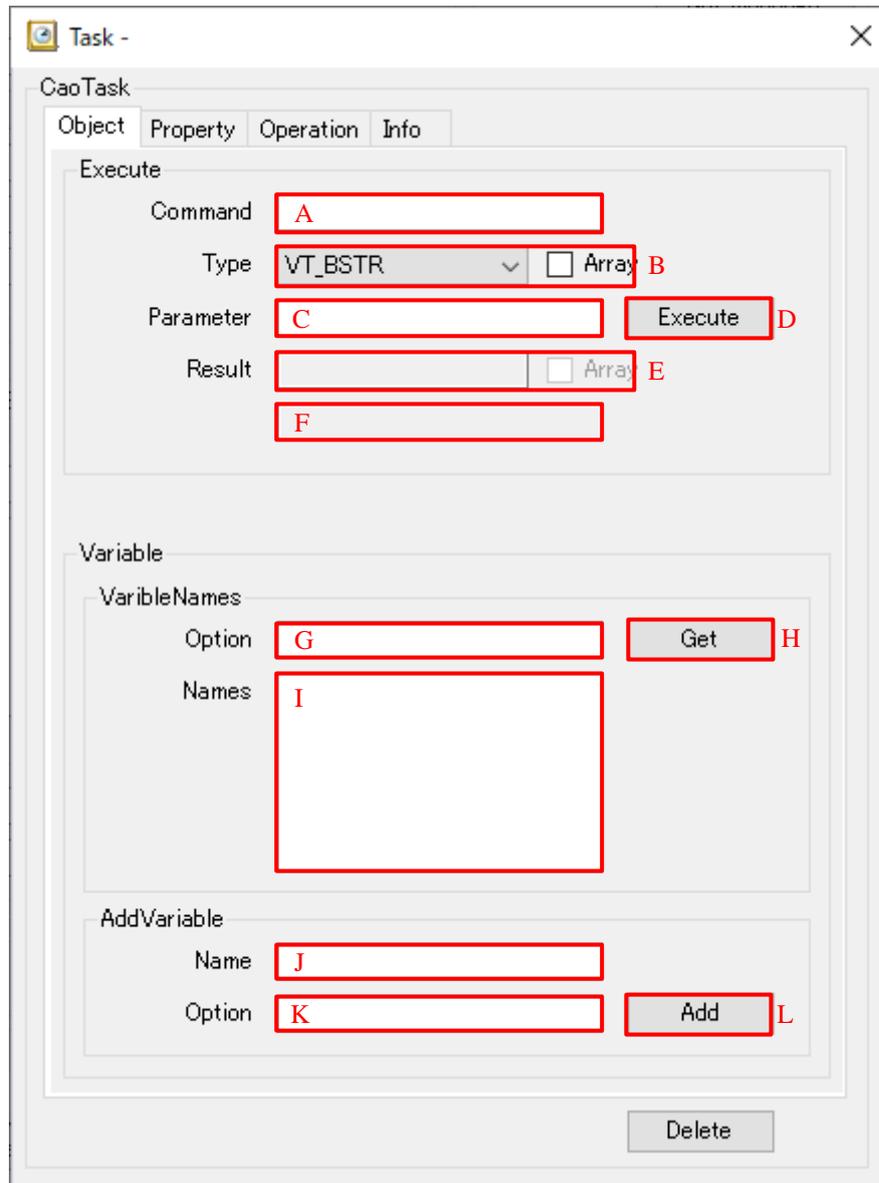


図 2-19 Task ウィンドウの Object タブ

A: コマンド名

D の Execute 実行時のコマンド名として使用します。

B: パラメータの型情報

D の Execute 実行時のパラメータの型情報として使用します。

C:パラメータ文字列

D の Execute 実行時のパラメータ文字列として使用します。

D:Execute の実行

A, B, C の値を使用して Execute を実行します。

正常終了時の結果の型と値は E, F に格納されます。

E:Execute の実行結果の型情報

D の Execute 実行結果の型情報が格納されます。

F:Execute の実行結果の結果文字列

D の Execute 実行結果が格納されます。

G:VariableNames のオプション

H の VariableNames 実行時のオプションとして使用します。

H:VariableNames の実行

VariableNames を実行します。

I に実行結果が格納されます。

I:VariableNames の実行結果

H の VariableNames 実行結果が格納されます。

J:作成するオブジェクト名

L の AddVariable 実行時の変数名がして使用します。

K:作成するオブジェクトのオプション

L の AddVariable 実行時のオプションとして使用します。

L:AddVariable の実行

J, K の値を使用して AddVariable を実行してオブジェクトを作成します。

2.8.2. Property タブ

Property タブではタスクオブジェクトのプロパティ情報を設定、取得ができます。

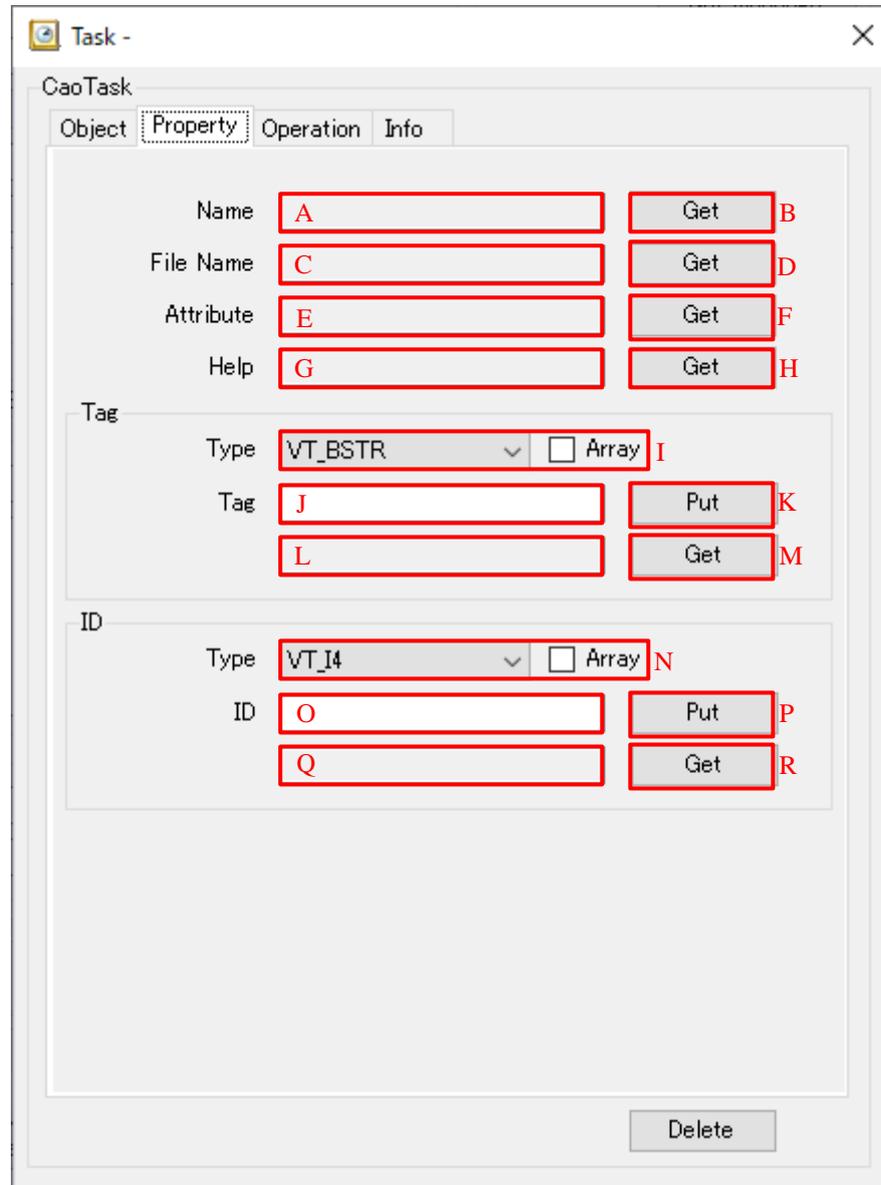


図 2-20 Task ウィンドウの Property タブ

A:タスク名の取得の実行結果

B のタスク名の取得の実行結果を格納します。

B:タスク名の取得

タスク名を取得し実行結果を A に格納します。

C:対応ファイル名の取得の実行結果

D の対応ファイル名の取得の実行結果を格納します。

D:対応ファイル名の取得

対応ファイル名を取得し実行結果を C に格納します。

E:属性の取得結果

F の属性の取得の実行結果を格納します。

F:属性の取得

属性を取得し実行結果を E に格納します。

G:ヘルプの取得結果

H のヘルプの取得の実行結果を格納します。

H:ヘルプの取得

ヘルプを取得し実行結果を G に格納します。

I:タグの型情報

K の put_Tag 実行時のパラメータの型情報として使用します。

M の get_Tag 実行結果の型情報として格納されます。

J:タグデータ文字列

K の put_Tag 実行時のタグデータとして使用します。

K:タグの設定

put_Tag を実行します。

L:タグの取得結果

M の get_Tag の実行結果の結果文字列が格納されます。

M:タグの取得

get_Tag を実行し L に実行結果の結果文字列が格納されます。

型情報は I に格納されます。

N:ID の型情報

P の put_ID 実行時のパラメータの型情報として使用します。

R の get_ID 実行結果の型情報として格納されます。

O:ID 文字列

P の put_ID 実行時のタグデータとして使用します。

P:ID の設定

put_ID を実行します。

Q:ID の取得結果

R の get_ID の実行結果が格納されます。

R:ID の取得

get_ID を実行し Q に実行結果が格納されます。

型情報は N に格納されます。

2.8.3. Operation タブ

Operation タブではタスク開始や停止, 削除ができます.

