

MagicRPA 利用マニュアル

Ver. 0.3 rev.1

2019 年 04 月 21 日

neil and Company, Ltd.

改版履歷

[illegible]

目次

MagicRPA 利用マニュアル	1
改版履歴	2
目次	3
1. はじめに	7
2. 機能	7
3. 利用ガイド	8
3.1. プログラムのインストール方法	8
3.2. プログラムのアンインストール方法	12
3.3. MagicRPA の起動	13
3.4. MagicRPA エディタの起動	14
3.5. RPA プログラムの保存	15
3.6. MagicRPA エディタの終了	16
3.7. RPA プログラムの実行	16
3.8. RPA プログラムの終了	18
3.9. RPA プログラムの停止（強制停止）	18
3.10. MagicRPA の終了	19
3.11. コマンドライン実行	19
4. RPA プログラム作成ガイド	20
4.1. 処理の流れ	20
4.2. コマンド一覧	20
4.2.1. INI ファイルの制御	20
4.2.2. DELETEINI コマンド	20
4.2.2.1. READINI コマンド	22
4.2.2.2. WRITEINI コマンド	22
4.2.3. ウィンドウの制御	23
4.2.3.1. ACW コマンド	23
4.2.3.2. GETALLWIN コマンド	24
4.2.3.3. GETHND コマンド	24
4.2.3.4. GETID コマンド	25
4.2.4. 外部プログラムの実行	25
4.2.4.1. CMDEXEC コマンド	25
4.2.4.2. DOSCMD コマンド	26
4.2.4.3. POWERSHELL コマンド	26
4.2.5. 画像の制御	27
4.2.5.1. CHKIMG コマンド	27
4.2.5.2. PEEKCOLOR コマンド	28
4.2.5.3. SEARCHIMG コマンド	28
4.2.6. クリップボードの操作	30

4.2.6.1. CLIPMGT コマンド	30
4.2.7. コメントの挿入	31
4.2.7.1. REM コマンド	31
4.2.8. 条件処理	31
4.2.8.1. IF コマンド	31
4.2.8.2. ELSE コマンド	31
4.2.8.3. FI コマンド	31
4.2.8.4. TRY コマンド	32
4.2.8.5. EXCEPT コマンド	32
4.2.8.6. ENDTRY コマンド	32
4.2.9. 数学関数	32
4.2.9.1. ARCCOS コマンド	32
4.2.9.2. ARCSIN コマンド	33
4.2.9.3. ARCTAN コマンド	33
4.2.9.4. CEIL コマンド	33
4.2.9.5. CHKNUM コマンド	33
4.2.9.6. COS コマンド	33
4.2.9.7. EXP コマンド	34
4.2.9.8. INT コマンド	34
4.2.9.9. LET コマンド	34
4.2.9.10. LOG コマンド	35
4.2.9.11. LOG10 コマンド	35
4.2.9.12. POWER コマンド	35
4.2.9.13. RANDOM コマンド	36
4.2.9.14. ROUND コマンド	36
4.2.9.15. SIN コマンド	36
4.2.9.16. SQRT コマンド	36
4.2.9.17. TAN コマンド	36
4.2.9.18. ZCUT コマンド	37
4.2.10. スキップ	37
4.2.10.1. BREAK コマンド	37
4.2.11. スリープ	37
4.2.11.1. SLEEP コマンド	37
4.2.12. 配列の制御	38
4.2.12.1. CALARRAY コマンド	38
4.2.12.2. JOIN コマンド	39
4.2.13. ファイルの制御	40
4.2.13.1. GETDIR コマンド	40
4.2.14. 変数/定数の操作	40
4.2.14.1. CONST コマンド	40

4.2.14.2. DIM コマンド	41
4.2.14.3. PUBLIC コマンド	41
4.2.14.4. VARTYPE コマンド	42
4.2.15. プログラムの終了	42
4.2.15.1. QUIT コマンド	42
4.2.16. ボタンの制御	43
4.2.16.1. CHKBTN コマンド	43
4.2.17. マウス操作の制御	43
4.2.17.1. BTN コマンド	43
4.2.17.2. MMV コマンド	44
4.2.17.3. MDG コマンド	44
4.2.18. キーボードの制御	45
4.2.18.1. KBD コマンド	45
4.2.19. メッセージボックスの制御	45
4.2.19.1. MSGBOX コマンド	45
4.2.20. 文字列の制御	46
4.2.20.1. ASCII コマンド	46
4.2.20.2. BETWEEN コマンド	46
4.2.20.3. CHAR コマンド	46
4.2.20.4. COPY コマンド	47
4.2.20.5. COPYB コマンド	47
4.2.20.6. LENGTH コマンド	47
4.2.20.7. LENGTHB コマンド	48
4.2.20.8. LEFT コマンド	48
4.2.20.9. MID コマンド	48
4.2.20.10. RIGHT コマンド	49
4.2.20.11. REPLACE コマンド	49
4.2.20.12. SPLIT コマンド	49
4.2.20.13. STRCONV コマンド	50
4.2.20.14. TRIM コマンド	50
4.2.20.15. VALUE コマンド	50
4.2.21. ループ処理	51
4.2.21.1. FOR コマンド	51
4.2.21.2. NEXT コマンド	51
4.2.21.3. REPEAT コマンド	51
4.2.21.4. UNTIL コマンド	52
4.2.21.5. WHILE コマンド	52
4.2.21.6. WEND コマンド	52
4.2.21.7. CONT コマンド	52
4.2.22. その他	53

4.2.22.1.	ABS コマンド	53
4.2.22.2.	CPUUSAGE コマンド	53
4.2.22.3.	KINDOS コマンド	53
4.2.22.4.	MEMUSAGE コマンド	54
4.3.	プログラムにコメントをする	54
4.4.	特殊処理の特徴	55
4.4.1.	DIM コマンド	55
4.4.2.	PUBLIC コマンド	55
4.4.3.	LET コマンド	55
4.4.4.	REM コマンド	55
4.4.5.	CONST コマンド	55
4.4.6.	TRY-ENDTRY コマンド	56
4.4.7.	IF-FI コマンド	56
4.4.8.	FOR-NEXT コマンド	57
4.4.9.	WHILE-WEND コマンド	57
4.4.10.	REPEAT-UNTIL コマンド	57
4.4.11.	BREAK コマンド	58
4.4.12.	CONT コマンド	58
4.4.13.	QUIT コマンド	58
5.	付録	59
5.1.	キーコード一覧	59
6.	商標に関する表記	62

1. はじめに

MagicRPA をご利用いただきありがとうございます。

MagicRPA は、「RPA- Robotic Process Automation」、すなわち人間が行う業務をロボットが代行し、効率化・自動化する取り組みを可能にするだけでなく、プログラムが書けない方でも簡単に設定できる、まるで”魔法”のようなソフトウェアです。

特に本ソフトウェアは画面制御を主にしたものであり、既存のマクロやプログラムとの親和性を重要視して作成しており、MagicRPA のプログラムから外部のプログラムやマクロを呼び出せる事に加え、逆に外部プログラムから MagicRPA プログラムの呼び出しも行えるようになっています。

2. 機能

MagicRPA では、下記の機能を利用できます。

- MagicRPA から外部プログラムやマクロを呼び出せます
- 外部プログラムやマクロから MagicRPA プログラムを実行することができます
- タスクスケジューラとコマンド実行を用いて、時間を指定しての実行が可能です
- 実際の手動操作を記録して保存をすることで、そのまま利用できます (v2.0 以降)