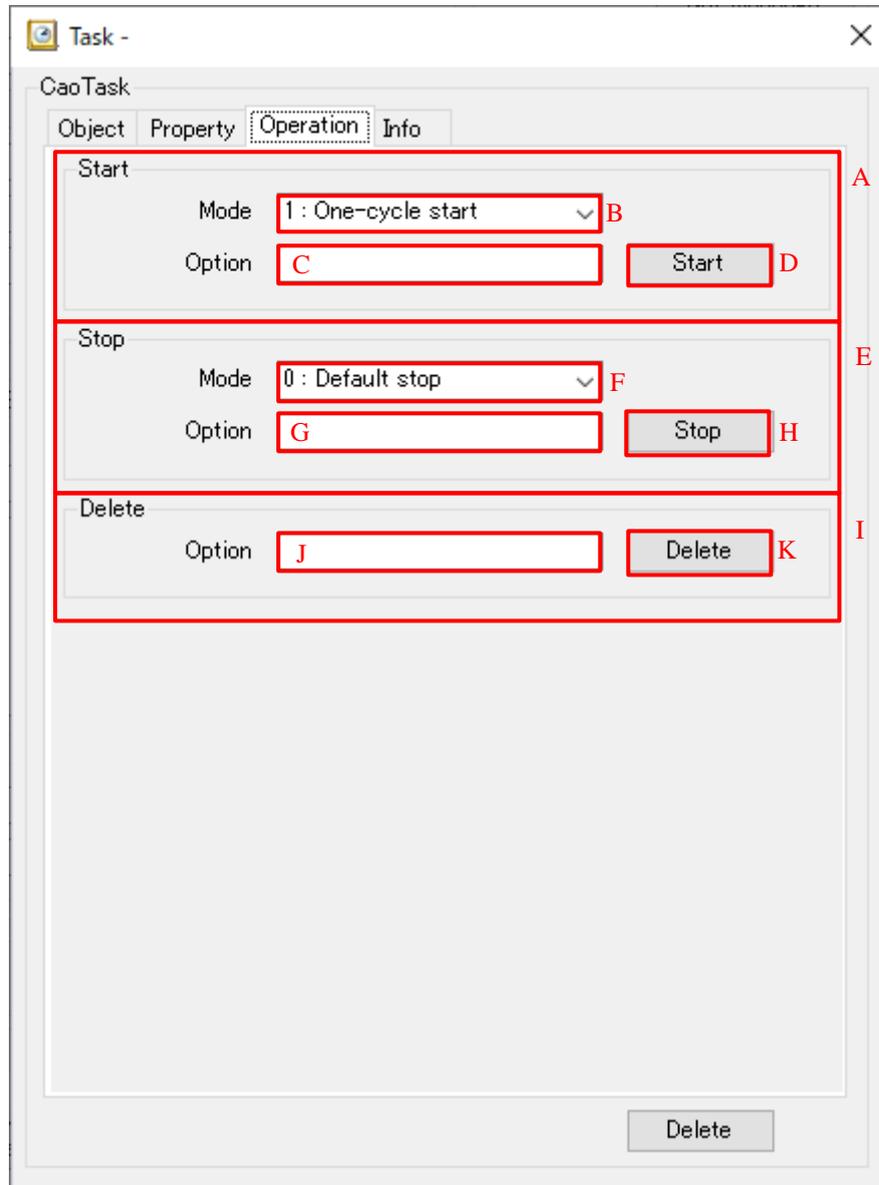


図 2-21 Task ウィンドウの Operation タブ

A:タスクの開始グループ

タスクの開始を行うグループです.

B:タスクの開始モード

D の Start 実行時の第一引数として使用します.

C:タスクの開始オプション

D の Start 実行時の第二引数として使用します.

D:タスクの開始

B, C の値を使用して Start を実行します.

E:タスクの停止グループ

タスクの停止を行うグループです.

F:タスクの停止モード

H の Stop 実行時の第一引数として使用します.

G:タスクの停止オプション

H の Stop 実行時の第二引数として使用します.

H:タスクの停止

F, G の値を使用して Stop を実行します.

I:タスクの削除グループ

タスクの削除を行うグループです.

J:タスクの削除オプション

K の Delete 実行時の第一引数として使用します.

K:タスクの停止

J の値を使用して Delete を実行します.

2.8.4. Info タブ

Info タブではそのオブジェクトを作る時の情報を確認できます. 詳細は 2.17 を参照ください.