3. 利用ガイド

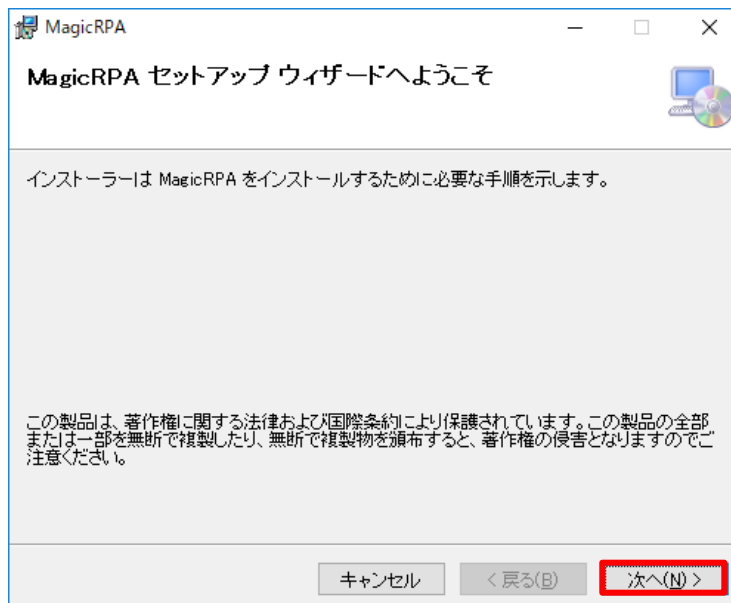
3.1. プログラムのインストール方法

MagicRPA のインストール手順です。

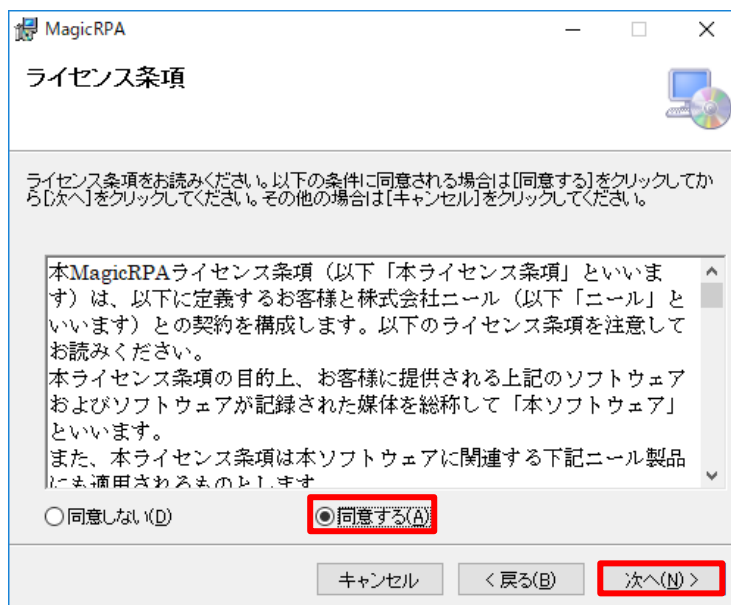
MagicRPA のインストールは、ダウンロードファイルに含まれているセットアッププログラムで行います。インストール中に再起動が必要な場合がありますので、必ず他の実行中のプログラムを閉じてからインストール作業を開始してください。

また、プログラムインストールにあたり、事前に Microsoft .NET Framework 4.6.1 のインストールが必要です。

1. ダウンロードしたファイルの中から「MagicRPA.msi」をダブルクリックします。
2. MagicRPA セットアップウィザードが表示されるので、「次へ(N)」をクリックします。



3. ライセンス条項が表示されるので、「同意する(A)」にチェックを入れ、「次へ(N)」をクリックします。



ライセンス条項

ライセンス条項をお読みください。以下の条件に同意される場合は[同意する]をクリックしてから[次へ]をクリックしてください。その他の場合は[キャンセル]をクリックしてください。

本MagicRPAライセンス条項（以下「本ライセンス条項」といいます）は、以下に定義するお客様と株式会社ニール（以下「ニール」といいます）との契約を構成します。以下のライセンス条項を注意してお読みください。

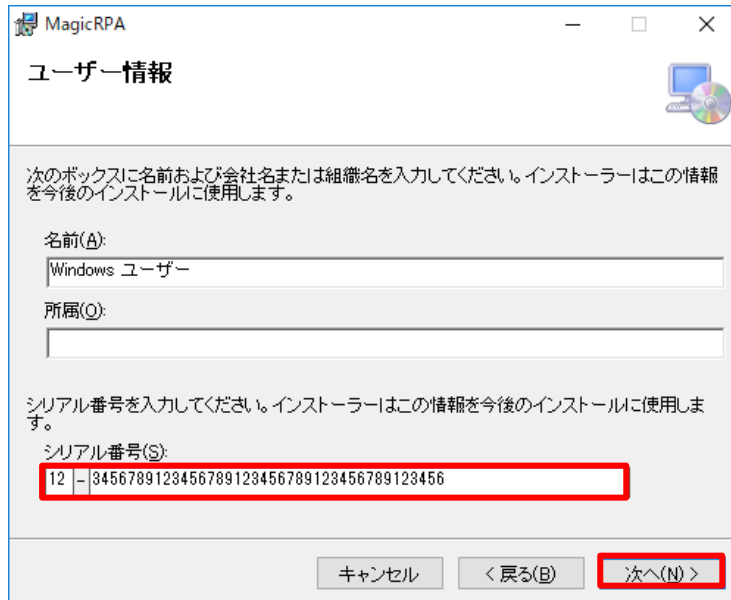
本ライセンス条項の目的上、お客様に提供される上記のソフトウェアおよびソフトウェアが記録された媒体を総称して「本ソフトウェア」といいます。

また、本ライセンス条項は本ソフトウェアに関連する下記ニール製品にも適用されるものとします。

☐ 同意しない(N) ☒ 同意する(A)

キャンセル < 戻る(B) 次へ(N) >

4. ユーザー情報が表示されるので、シリアル番号を記入し、「次へ(N)」をクリックします。
- ※非商用（商用時の導入評価を含む）の場合はシリアル番号の入力は不要です。
- 商用利用の場合のみ入力してください。



ユーザー情報

次のボックスに名前および会社名または組織名を入力してください。インストーラーはこの情報を今後のインストールに使用します。

名前(A):
Windows ユーザー

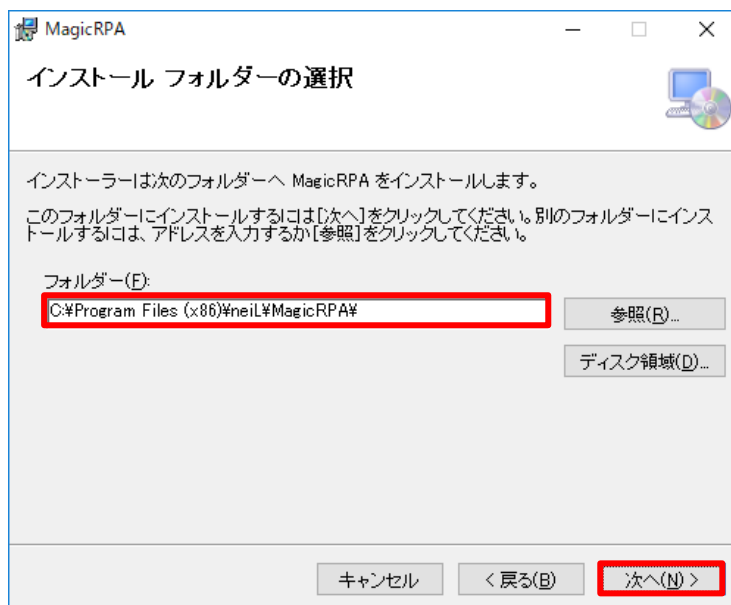
所属(O):