2.9. Extension ウィンドウ

Extension ウィンドウでは変数オブジェクト作成, 拡張コマンドの実行を行うことができます。

2.9.1. Object タブ

Object タブでは変数オブジェクトの作成や拡張コマンドの実行ができます。

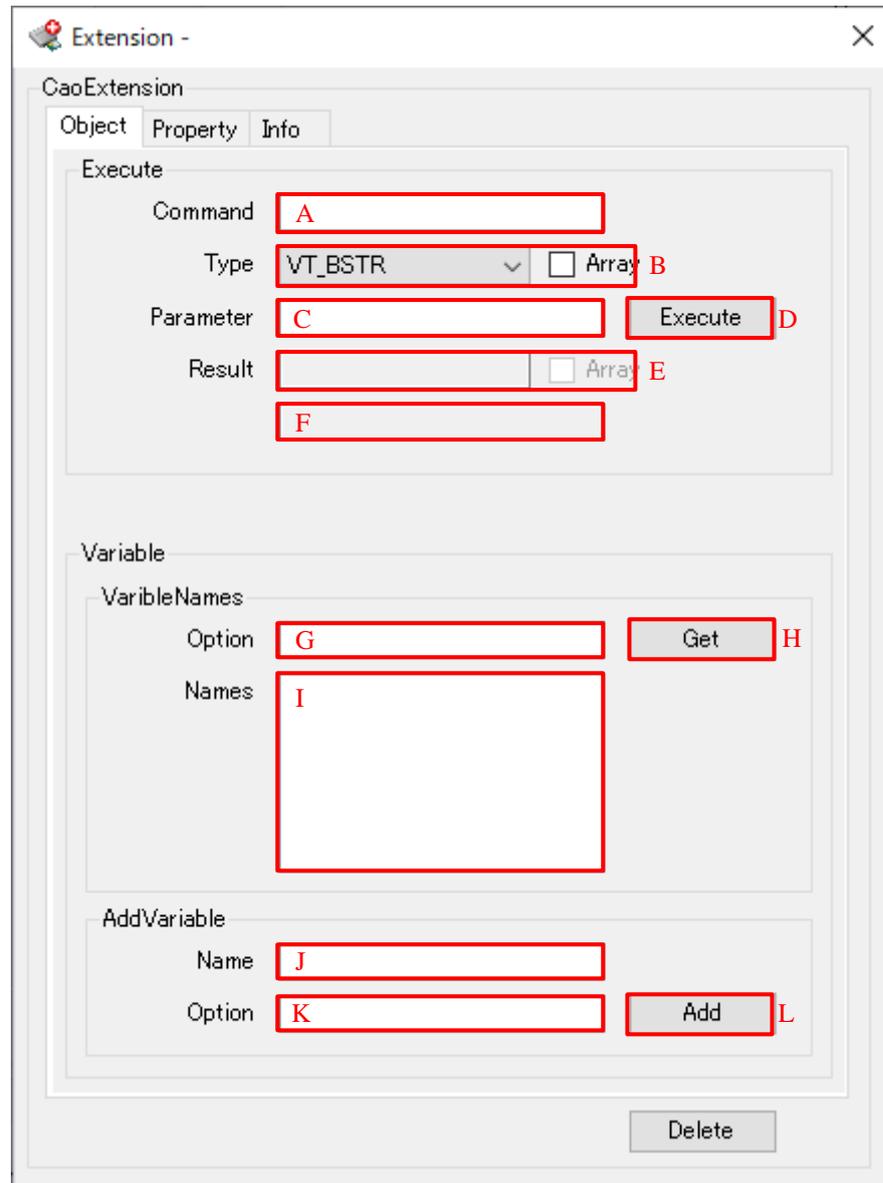


図 2-22 Extension ウィンドウの Object タブ

A:コマンド名

D の Execute 実行時のコマンド名として使用します。

B:パラメータの型情報

D の Execute 実行時のパラメータの型情報として使用します。

C:パラメータ文字列

D の Execute 実行時のパラメータ文字列として使用します。

D:Execute の実行

A, B, C の値を使用して Execute を実行します。

正常終了時の結果の型と値は E, F に格納されます。

E:Execute の実行結果の型情報

D の Execute 実行結果の型情報が格納されます。

F:Execute の実行結果

D の Execute 実行結果が格納されます。

G:VariableNames のオプション

H の VariableNames 実行時のオプションとして使用します。

H:VariableNames の実行

VariableNames を実行します。

I に実行結果が格納されます。

I:VariableNames の実行結果

H の VariableNames 実行結果が格納されます。

J:作成するオブジェクト名

L の AddVariable 実行時の変数名がして使用します。

K:作成するオブジェクトのオプション

L の AddVariable 実行時のオプションとして使用します。

L:AddVariable の実行

J, K の値を使用して AddVariable を実行してオブジェクトを作成します。

2.9.2. Property タブ

Property タブではタスクオブジェクトのプロパティ情報を設定、取得ができます。

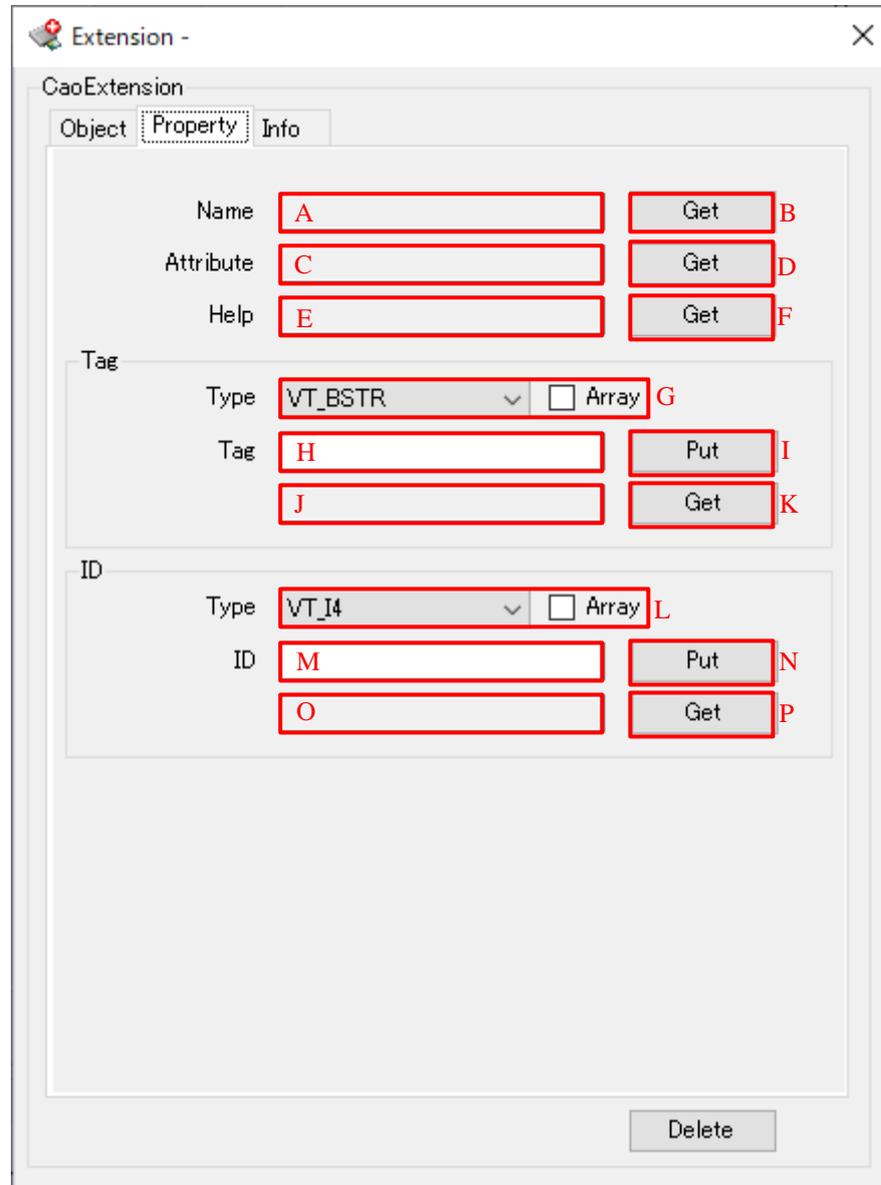


図 2-23 Extension ウィンドウの Property タブ

A: 拡張ボード名の取得の実行結果

B の拡張ボード名の取得の実行結果を格納します。

B: 拡張ボード名の取得

拡張ボード名を取得し実行結果を A に格納します。

C: 属性の取得結果

D の属性の取得の実行結果を格納します。

D:属性の取得

属性を取得し実行結果を C に格納します。

E:ヘルプの取得結果

F のヘルプの取得の実行結果を格納します。

F:ヘルプの取得

ヘルプを取得し実行結果を E に格納します。

G:タグの型情報

I の put_Tag 実行時のパラメータの型情報として使用します。

K の get_Tag 実行結果の型情報として格納されます。

H:タグデータ文字列

I の put_Tag 実行時のタグデータとして使用します。

I:タグの設定

put_Tag を実行します。

J:タグの取得結果

K の get_Tag の実行結果が格納されます。

K:タグの取得

get_Tag を実行し J に実行結果が格納されます。

型情報は I に格納されます。

L:ID の型情報

N の put_ID 実行時のパラメータの型情報として使用します。

P の get_ID 実行結果の型情報として格納されます。

M:ID 文字列

N の put_ID 実行時のタグデータとして使用します。

N:ID の設定

put_ID を実行します。

O:ID の取得結果

P の get_ID の実行結果が格納されます。

P:ID の取得

get_ID を実行し O に実行結果が格納されます。

型情報は L に格納されます。

2.9.3. Info タブ

Info タブではそのオブジェクトを作る時の情報を確認できます。詳細は 2.17 を参照ください。

2.10. Command ウィンドウ

Command ウィンドウでは拡張コマンドの実行を行うことができます。

2.10.1. Object タブ

Object タブではコマンドの実行ができます。

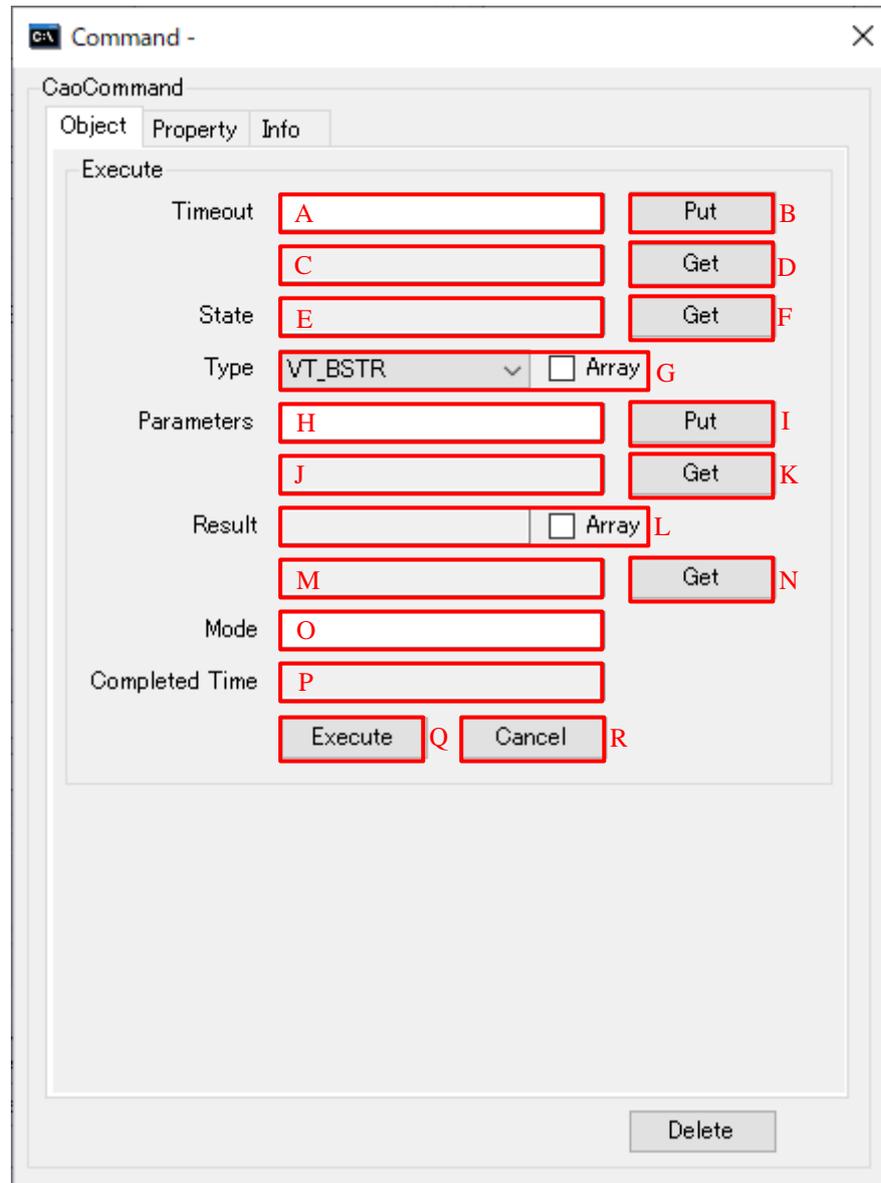


図 2-24 Command ウィンドウの Object タブ

A:タイムアウトの取得の実行結果

B のタイムアウトの取得の実行結果を格納します。

B:タイムアウトの取得

タイムアウトを取得し実行結果を A に格納します。

Q の Execute 実行前に設定をしておく必要があります。

C:タイムアウトの設定の実行結果

D のタイムアウトの設定の実行結果を格納します。

D:タイムアウトの設定

タイムアウトを設定し実行結果を C に格納します。

E:状態の取得結果

F の状態の取得の実行結果を格納します。

F:状態の取得

状態を取得し実行結果を E に格納します。

G:コマンドパラメータの型情報

I の put_Parameters 実行時のパラメータの型情報として使用します。

K の get_Parameters 実行結果の型情報として格納されます。

H:コマンドパラメータデータ文字列

I の put_Parameters 実行時のタグデータとして使用します。

I:コマンドパラメータの設定

put_Parameters を実行します。

Q の Execute 実行前に設定をしておく必要があります。

J:コマンドパラメータの取得結果

K の get_Parameters の実行結果が格納されます。

K:コマンドパラメータの取得

get_Parameter を実行し J に実行結果が格納されます。

型情報は I に格納されます。

L:Result の実行結果の型情報

N の Result 実行結果の型情報として格納されます。

M:Result の実行結果

N の Result 実行結果が格納されます。

N:Result の実行

Result を実行します。

結果の型と値は L, M に格納されます。

O:実行モード

Q の Execute 実行時の第一引数として使用します。

P:Execute 実行完了日時

N の Execute 実行完了日時が格納されます⁶⁾。

モードが同期実行(0)の場合は Execute 実行完了日時が格納されます。

モードが非同期実行(1)の場合はコマンド実行完了イベント受信日時が格納されます。

Q:Execute の実行

Execute を実行します⁶⁾。

モードが同期実行(0)の場合は Execute 実行完了日時が格納されます。

モードが非同期実行(1)の場合はコマンド実行完了イベント受信日時が格納されます。

R:Execute のキャンセル

実行中の Execute をキャンセルします。

⁶⁾ 現在、コマンド実行完了イベント未対応のため、非同期実行では Execute 実行完了日時は記録されません。

2.10.2. Property タブ

Property タブではタスクオブジェクトのプロパティ情報を設定、取得ができます。

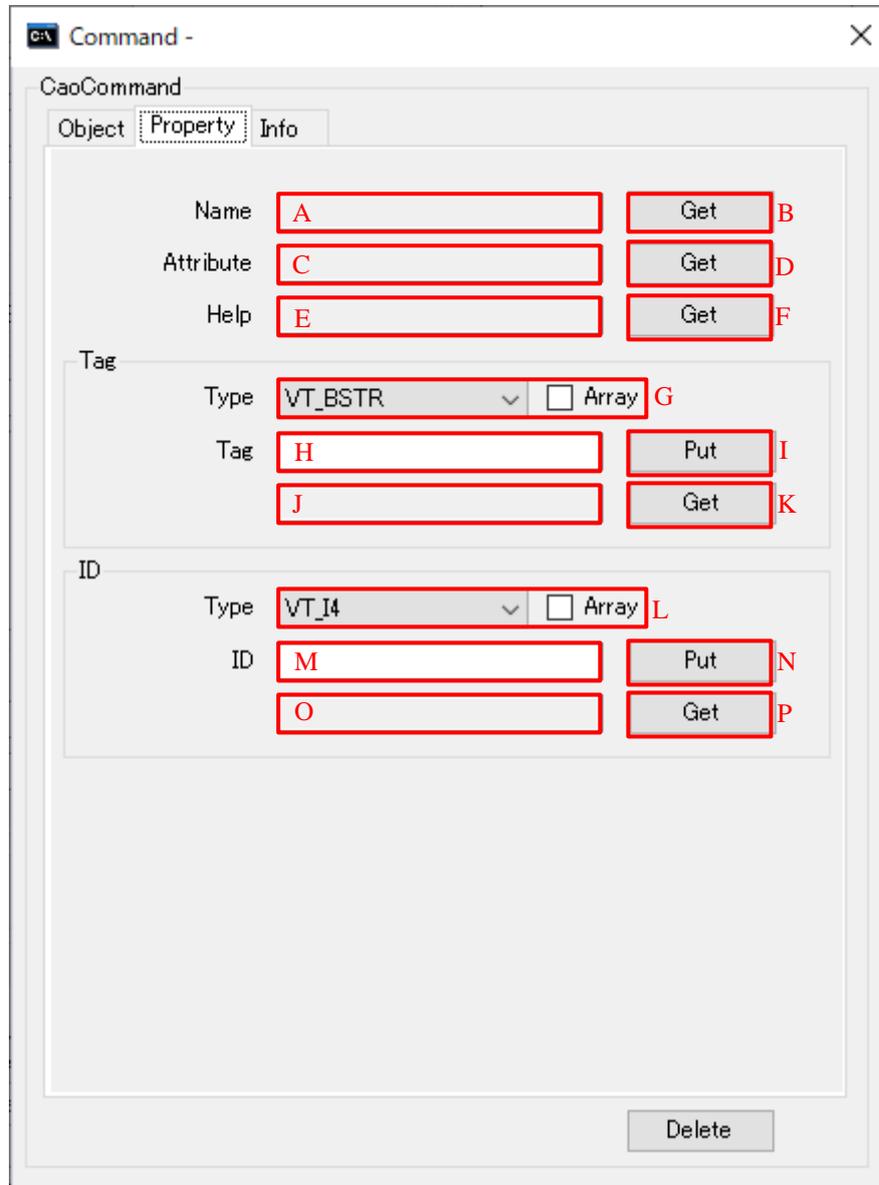


図 2-25 Command ウィンドウの Property タブ

A: コマンド名の取得の実行結果

B のコマンド名の取得の実行結果を格納します。

B: コマンド名の取得

コマンド名を取得し実行結果を A に格納します。

C: 属性の取得結果

D の属性の取得の実行結果を格納します。

D:属性の取得

属性を取得し実行結果を C に格納します。

E:ヘルプの取得結果

F のヘルプの取得の実行結果を格納します。

F:ヘルプの取得

ヘルプを取得し実行結果を E に格納します。

G:タグの型情報

I の put_Tag 実行時のパラメータの型情報として使用します。

K の get_Tag 実行結果の型情報として格納されます。

H:タグデータ文字列

I の put_Tag 実行時のタグデータとして使用します。

I:タグの設定

put_Tag を実行します。

J:タグの取得結果

K の get_Tag の実行結果が格納されます。

K:タグの取得

get_Tag を実行し J に実行結果が格納されます。

型情報は I に格納されます。

L:ID の型情報

N の put_ID 実行時のパラメータの型情報として使用します。

P の get_ID 実行結果の型情報として格納されます。

M:ID 文字列

N の put_ID 実行時のタグデータとして使用します。

N:ID の設定

put_ID を実行します。

O:ID の取得結果

P の get_ID の実行結果が格納されます。

P:ID の取得

get_ID を実行し O に実行結果が格納されます。

型情報は L に格納されます。

2.10.3. Info タブ

Info タブではそのオブジェクトを作る時の情報を確認できます。詳細は 2.17 を参照ください。

2.11. Message ウィンドウ