シリアル番号を入力してください。インストーラーはこの情報を今後のインストールに使用します。

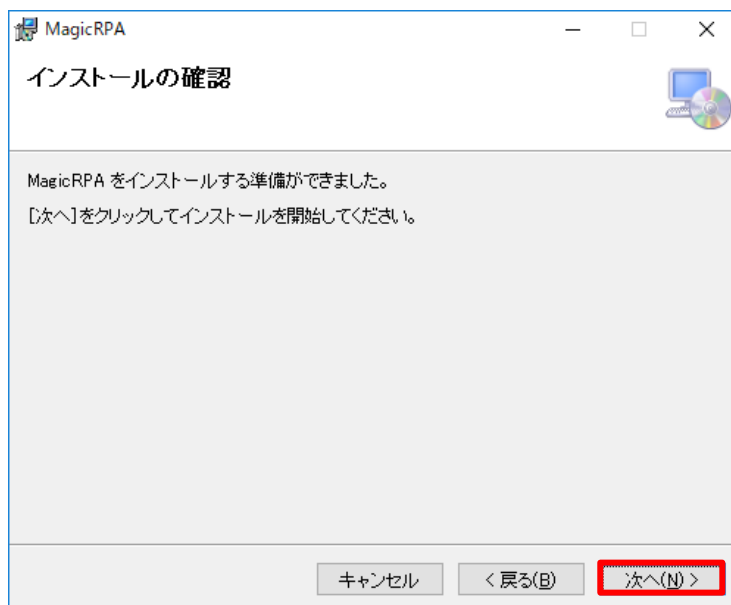
シリアル番号(S):
12 - 3456789123456789123456789123456789123456

キャンセル < 戻る(B) 次へ(N) >

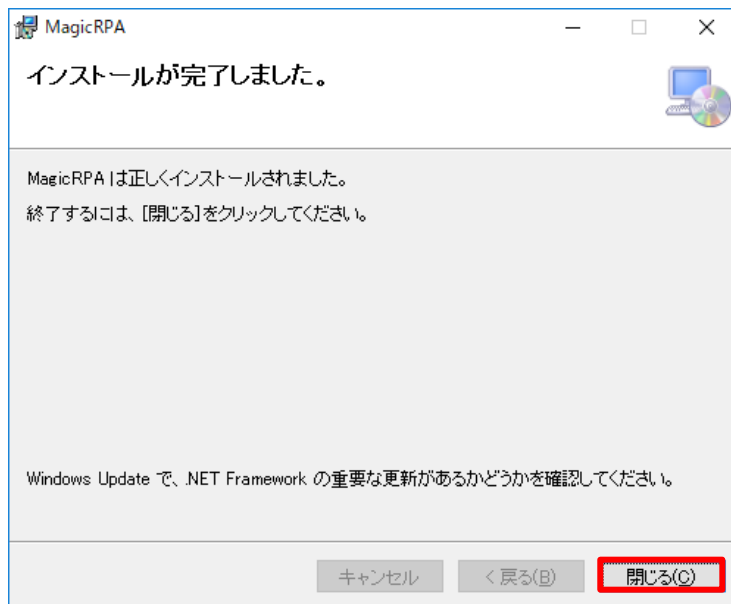
5. インストールフォルダーの選択が表示されるので、インストール先を選択し、「次へ(N)」をクリックします。



6. インストールの確認が表示されるので、「次へ(N)」をクリックします。



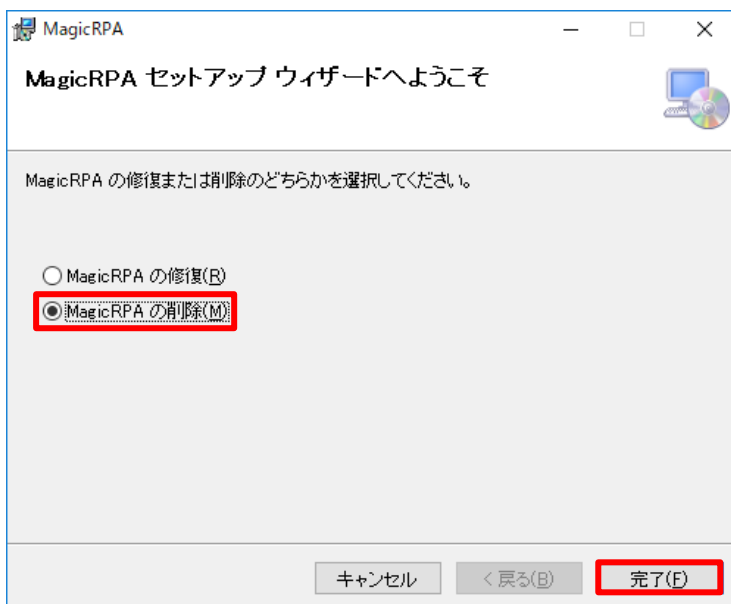
7. インストール完了画面が表示されますので、「閉じる(C)」をクリックします。



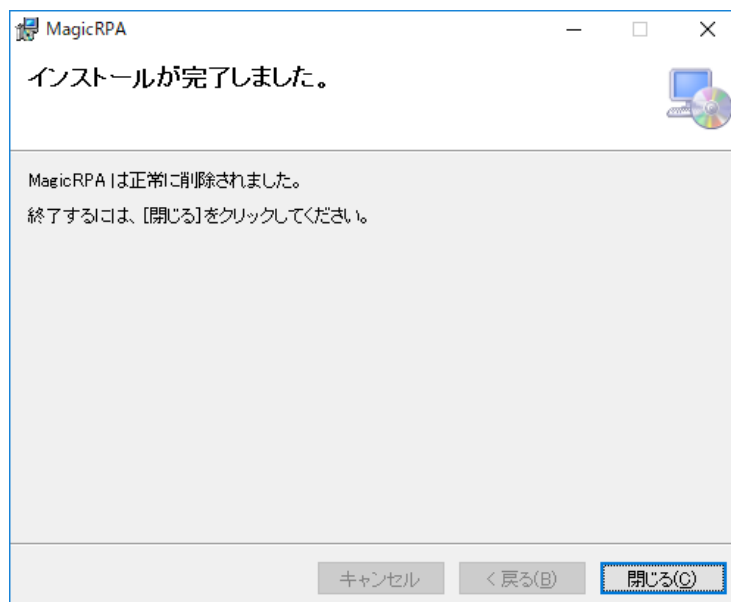
3.2. プログラムのアンインストール方法

MagicRPA のアンインストール手順です。

1. インストール時にダウンロードしたファイルの中から「MagicRPA.msi」をダブルクリックします。
2. MagicRPA セットアップウィザードが表示されるので、「MagicRPA の削除(M)」にチェックを入れ、「完了(F)」をクリックします。



3. アンインストール完了画面が表示されますので、「閉じる(C)」をクリックします。



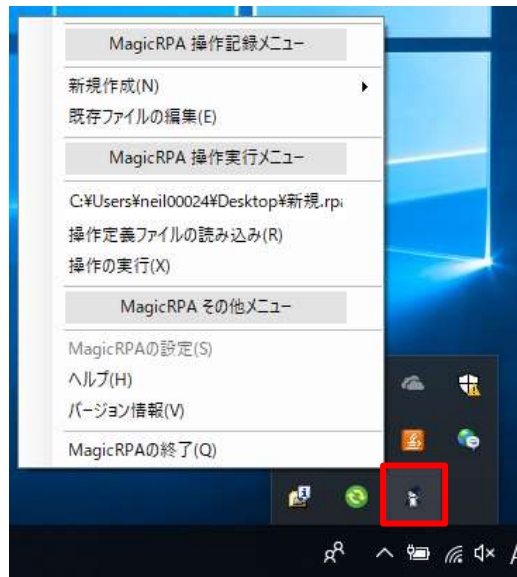
3.3. MagicRPA の起動

MagicRPA を起動します。

1. デスクトップ上の MagicRPA.exe をダブルクリックします。



2. アイコンがタスクバーに常駐するので、右クリックでコンテキストメニューを表示します。



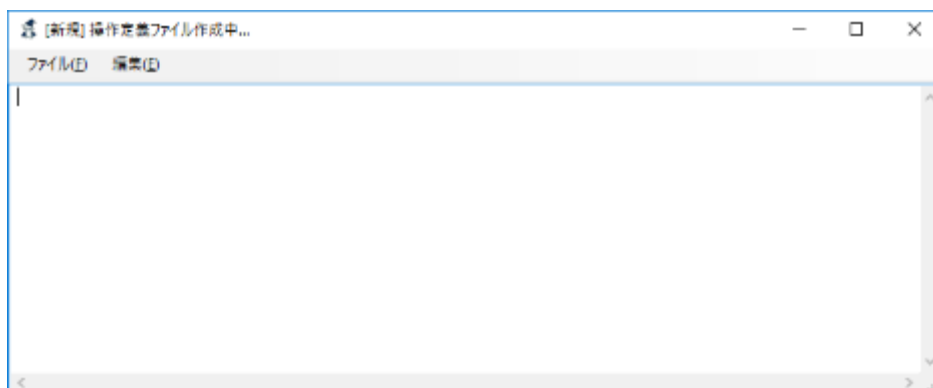
3.4. MagicRPA エディタの起動

プログラムを作成するためのエディタを起動します。

1. コンテキストメニュー内の「新規作成 (N)」をクリックし、「エディタで作成」をクリックします。



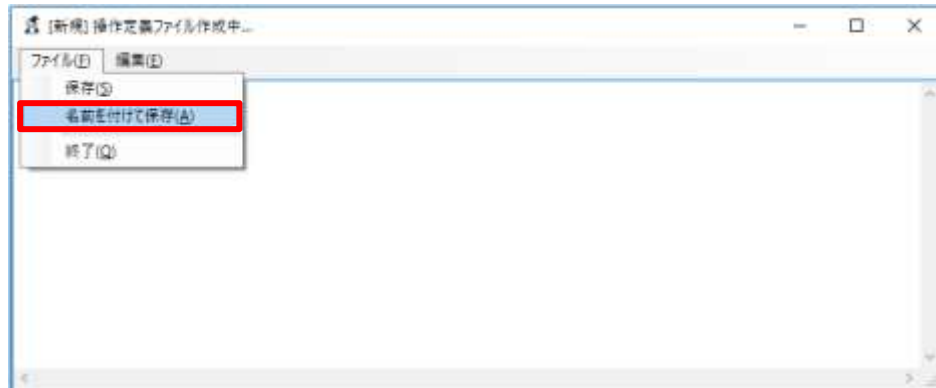
2. エディタが起動しますので、プログラムを作成します。



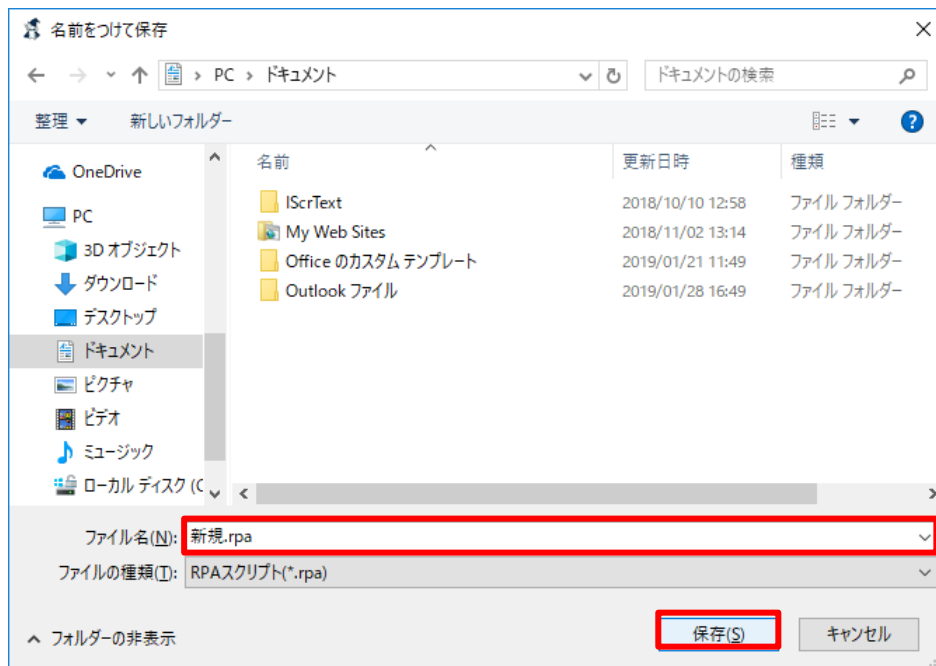
3.5. RPA プログラムの保存

作成したプログラムを保存します。

1. エディタ内「ファイル(F)」をクリックし、「名前を付けて保存(A)」します。
上書き保存をしたい場合は、「保存(S)」をクリックします。



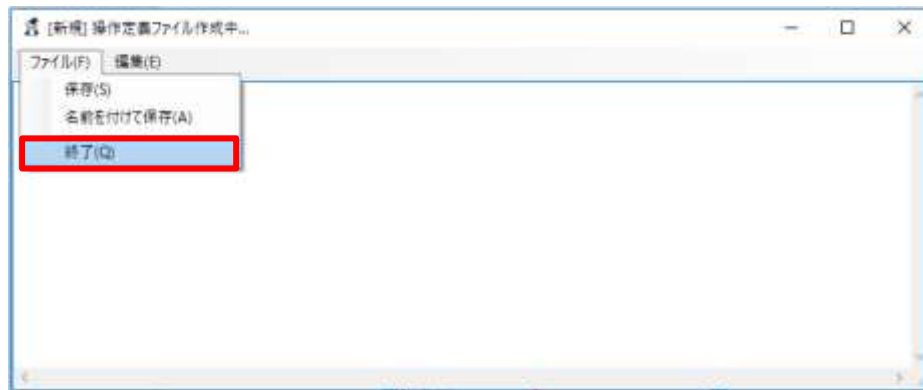
2. 任意の名前を付けて「保存」をクリックします。



3.6. MagicRPA エディタの終了

エディタを終了します。

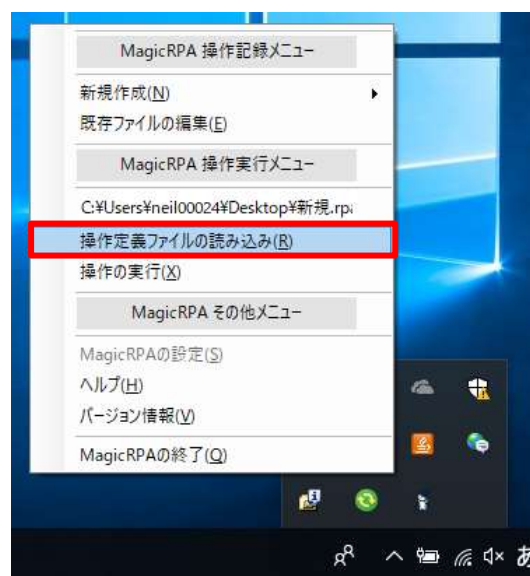
1. エディタ内の「ファイル(F)」をクリックし、「終了(Q)」をクリックします。
画面右上の「×」ボタンのクリックでも終了できます。