Message ウィンドウではメッセージの確認や返信, クリアを行うことができます。

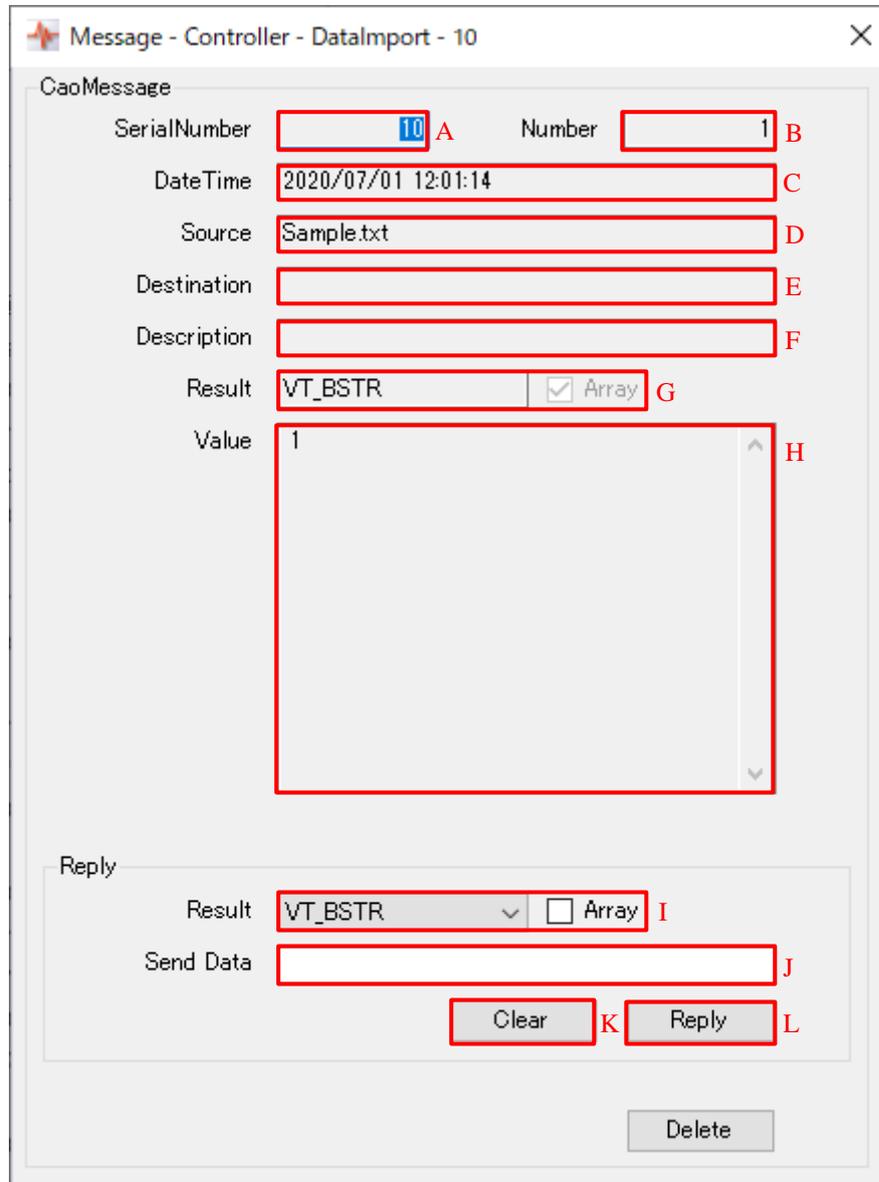


図 2-26 Message ウィンドウ

A:メッセージ連番

エンジンで自動的に付加される連番が格納されます。

B:メッセージ番号

メッセージ番号が格納されます。

C:作成日時

作成日時が格納されます。

D:送り元

送り元が格納されます。

E:送り先

送り先が格納されます。

F:説明

説明が格納されます。

G:メッセージ本文の型情報

H のメッセージ本文の型情報が格納されます。

H:メッセージ本文データ文字列

メッセージ本文のデータ文字列が格納されます。

I:メッセージの返信の型情報

メッセージの返信の型情報が格納されます。

J:メッセージの返信データ文字列

メッセージの返信データ文字列が格納されます。

K:メッセージのクリア

メッセージのクリアを実行します。

L:メッセージの返信

I, J の値を使用してメッセージの返信を実行します。

2.12. Engine Status ウィンドウ

Engine Status ウィンドウでは一定間隔でオブジェクト数の確認を行うことができます。

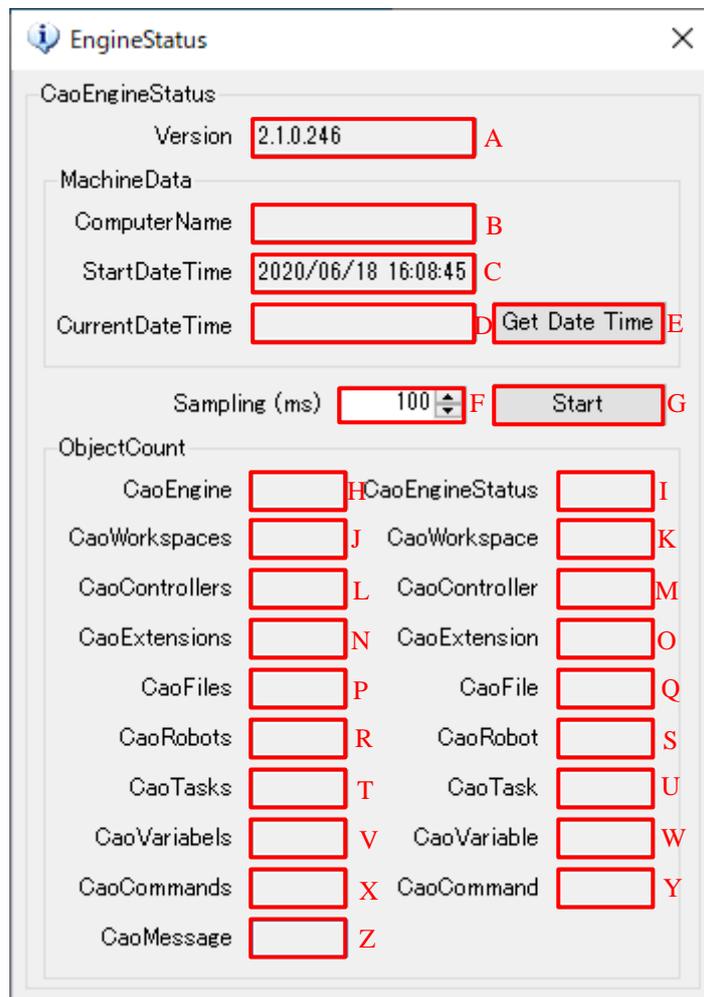


図 2-27 EngineStatus ウィンドウ

A:エンジンのバージョン

エンジンのバージョンが格納されます。

B:コンピュータ名

コンピュータ名が格納されます。

C:エンジンの開始日時

エンジンの開始日時が格納されます。

D:現在の日時

E の現在の日時の取得結果が格納されます。

E:現在の日時の取得

現在の日時を取得し D に格納します。

F:オブジェクトカウント取得のサンプリング間隔

G のオブジェクトカウント取得の開始のサンプリング間隔として使用します。

G:オブジェクトカウント取得の開始 / 終了

F のサンプリング間隔でオブジェクトカウントの取得を開始 / 終了します。

H:CaoEngine のオブジェクトカウント

サンプリング間隔で CaoEngine のオブジェクトカウントを取得し格納します。

I:CaoEngineStatus のオブジェクトカウント

サンプリング間隔で CaoEngineStatus のオブジェクトカウントを取得し格納します。

J:CaoWorkspaces のオブジェクトカウント

サンプリング間隔で CaoWorkspaces のオブジェクトカウントを取得し格納します。

K:CaoWorkspace のオブジェクトカウント

サンプリング間隔で CaoWorkspace のオブジェクトカウントを取得し格納します。

L:CaoControllers のオブジェクトカウント

サンプリング間隔で CaoControllers のオブジェクトカウントを取得し格納します。

M:CaoController のオブジェクトカウント

サンプリング間隔で CaoController のオブジェクトカウントを取得し格納します。

N:CaoExtensions のオブジェクトカウント

サンプリング間隔で CaoExtensions のオブジェクトカウントを取得し格納します。

O:CaoExtension のオブジェクトカウント

サンプリング間隔で CaoExtension のオブジェクトカウントを取得し格納します。

P:CaoFiles のオブジェクトカウント

サンプリング間隔で CaoFiles のオブジェクトカウントを取得し格納します。

Q:CaoFile のオブジェクトカウント

サンプリング間隔で CaoFile のオブジェクトカウントを取得し格納します。

R:CaoRobots のオブジェクトカウント

サンプリング間隔で CaoRobots のオブジェクトカウントを取得し格納します。

S:CaoRobot のオブジェクトカウント

サンプリング間隔で CaoRobot のオブジェクトカウントを取得し格納します。

T:CaoTasks のオブジェクトカウント

サンプリング間隔で CaoTasks のオブジェクトカウントを取得し格納します。

U:CaoTask のオブジェクトカウント

サンプリング間隔で CaoTask のオブジェクトカウントを取得し格納します。

V:CaoVariables のオブジェクトカウント

サンプリング間隔で CaoVariables のオブジェクトカウントを取得し格納します。

W:CaoVariable のオブジェクトカウント

サンプリング間隔で CaoVariable のオブジェクトカウントを取得し格納します。