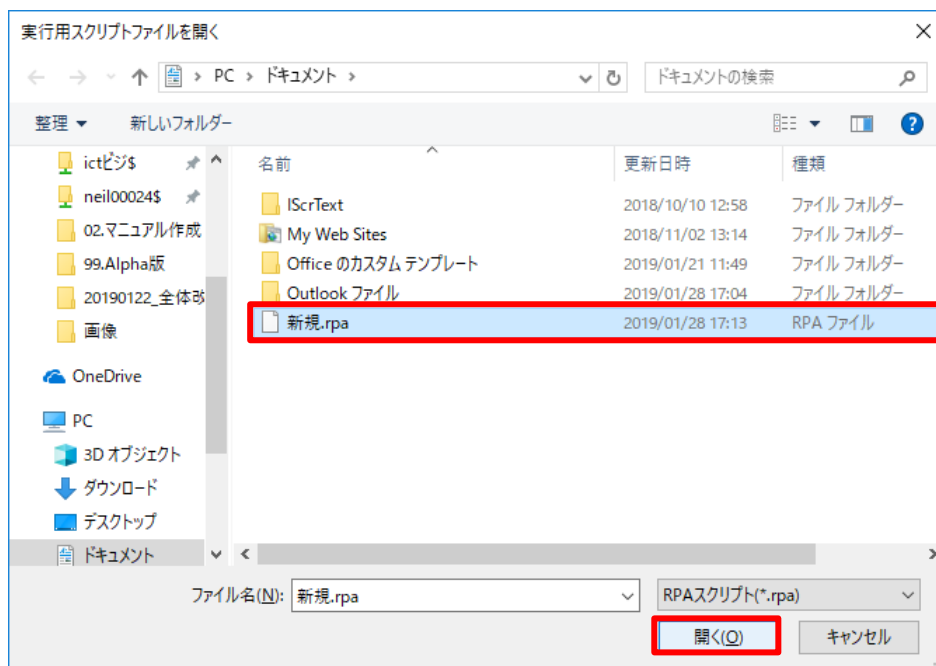
3.7. RPA プログラムの実行

作成したプログラムを実行します。

1. コンテキストメニュー内の「操作定義ファイルの読み込み」をクリックし、実行したいプログラムを読み込みます。



2. 「実行用スクリプトファイルを開く」にて、実行したいファイルを選択し、「開く」をクリックします。

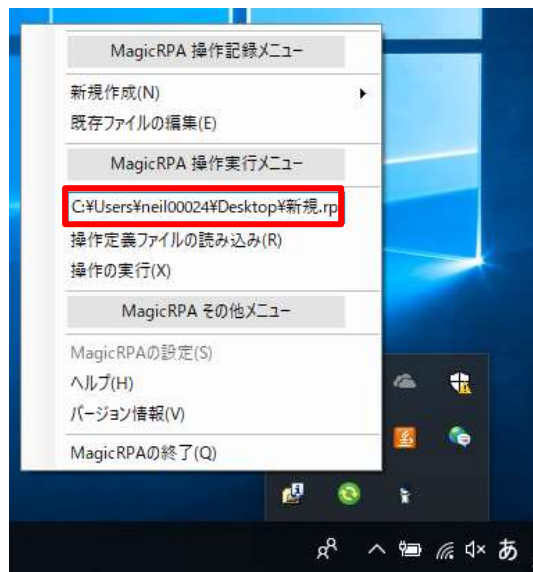


3. メニュー内「操作の実行」をクリックし、プログラムを実行します。



<<補足>>ファイルを読み込んでいない状態でのクリックはできません。

4. コンテキストメニュー内のファイルパスをダブルクリックで、エディタが表示されます。



3.8. RPA プログラムの終了

実行中のプログラムの処理が完了した時点で自動的に終了します。

実行中の RPA プログラムを終了する場合は、[3.9.RPA プログラムの停止\(強制停止\)](#)を参照ください。

3.9. RPA プログラムの停止（強制停止）

「Pause」キーをクリックし、無限ループが発生している実行中の RPA プログラムや、処理待ちなどで実行状態でありながらストップしてしまった RPA プログラムを強制停止します。

3.10. MagicRPA の終了

MagicRPA を終了します。

1. メニュー内「MagicRPA の終了(Q)」をクリックします。



3.11. コマンドライン実行

MagicCmd.exe を利用し、MagicRPA の RPA プログラムをコマンドラインから実行する事により、他のプログラムから RPA プログラムの呼び出しやタスクスケジューラ等による自動実行が行えるようになります。

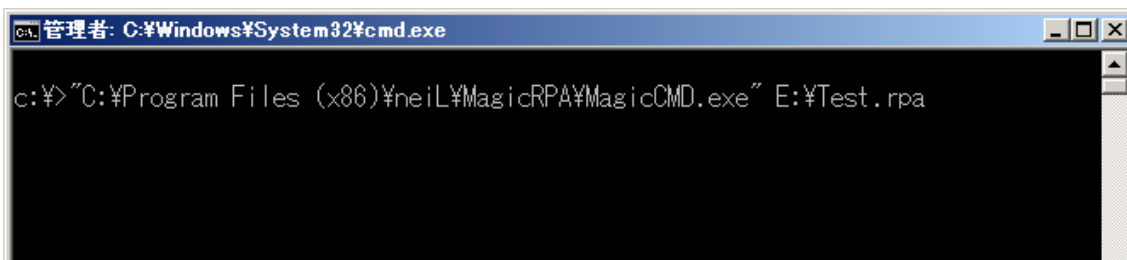
MagicCmd.exe は「管理者として実行」で起動したコマンドプロンプトより、下記のように実行してください。

MagicCMD.exe <<実行するスクリプト>>

※例 1 : "C:¥Program Files (x86)¥neil¥MagicRPA¥MagicCMD.exe" E:¥Test.rpa

※例 2 : cd "C:¥Program Files (x86)¥neil¥MagicRPA¥"

MagicCMD.exe E:¥Test.rpa

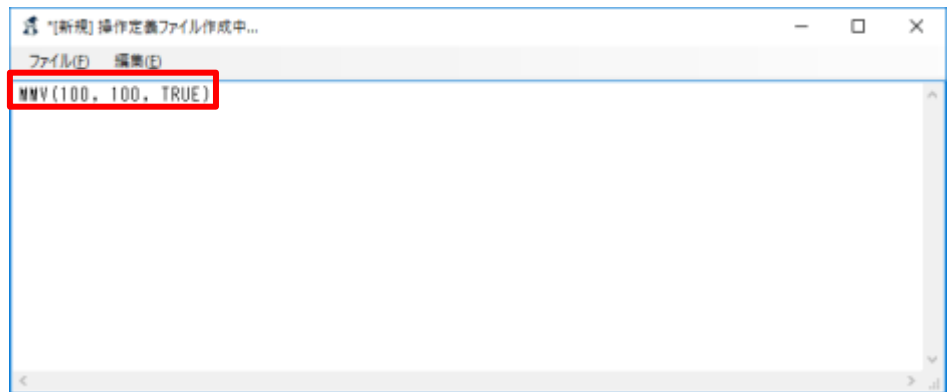


4. RPA プログラム作成ガイド

4.1. 処理の流れ

プログラム作成における処理の流れを説明します。
処理は上から順番に処理されます。

- 1. エディタ内に利用したいコマンドとパラメータを記入します。
エディタの開き方は [3.4.MagicRPA エディタの起動](#)を参照ください。



- 2. プログラムを保存します。
《補足》詳細は [3.5.RPA プログラムの保存](#)を参照ください。
- 3. プログラムを実行します。
《補足》詳細は [3.7.RPA プログラムの実行](#)を参照ください。

4.2. コマンド一覧

4.2.1. INI ファイルの制御

4.2.2. DELETEINI コマンド

コマンド名	DELETEINI	
機能	INI ファイルのデータを削除します。	
書式	DELETEINI (ファイル名[String], セクション名[String], キー名[String])	
引数の説明 ※[]で囲っている引数は省略可能です。	ファイル名	INI ファイル名をフルパスで指定します。
	セクション名	INI ファイルのセクション名を指定します。
	[キー名]	INI ファイルのキー名を指定します。 指定しない場合、セクション毎削除を行います。
戻り値	削除が正常に行えた : TRUE 削除に失敗した : FALSE	
備考		

4.2.2.1. READINI コマンド

コマンド名	READINI	
機能	INI ファイルよりデータを読み込みます。	
書式	READINI(ファイル名 [String], セクション名 [String], キー名 [String])	
引数の説明	ファイル名	INI ファイル名をフルパスで指定します。
	セクション名	INI ファイルのセクション名を指定します。
	キー名	INI ファイルのキー名を指定します。
戻り値	正常 : 取得したデータ 異常 : "" (空文字)	
備考		

4.2.2.2. WRITEINI コマンド

コマンド名	WRITEINI	
機能	INI ファイルにデータを書き込みます。	
書式	WRITEINI(ファイル名 [String], セクション名 [String], キー名 [String], 値 [String])	
引数の説明	ファイル名	INI ファイル名をフルパスで指定します。
	セクション名	INI ファイルのセクション名を指定します。
	キー名	INI ファイルのキー名を指定します。
	値	INI ファイルに書き込む値を指定します。
戻り値	書き込みが正常に行えた : TRUE 書き込みに失敗した : FALSE	
備考		