X:CaoCommands のオブジェクトカウント

サンプリング間隔で CaoCommands のオブジェクトカウントを取得し格納します。

Y:CaoCommand のオブジェクトカウント

サンプリング間隔で CaoCommand のオブジェクトカウントを取得し格納します。

Z:CaoMessage のオブジェクトカウント

サンプリング間隔で CaoMessage のオブジェクトカウントを取得し格納します。

2.13. Engine ウィンドウ

Engine ウィンドウではワークスペースオブジェクトの作成を行うことができます。

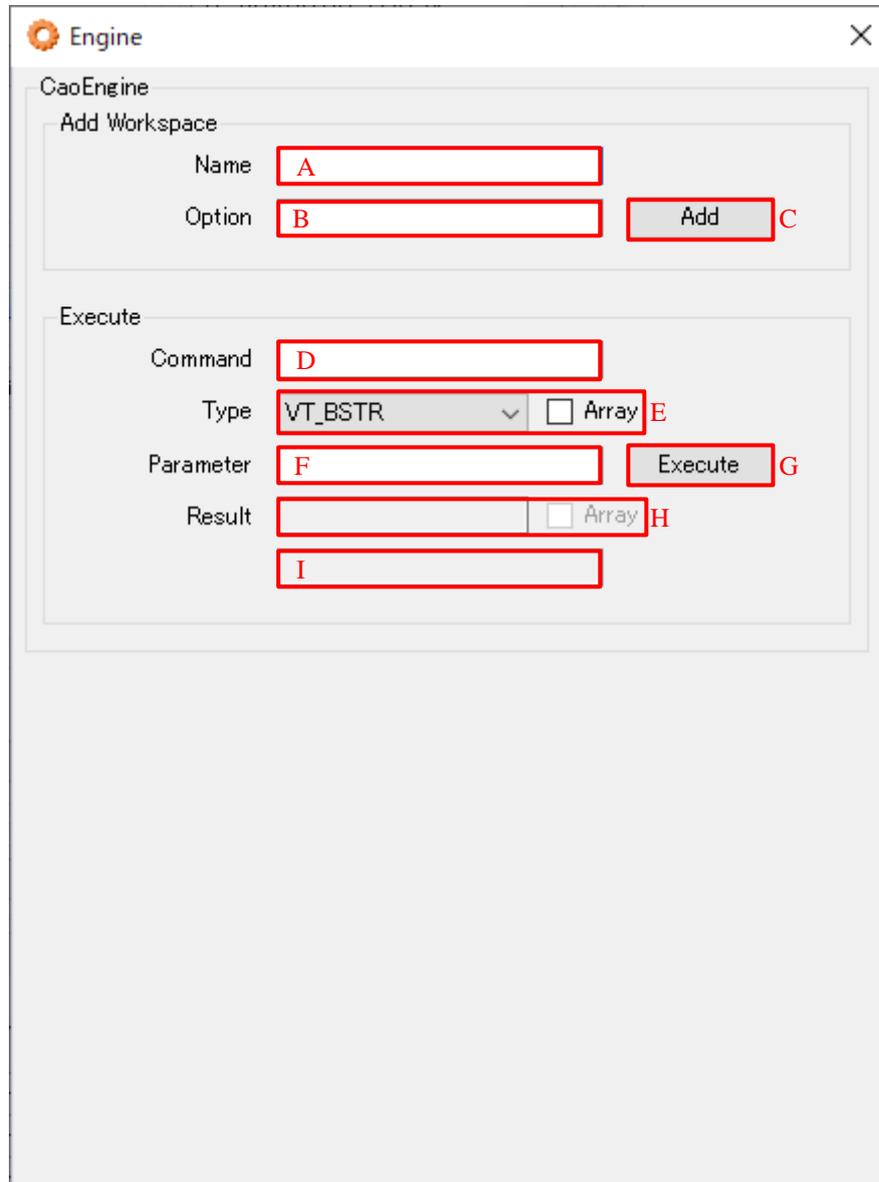


図 2-28 Engine ウィンドウ

A:作成するワークスペースオブジェクト

C の AddWorkspace 実行時の変数名として使用します。

B:作成するワークスペースオブジェクトのオプション

C の AddWorkspace 実行時のオプションとして使用します。

C:AddWorkspace の実行

A, B の値を使用して AddWorkspace を実行してワークスペースオブジェクトを作成します。

D:コマンド名

G の Execute 実行時のコマンド名として使用します。

E:パラメータの型情報

G の Execute 実行時のパラメータの型情報として使用します。

F:パラメータ文字列

G の Execute 実行時のパラメータ文字列として使用します。

G:Execute の実行

D, E, F の値を使用して Execute を実行します。

正常終了時の結果の型と値は H, I に格納されます。

H:Execute の実行結果の型情報

G の Execute 実行結果の型情報が格納されます。

I:Execute の実行結果

G の Execute 実行結果が格納されます。

2.14. オブジェクトウィンドウ

作成, 削除したオブジェクトがツリー状態で表示されます。

オブジェクトをダブルクリックまたは[右クリック] - [Show]をクリックすることで, 対象のオブジェクトウィンドウをアクティブにすることができます。

[右クリック] - [Delete]をクリックすることで, 対象のオブジェクトとオブジェクトウィンドウを削除します。

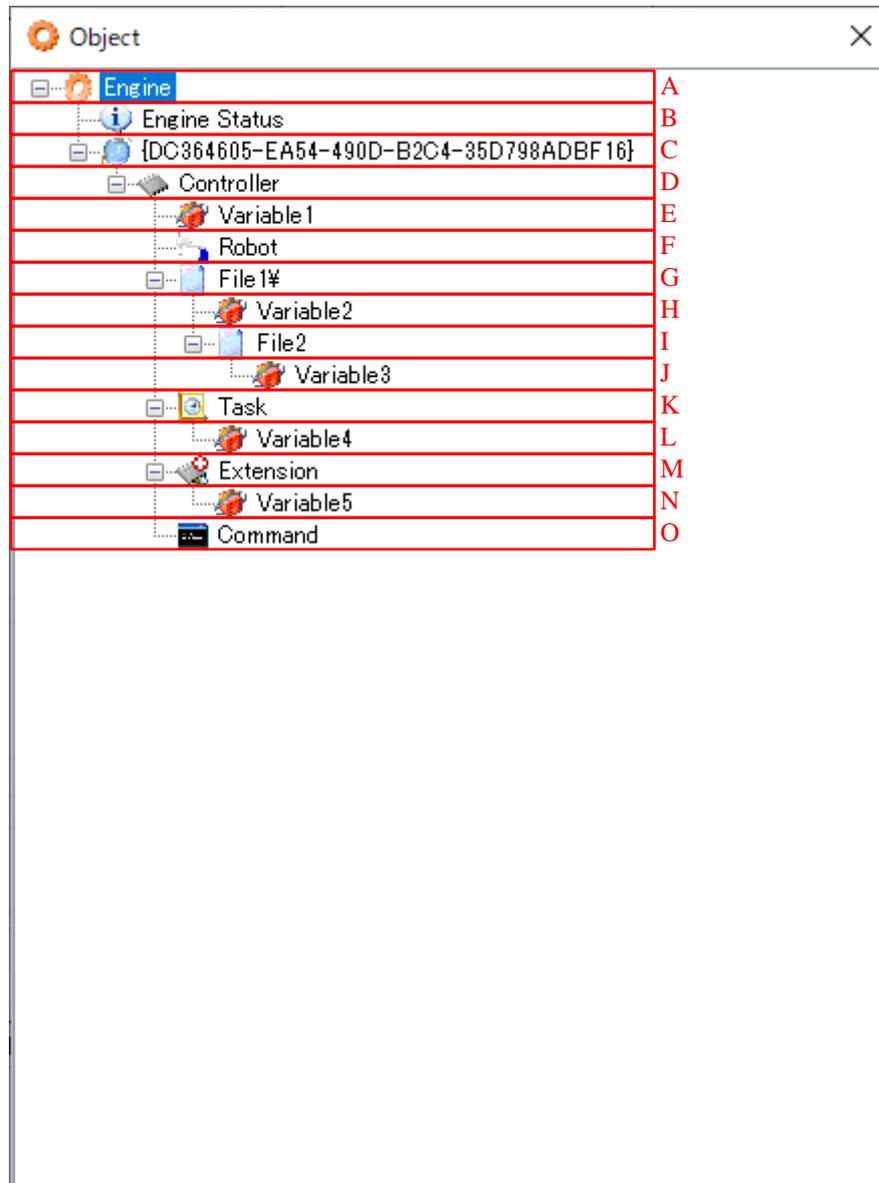


図 2-29 オブジェクトウィンドウ

A:CaoEngine オブジェクト

CaoEngine オブジェクトです。

B:CaoEngineStatus オブジェクト

CaoEngineStatus オブジェクトです。

C:CaoWorkspace オブジェクト

CaoWorkspace オブジェクトです。

D:CaoController オブジェクト

CaoController オブジェクトです。

E:CaoVariable オブジェクト

CaoController から生成された CaoVariable オブジェクトです。

F:CaoRobot オブジェクト

CaoController から生成された CaoRobot オブジェクトです。

G:CaoFile オブジェクト

CaoController から生成された CaoFile オブジェクトです。

H:CaoVariable オブジェクト

CaoFile から生成された CaoVariable オブジェクトです。

I:CaoFile オブジェクト

CaoFile から生成された CaoFile オブジェクトです。

J:CaoVariable オブジェクト

CaoFile から生成された CaoVariable オブジェクトです。

K:CaoTask オブジェクト

CaoController から生成された CaoTask オブジェクトです。

L:CaoVariable オブジェクト

CaoTask から生成された CaoVariable オブジェクトです。

M:CaoExtension オブジェクト

CaoController から生成された CaoExtension オブジェクトです。

N:CaoVariable オブジェクト

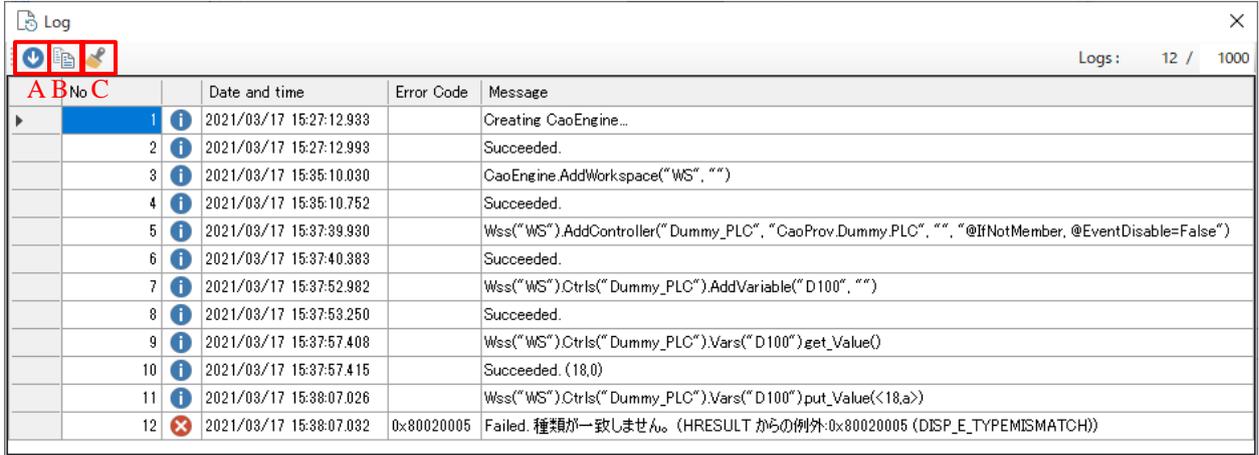
CaoExtension から生成された CaoVariable オブジェクトです。

O:CaoCommand オブジェクト

CaoController から生成された CaoCommand オブジェクトです.