4.2.3. ウィンドウの制御

4.2.3.1. ACW コマンド

コマンド名	ACW	
機能	ウィンドウのサイズや位置を変更します。	
書式	ACW(ウィンドウ名[String], X 座標[Short], Y 座標[Short], 幅[Short], 高さ[Short], アクティブ化フラグ[Boolean], 待機時間[Short])	
引数の説明 ※[]で囲っている引数は省略可能です。	ウィンドウ名	完全一致による指定が必要です。
	[X 座標]	ウィンドウの左上から X 座標を指定します。 省略時は操作しません。
	[Y 座標]	ウィンドウの左上から Y 座標を指定します。 省略時は操作しません。
	[幅]	ウィンドウの幅を指定します。 省略時は操作しません。
	[高さ]	ウィンドウの高さを指定します。 省略時は操作しません。
	[アクティブ化フラグ]	ウィンドウをアクティブ化するか指定します。 TRUE: アクティブ化する FALSE: アクティブ化しない 省略時は FALSE が設定されます。
	[待機時間]	処理完了後の待機時間(ミリ秒)を指定します。 省略時は待機しません。
戻り値	ウィンドウが見つかり、操作が行えた : TRUE ウィンドウが見つからない、または操作に失敗した : FALSE	
備考		

4.2.3.2. GETALLWIN コマンド

コマンド名	GETALLWIN	
機能	全てのウィンドウハンドルを取得します。	
書式	GETALLWIN(ウィンドウハンドル[hwnd], 取得ハンドルの数[String], 取得ハンドルの値[String])	
引数の説明	ウィンドウハンドル	「0」を指定すると全てのハンドルを取得します。ハンドルを指定した場合は、子プロセスのハンドルを対象とします。
	取得ハンドルの数	取得したハンドルの数を格納する変数を指定します。
	取得ハンドルの値	取得したハンドルの値を格納する配列変数を指定します。
戻り値	ハンドルを取得できた : TRUE ハンドルを取得できない : FALSE	
備考		

4.2.3.3. GETHND コマンド

コマンド名	GETHND	
機能	ウィンドウ内オブジェクトのハンドルを取得します。	
書式	GETHND(ウィンドウハンドル[hwnd], 表示名[String], 順番[UInteger])	
引数の説明 ※[]で囲っている引数は省略可能です。	ウィンドウハンドル	実行中のウィンドウハンドルを指定します。
	表示名	取得するハンドルに表示されたキャプション名を指定します。
	[順番]	同一キャプションがある場合、取得する順番を指定します。 既定値は 1 (最初に見つかったもの) です。
戻り値	ハンドルを取得できた : ハンドル値 ハンドルを取得できない : 0	
備考		

4.2.3.4. GETID コマンド

コマンド名	GETID	
機能	ウィンドウが表示されたかチェックします。	
書式	GETID(ウィンドウタイトル[String], 待機時間[Decimal])	
引数の説明 ※[]で囲っている引数は省略可能です。	ウィンドウタイトル	チェックするウィンドウのタイトルを指定します。 ※ウィンドウタイトルは完全一致で検索します。 見つからない場合は部分一致で再検索します。
	[待機時間]	ウィンドウが表示されるまでの待機時間(秒)を指定します。 既定値は-1(待機しない)です。
戻り値	指定タイトルのウィンドウを取得できた : ウィンドウのハンドル値 指定タイトルのウィンドウを取得できない : -1	
備考		

4.2.4. 外部プログラムの実行

4.2.4.1. CMDEXEC コマンド

コマンド名	CMDEXEC	
機能	外部プログラムを実行します。	
書式	CMDEXEC(実行ファイル名[String], 同期フラグ[Boolean], X 座標[Short], Y 座標[Short], 幅[Short], 高さ[Short])	
引数の説明 ※[]で囲っている引数は省略可能です。	実行ファイル名	実行するファイルを指定します。
	[同期フラグ]	実行終了まで待機するかを指定します。 既定値は FALSE(同期する)です。
	[X 座標,Y 座標]	実行ファイルのウィンドウが表示される場所を指定します。 既定値はシステム既定です。
	[幅,高さ]	実行ファイルのウィンドウサイズを指定します。 既定値はシステム既定です。
戻り値	プログラムが実行できない : -1 プログラム実行された(同期フラグ:TRUE) : 実行されたプログラムのハンドル値 プログラム実行された(同期フラグ:FALSE) : コマンドの戻り値	
備考		

4.2.4.2. DOSCMD コマンド

コマンド名	DOSCMD	
機能	コマンドプロンプト経由でコマンドを実行します。	
書式	DOSCMD(コマンド[String], 同期フラグ[Boolean], 表示/非表示フラグ[Boolean], 出力[String])	
引数の説明 ※[]で囲っている引数は省略可能です。	コマンド	実行するコマンドを指定します。
	[同期フラグ]	コマンド実行終了まで待機するか指定します。 規定値は FALSE(同期する)です。
	[表示/非表示フラグ]	コマンドプロンプトを表示するかを指定します。 規定値は FALSE(表示しない)です。
	[出力]	標準出力に出力された結果を保存する変数を指定します。 ※同期フラグが有効でないと結果は保存されません。
戻り値	コマンドが実行された : TRUE コマンドが実行できない : FALSE	
備考		

4.2.4.3. POWERSHELL コマンド

コマンド名	POWERSHELL	
機能	POWERSHELL 経由でコマンドを実行します。	
書式	POWERSHELL(コマンド[String], 同期フラグ[Boolean], 表示/非表示フラグ[Boolean], 出力[String])	
引数の説明 ※[]で囲っている引数は省略可能です。	コマンド	実行するコマンドを指定します。
	[同期フラグ]	コマンド実行終了まで待機するか指定します。 既定値は FALSE(同期する)です。
	[表示/非表示フラグ]	POWERSHELL 画面を表示するか指定します。 既定値は FALSE(表示しない)です。
	[出力]	標準出力に出力された結果を保存する変数を指定します。 ※同期フラグが有効でないと結果は保存されません。
戻り値	コマンドが実行された : TRUE コマンドが実行できない : FALSE	
備考		

4.2.5. 画像の制御

4.2.5.1. CHKIMG コマンド

コマンド名	CHKIMG	
機能	指定画像が画面上にあるかチェックします。	
書式	CHKIMG(画像名[String], 透過設定フラグ[Integer], X 座標[Short], Y 座標[Short], 幅[Short], 高さ[Short], ヒット回数[UInteger], 画像誤差判定[UInteger])	
引数の説明 ※[]で囲っている引数は省略可能です。	画像名	画像名のフルパスを指定します。
	[X 座標]	画面の左上から X 座標を指定します。 省略時は操作しません。
	[Y 座標]	画面の左上から Y 座標を指定します。 省略時は操作しません。
	[幅]	画面の幅を指定します。 省略時は操作しません。
	[高さ]	画面の高さを指定します。 省略時は操作しません。
	[ヒット回数]	ヒット回数を格納する変数を指定します。
	[画像誤差判定]	画像の誤差を判定します。
戻り値	画像がある : TRUE 画像がない : FALSE	
備考		

4.2.5.2. PEEKCOLOR コマンド

コマンド名	PEEKCOLOR	
機能	デスクトップ画面、もしくはクリップボードに保存された画像の特定座標の色情報を取得します。	
書式	PEEKCOLOR(X 座標[Short], Y 座標[Short], クリップボード [Boolean], 出力順序[Boolean])	
引数の説明 ※[]で囲っている引数は省略可能です。	[X 座標、Y 座標]	色情報を取得する際の X 座標、Y 座標を指定します。
	[クリップボード]	TRUE の場合、クリップボードに保存された画像から色情報を取得します。 FALSE の場合、デスクトップ画面から色情報を取得します。 既定値は FALSE です。
	[出力順序]	色情報を出力する順序を指定します。 TRUE : RGB FALSE : BGR 既定値は FALSE です。
戻り値	取得できた : 取得した色情報を 32 ビットで出力 0x00RRGGBB もしくは 0x00BBGGRR 取得できない : -1 (0xFFFFFFFF)	
備考		