2.15. ログウィンドウ

行った操作やその結果がログとして記録されます。



No	Date and time	Error Code	Message
1	2021/03/17 15:27:12.933		Creating CaoEngine...
2	2021/03/17 15:27:12.993		Succeeded.
3	2021/03/17 15:35:10.030		CaoEngine.AddWorkspace("WS", "")
4	2021/03/17 15:35:10.752		Succeeded.
5	2021/03/17 15:37:39.930		Wss("WS").AddController("Dummy_PLC", "CaoProv.Dummy.PLC", "", "@IfNotMember, @EventDisable=False")
6	2021/03/17 15:37:40.383		Succeeded.
7	2021/03/17 15:37:52.982		Wss("WS").Ctrls("Dummy_PLC").AddVariable("D100", "")
8	2021/03/17 15:37:53.250		Succeeded.
9	2021/03/17 15:37:57.408		Wss("WS").Ctrls("Dummy_PLC").Vars("D100").get_Value()
10	2021/03/17 15:37:57.415		Succeeded. (18,0)
11	2021/03/17 15:38:07.026		Wss("WS").Ctrls("Dummy_PLC").Vars("D100").put_Value(<18,a>)
12	2021/03/17 15:38:07.032	0x80020005	Failed. 種類が一致しません。(HRESULT からの例外:0x80020005 (DISP_E_TYPEREMISMATCH))

図 2-30 ログウィンドウ

A:自動スクロール

ログ追記時の自動スクロールを有効 / 無効にします。

B:コピー

選択されている項目をクリップボードにコピーします。

C:クリア

表示されているログをすべてクリアします。

2.16. Image ウィンドウ

Image ウィンドウではサムネイル表示された画像を等倍サイズで確認することができます。

サムネイルされた画像は File ウィンドウの Value タブの get_Value の結果、Variable ウィンドウの get_Value の結果等に表示されます(図 2-31, 図 2-32)。