4.2.5.3. SEARCHIMG コマンド

コマンド名	SEARCHIMG	
機能	画面上の画像を検索し、見つかった座標を返します。	
書式	SEARCHIMG (X 座標[String], Y 座標[String], 検索する画像 [String], 透過色/色無視[UInteger], 順番[UInteger], 青色幅 [UInteger], 緑色幅[UInteger], 赤色幅[UInteger])	
引数の説明 ※[]で囲っている引数は省略可能です。	[X 座標]	検索後に見つかった画像の X 座標を保存する変数を指定します。 X 座標・Y 座標省略時は画面全体を対象とします。
	[Y 座標]	検索後に見つかった画像の Y 座標を保存する変数を指定します。
	[検索する画像]	検索する画像のフルパスを指定します。 省略時はクリップボードの画像が指定されます。

	[透過色/色無視]	<p>透過色を指定します。</p> <p>0 : 透過色を設定しない(既定値)</p> <p>1 : 検索する画像の左上 1 ピクセルを透過色とします。</p> <p>2 : 検索する画像の右上 1 ピクセルを透過色とします。</p> <p>3 : 検索する画像の左下 1 ピクセルを透過色とします。</p> <p>4 : 検索する画像の右下 1 ピクセルを透過色とします。</p> <p>また、本指定を-1 にすると、色を無視した形状比較を行います。</p>
	[順番]	<p>検索対象が複数存在する場合の取得順を指定します。</p> <p>省略時は初めに見つかった画像を取得します。</p>
	[色幅]	<p>青、緑、赤に対して、チェック時に色幅を持たせます(省略時は完全一致のみ)。</p> <p>IMG_MSK_1 : 各色に対し、2/256 の色幅を許します。</p> <p>IMG_MSK_2 : 各色に対し、4/256 の色幅を許します。</p> <p>IMG_MSK_3 : 各色に対し、8/256 の色幅を許します。</p> <p>IMG_MSK_4 : 各色に対し、16/256 の色幅を許します。</p> <p>IMG_MSK_5 : 各色に対し、32/256 の色幅を許します。</p> <p>IMG_MSK_6 : 各色に対し、64/256 の色幅を許します。</p> <p>IMG_MSK_7 : 各色に対し、128/256 の色幅を許します。</p> <p>IMG_MSK_8 : 各色に対し、チェックを行いません。</p>
戻り値	<p>画像が見つかった : TRUE</p> <p>画像が見つからない : FALSE</p>	
備考		

4.2.6. クリップボードの操作

4.2.6.1. CLIPMGT コマンド

コマンド名	CLIPMGT	
機能	クリップボードの操作を行います。	
書式	CLIPMGT(値[String], 操作[Boolean])	
引数の説明 ※[]で囲っている引数は省略可能です。	値	クリップボード操作時に使用する変数を指定します。
	操作	クリップボードの操作方法を指定します。規定値は TRUE(クリップボードから値を取得)です。 TRUE : クリップボードから値を取得 FALSE : クリップボードに値をセット
戻り値	操作が成功した : TRUE 操作が失敗した : FALSE	
備考		

4.2.7. コメントの挿入

4.2.7.1. REM コマンド

コマンド名	REM	
機能	該当行の処理を行いません。(コメント行となります。)	
書式	REM(コメント[String])	
引数の説明	コメント	コメントを指定します。
戻り値	-	
備考	詳細は 4.4.4. REM コマンド を参照ください。	

4.2.8. 条件処理

4.2.8.1. IF コマンド

コマンド名	IF	
機能	IF に設定した条件に応じて、IF から ELSE 間、もしくは IF から FI 間の処理を実行します。	
書式	IF [真の場合の式]	
戻り値	真の場合の式が TRUE だった場合 : TRUE 真の場合の式が FALSE だった場合 : FALSE	
備考	詳細は 4.4.7. IF-FI コマンド を参照ください。	

4.2.8.2. ELSE コマンド

コマンド名	ELSE	
機能	ELSE に設定した条件に応じて、ELSE から FI 間の処理を実行します。 必要に応じて、偽の場合の条件を設定します。	
書式	ELSE [偽の場合の式(省略可能)]	
戻り値	-	
備考	詳細は 4.4.7. IF-FI コマンド を参照ください。	

4.2.8.3. FI コマンド

コマンド名	FI	
機能	条件処理を終了します。	
書式	FI	
戻り値	処理が正常に行われた : TRUE 処理に失敗した : FALSE	
備考	詳細は 4.4.7. IF-FI コマンド を参照ください。	

4.2.8.4. TRY コマンド

コマンド名	TRY
機能	TRY から EXCEPT の間の処理を実行します。 この間でエラーが発生した場合は 4.2.9.5. EXCEPT コマンド に進みます。 EXCEPT-ENDTRY 間の処理が完了するか EXIT した場合、TRY-EXCPT 間のエラーが発生した次の行から処理を再開します。
書式	TRY
戻り値	-
備考	詳細は 4.4.6. TRY-ENDTRY コマンド を参照ください。

4.2.8.5. EXCEPT コマンド

コマンド名	EXCEPT
機能	EXCEPT-ENDTRY 間の処理を実行します。 EXCEPT-ENDTRY 間の処理が完了するか EXIT した場合、TRY-EXCPT 間のエラーが発生した次の行から処理を再開します。
書式	EXCEPT
戻り値	-
備考	詳細は 4.4.6. TRY-ENDTRY コマンド を参照ください。

4.2.8.6. ENDTRY コマンド

コマンド名	ENDTRY
機能	TRY 処理を終了します。
書式	ENDTRY
戻り値	処理が正常に行われた : TRUE 処理に失敗した : FALSE
備考	詳細は 4.4.6. TRY-ENDTRY コマンド を参照ください。

4.2.9. 数学関数

4.2.9.1. ARCCOS コマンド

コマンド名	ARCCOS
機能	逆三角関数(アークコサイン)を記述します。
書式	ARCCOS(値[Integer])
戻り値	計算結果
備考	

4.2.9.2. ARCSIN コマンド

コマンド名	ARCSIN
機能	逆三角関数(アークサイン)を記述します。
書式	ARCSIN(値[Integer])
戻り値	計算結果
備考	

4.2.9.3. ARCTAN コマンド

コマンド名	ARCTAN
機能	逆三角関数(アークタンジェント)を記述します。
書式	ARCTAN(値[Integer])
戻り値	計算結果
備考	

4.2.9.4. CEIL コマンド

コマンド名	CEIL
機能	小数点を切り上げる数学関数です。
書式	CEIL(値[Decimal])
戻り値	小数点を切り上げた値
備考	

4.2.9.5. CHKNUM コマンド

コマンド名	CHKNUM
機能	数値であるか調べます。
書式	CHKNUM(値[Integer])
戻り値	数値だった : TRUE 数値じゃなかった : FALSE
備考	

4.2.9.6. COS コマンド

コマンド名	COS
機能	三角関数(コサイン)を記述します。
書式	COS(値[Integer])
戻り値	計算結果
備考	

コマンド名	LET	
機能	RPA 変数の演算を行います。	
書式	LET(①変数名/定数名[String] 演算子[Operator] ②変数名/定数名[String] = ③変数名/定数名[String])	
引数の説明	①変数名/定数名	演算を行いたい値の 1 つ目が格納されている変数/定数を指定します。
	演算子	<p>行いたい演算に沿った演算子を指定します。 指定可能演算子</p> <p>(), [] 優先順位高い</p> <p>*, /, MOD</p> <p>+, -, &</p> <p>>, <, =, >=, <=, <></p> <p>AND</p> <p>OR, XOR 優先順位低い</p>
	②変数名/定数名	演算を行いたい値の 2 つ目が格納されている変数/定数を指定します。
	③変数名/定数名	演算結果を格納する変数/定数を指定します。
戻り値	<p>演算が正常に行われた : TRUE</p> <p>演算に失敗した : FALSE</p>	
備考	詳細は 4.4.3. LET コマンド を参照ください。	