サムネイルされた画像をクリックすることで Image ウィンドウを表示することができます(図 2-33)。

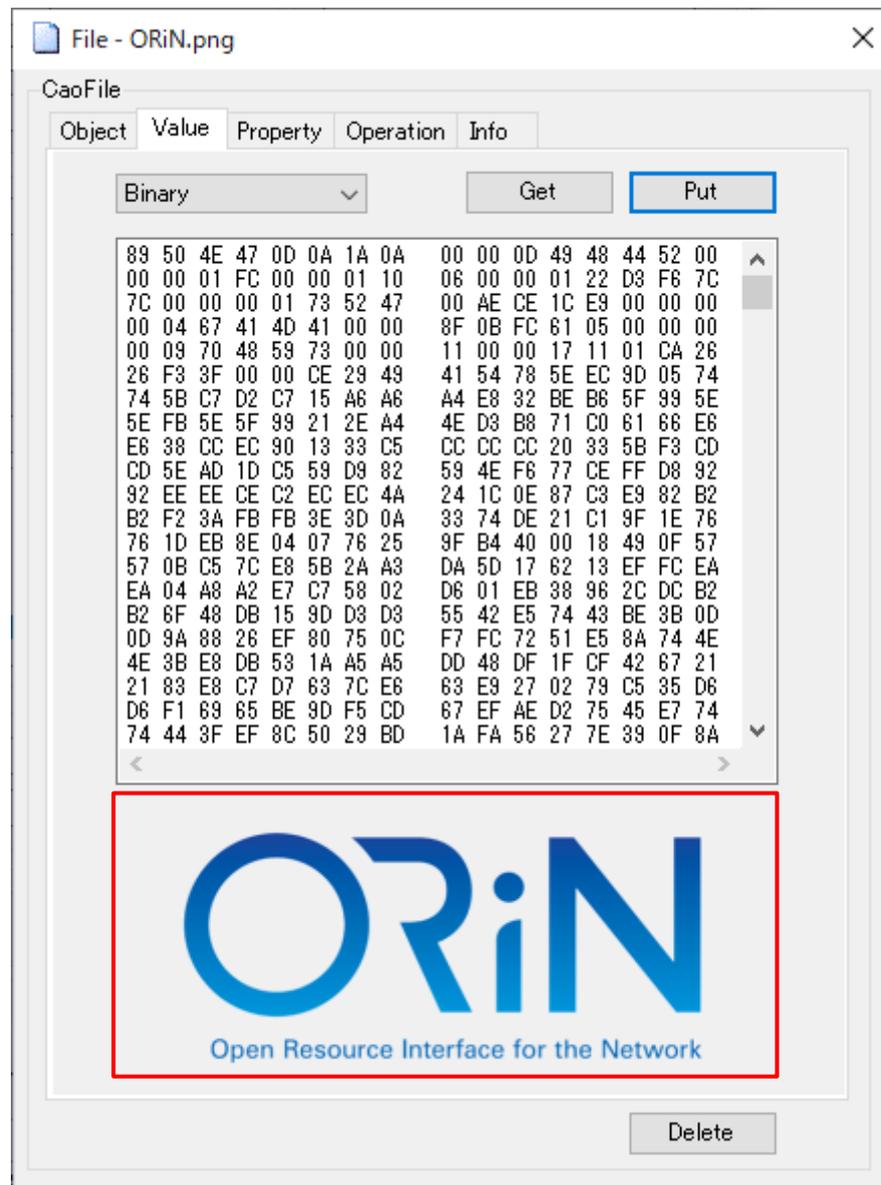


図 2-31 File ウィンドウの Value プロパティ

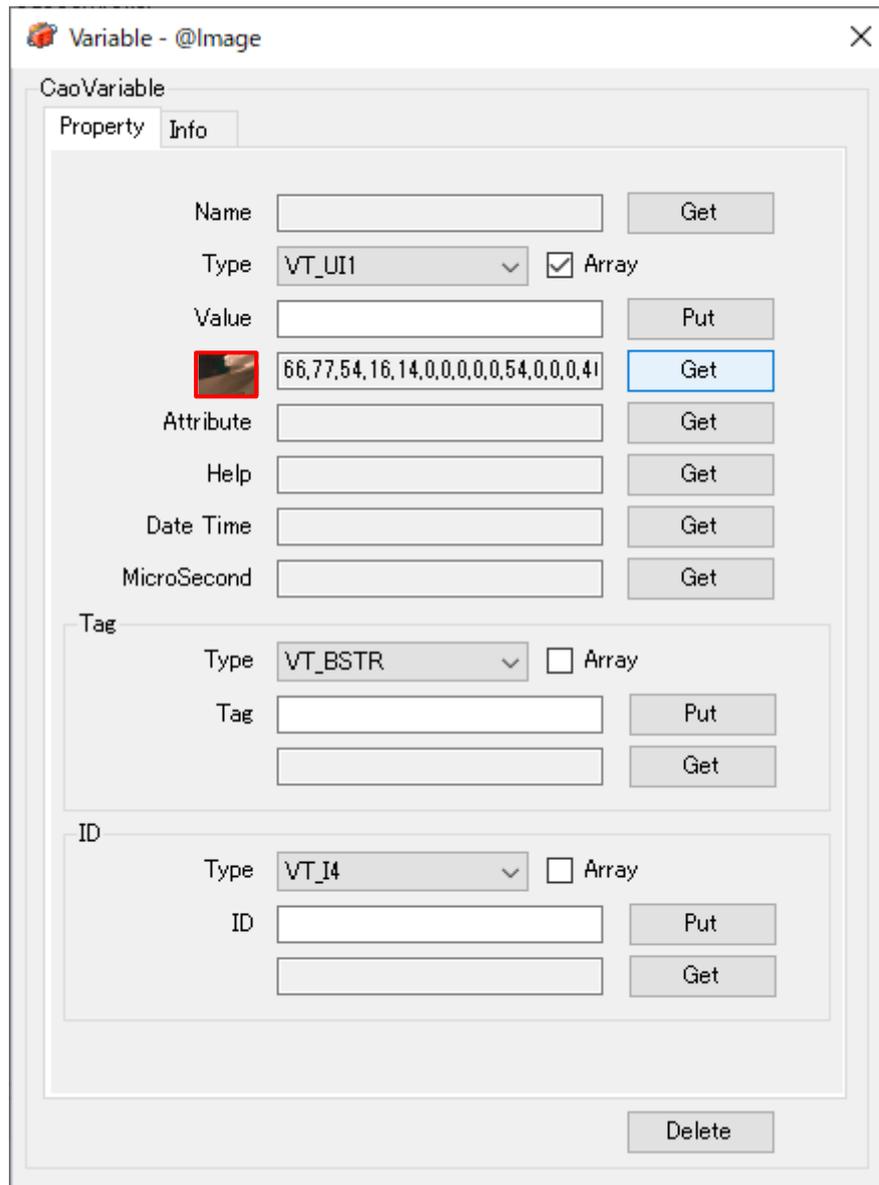


図 2-32 Variable ウィンドウの Property タブ



図 2-33 Image ウィンドウ

2.17. Info タブ

Info タブではそのオブジェクトを作る時の情報を確認することができます。

図 2-34 では変数名に“Variable”, オプションに“Test=1”を指定して作成されたことが確認できます。

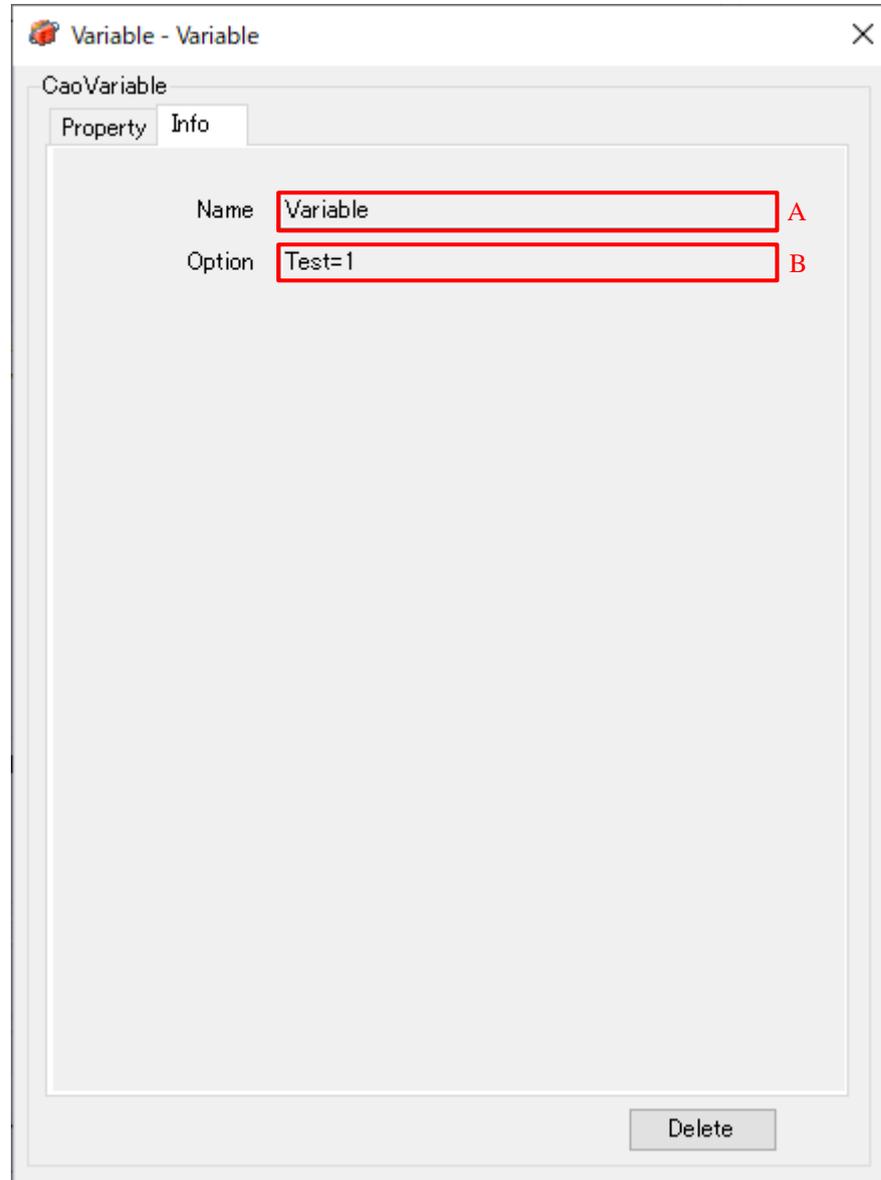


図 2-34 Info タブ

A:オブジェクト名

作成したオブジェクトのオブジェクト名が格納されます。

B:オプション名

作成したオブジェクトのオプション名が格納されます。

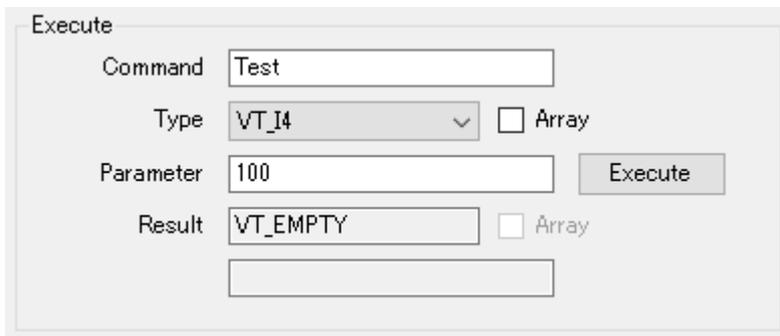
3. 付録

付録A. 書き込み値, 型情報の指定方法

CAO の拡張メソッド(Execute)の実行等については引数として数値や文字列, 数値配列, RAC 文字列等が指定できます. CaoTester2 での設定方法を拡張メソッドで実行する形で説明していきます. CaoTester2 で使用できるデータ型については付録 B を参照してください.

[数値を指定]

Execute の“Test”コマンドに VT_I4 の 100 を指定して実行した結果を以下に示します.



Execute

Command

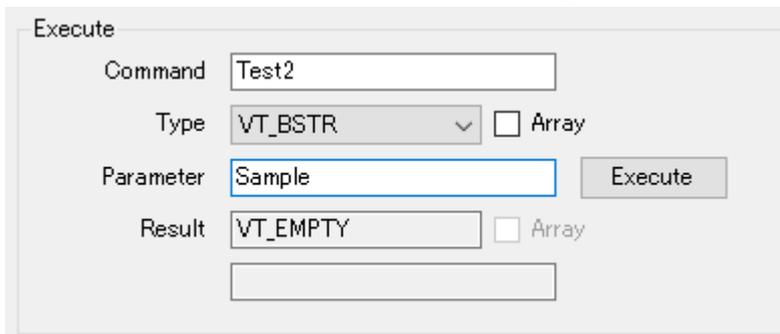
Type Array

Parameter

Result Array

[文字列を指定]

Execute の“Test2”コマンドに VT_BSTR の“Sample”を指定して実行した結果を以下に示します.



Execute

Command

Type Array

Parameter

Result Array

[数値配列を指定]

配列を指定する場合は, Array のチェックボックスにチェックを入れます.

Execute の“Test3”コマンドに(VT_I4 | VT_ARRAY)の(100, 200)を指定して実行した結果を以下に示します.

The screenshot shows the 'Execute' dialog box for 'Test3'. The 'Command' field contains 'Test3'. The 'Type' dropdown is set to 'VT_I4' and the 'Array' checkbox is checked. The 'Parameter' field contains '100,200'. The 'Result' dropdown is set to 'VT_I4' and the 'Array' checkbox is unchecked. The result field displays '43'.

[RAC 文字列を指定]

単純な型を指定する際にも RAC 文字列を使用することができます。

Execute の“Test3”コマンドに(VT_VARIANT)の(8,Sample)を指定して実行した結果を以下に示します。

The screenshot shows the 'Execute' dialog box for 'Test2'. The 'Command' field contains 'Test2'. The 'Type' dropdown is set to 'VT_VARIANT' and the 'Array' checkbox is checked. The 'Parameter' field contains '(8,Sample)'. The 'Result' dropdown is set to 'VT_EMPTY' and the 'Array' checkbox is unchecked. The result field is empty.

[複雑な RAC 文字列を指定]

Execute の“Test4”コマンドに(VT_VARIANT | VT_ARRAY)の(3,1),(3,0),(8,abc),(8,123456)を指定して実行した結果を以下に示します。

The screenshot shows the 'Execute' dialog box for 'Test4'. The 'Command' field contains 'Test4'. The 'Type' dropdown is set to 'VT_VARIANT' and the 'Array' checkbox is checked. The 'Parameter' field contains '(3,1),(3,0),(8,abc),(8,123456)'. The 'Result' dropdown is set to 'VT_BSTR' and the 'Array' checkbox is unchecked. The result field displays 'Ver. 1.04'.

付録B. 使用できるデータ型

CaoTester2 で使用できるデータ型一覧を以下に示します。

RAC 文字列の指定方法の詳細については「RAC ユーザーズガイド」の「3.2.データの記述方式」を参照してください。

データ型	値	意味
VT_EMPTY	0	値が未指定
VT_NULL	1	NULL
VT_BOOL	11	ブール型
VT_I1	16	符号付き 1 バイト整数型
VT_UI1	17	符号無し 1 バイト整数型
VT_I2	2	符号付き 2 バイト整数型
VT_UI2	18	符号無し 2 バイト整数型
VT_I4	3	符号付き 4 バイト整数型
VT_UI4	19	符号無し 4 バイト整数型
VT_I8	20	符号付き 8 バイト整数型
VT_UI8	21	符号無し 8 バイト整数型
VT_R4	4	単精度浮動小数点数型
VT_R8	5	倍精度浮動小数点数型
VT_BSTR	8	文字列型
VT_DATE	7	日付型
VT_CY	6	通貨型
VT_VARIANT	12	VARIANT 型 ⁽⁷⁾
VT_ARRAY	8192	配列型 ⁽⁸⁾

⁷ VARIANT 型を使用する際は必ず VT_ARRAY と合わせて使用する必要があります。

⁸ VT_EMPTY や VT_NULL の配列指定できない型には VT_ARRAY を使用した場合、配列は無視されます。

付録C. License

付録C.1. DockPanel Suite

The MIT License

Copyright (c) 2007-2012 Weifen Luo (email: weifenluo@yahoo.com) and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.