4.2.9.10. LOG コマンド

コマンド名	LOG
機能	自然対数を求める数学関数です。 引数が1つであれば自然対数となり、2つであれば底を指定します。
書式	LOG(値[Double]) LOGN(値[Double], 底[Integer])
戻り値	計算結果
備考	

4.2.9.11. LOG10 コマンド

コマンド名	LOGN10
機能	常用対数を求める数学関数です。
書式	LOGN10(値[Double])
戻り値	計算結果
備考	

4.2.9.12. POWER コマンド

コマンド名	POWER
機能	累乗を求める数学関数です。
書式	POWER(数値[Double], 指数[Double])
戻り値	計算結果
備考	

4.2.9.13. RANDOM コマンド

コマンド名	RANDOM
機能	ランダムな数値を返します。
書式	RANDOM(レンジ[Double])
戻り値	数値
備考	

4.2.9.14. ROUND コマンド

コマンド名	ROUND
機能	四捨五入を行う数学関数です。
書式	ROUND(数値[Double], 丸める桁数[Integer])
戻り値	計算結果
備考	

4.2.9.15. SIN コマンド

コマンド名	SIN
機能	三角関数(サイン)を記述します。
書式	SIN(値[Integer])
戻り値	計算結果
備考	

4.2.9.16. SQRT コマンド

コマンド名	SQRT
機能	平方根を求める数学関数です。
書式	SQRT(値[Integer])
戻り値	計算結果
備考	

4.2.9.17. TAN コマンド

コマンド名	TAN
機能	三角関数(タンジェント)です。
書式	TAN(値[Integer])
戻り値	計算結果
備考	

4.2.9.18. ZCUT コマンド

コマンド名	ZCUT
機能	マイナス値をゼロにする数学関数です。
書式	ZCUT(値[Integer])
戻り値	計算結果
備考	

4.2.10. スキップ

4.2.10.1. BREAK コマンド

コマンド名	BREAK
機能	TRY-ENDTRY,IF-FI,FOR-NEXT,WHILE-WEND,REPEAT-UNTIL を処理している際に、プログラム終了までプログラム処理をスキップ します。 プログラムを再開する際は、FI や NEXT など処理を終了するコマン ドの次の行から行われます。
書式	BREAK
戻り値	-
備考	詳細は 4.4.11. BREAK コマンド 参照ください。

4.2.11. スリープ

4.2.11.1. SLEEP コマンド

コマンド名	SLEEP
機能	処理を待機します。
書式	SLEEP()
戻り値	処理の待機に成功した : TRUE 処理の待機に失敗した : FALSE
備考	

4.2.12.配列の制御

4.2.12.1. CALARRAY コマンド

コマンド名	CALARRAY	
機能	配列に対して計算を行います。	
書式	CALARRAY(結果[String], 対象の配列[Integer], 計算方式[Integer], 開始位置[UInteger], 終了位置[UInteger])	
引数の説明 ※[]で囲っている引数は省略可能です。	結果	結果を保存する変数を指定します。
	対象の配列	計算を行う対象配列を指定します。
	計算方式	計算方式は以下文字列で指定します。 ADD : 配列に保存された値を全て加算した値を取得します。 MIN : 配列に保存された値の最小値を取得します。 MAX : 配列に保存された値の最大値を取得します。 AVR : 配列に保存された値の平均値を取得します。
	[配列開始位置]	演算対象配列の開始位置を指定します。 省略時は先頭が指定されます。
	[配列終了位置]	演算対象配列の終了位置を指定します。 省略時は末尾が指定されます。
戻り値	計算に成功した : TRUE 計算に失敗した : FALSE	
備考		

4.2.12.2. JOIN コマンド

コマンド名	JOIN	
機能	配列の中身を区切り文字で結合します。	
書式	JOIN(配列名[String], 区切り文字[String], 開始位置[UInteger], 終了位置[UInteger])	
引数の説明 ※[]で囲っている引数は省略可能です。	配列名	配列が格納されている変数名を指定します。
	区切り文字	配列を結合する際の区切り文字を指定します。
	[開始位置]	演算対象配列の開始位置を指定します。 省略時は先頭が指定されます。
	[終了位置]	演算対象配列の終了位置を指定します。 省略時は末尾が指定されます。
戻り値	配列の中身を結合できた : TRUE 配列の中身を結合できない : FALSE	
備考		

4.2.13. ファイルの制御

4.2.13.1. GETDIR コマンド

コマンド名	GETDIR	
機能	ディレクトリ内のファイル名一覧を取得します。	
書式	GETDIR(ファイル名[String], ディレクトリ[String], ワイルドカード[UInteger], 隠しファイル表示[Boolean], 並び順[UInteger])	
引数の説明 ※[]で囲っている引数は省略可能です。	ファイル名	取得したファイル名を格納する配列変数を指定します。
	ディレクトリ	ファイル名一覧を取得するディレクトリを指定します。
	[ワイルドカード]	ファイル名を検索するための条件を指定します。 規定値は""(すべての文字列)です。 * : すべての文字列 ? : 任意の 1 文字
	[隠しファイル表示]	隠しファイルの表示有無を指定します。 規定値は FALSE(表示しない)です。
	[並び順]	出力結果の並び順を指定します。 規定値は 0(ファイル名でソート)です。 0 : ファイル名 1 : サイズ 2 : 作成日 3 : 更新日 4 : アクセス日
戻り値	ファイル一覧を取得できた : TRUE ファイル一覧を取得できない : FALSE	
備考		

4.2.14. 変数/定数の操作

4.2.14.1. CONST コマンド

コマンド名	CONST	
機能	RPA 定数を宣言します。	
書式	CONST(定数名[String])	
引数の説明	定数名	宣言したい定数名を指定します。
戻り値	-	
備考	詳細は 4.4.5. CONST コマンド を参照ください。	

4.2.14.2. DIM コマンド

コマンド名	DIM	
機能	RPA のローカル変数を宣言します。	
書式	DIM(変数名[String])	
引数の説明	変数名	宣言したいローカル変数名を指定します。
戻り値	-	
備考	詳細は 4.4.1. DIM コマンド を参照ください。	

4.2.14.3. PUBLIC コマンド

コマンド名	PUBLIC	
機能	RPA のグローバル変数を宣言します。	
書式	PUBLIC(変数名[String])	
引数の説明	変数名	宣言したいグローバル変数名を指定します。
戻り値	-	
備考	詳細は 4.4.2. PUBLIC コマンド を参照ください。	

4.2.14.4. VARTYPE コマンド

コマンド名	VARTYPE	
機能	変数の型を取得、または変換を行います。	
書式	VARTYPE(判別する変数[String], 変換後の型[String])	
引数の説明 ※[]で囲っている引数は省略可能です。	判別する変数	判別する変数を指定します。
	[変換後の型]	変換後の型を下記から指定します。 BOOLEAN STRING INTEGER CINTEGER DECIMAL HANDLE DATE NULL
戻り値	変数の型に応じた数値を返します。 Null : 0 真偽値 : 1 符号なし整数 : 2 符号付き整数 : 3 小数を含む値 : 4 ポインタ : 5 文字列 : 6 配列 : 8 Date 型 : 9 その他 : 10 変数の型を変換した場合 : 変換後の値 型変換に失敗した場合 : NULL	
備考		

4.2.15. プログラムの終了

4.2.15.1. QUIT コマンド

コマンド名	QUIT
機能	プログラムそのものを終了します。
書式	QUIT
戻り値	-
備考	詳細は 4.4.13. QUIT コマンド を参照ください。

4.2.16. ボタンの制御

4.2.16.1. CHKBTN コマンド

コマンド名	CHKBTN	
機能	ボタン、チェックボックス、ラジオボタン等の状態をチェックします。	
書式	CHKBTN(ウィンドウ名[String], 対象アイテム[String], 順番[UInteger])	
引数の説明 ※[]で囲っている引数は省略可能です。	ウィンドウ名	ウィンドウ名を指定します。 完全一致による指定が必要です。
	対象アイテム名	対象アイテムの表示名を指定します。 完全一致による指定が必要です。
	[順番]	何個目の対象アイテムを処理対象とするか指定します。 省略時は 1 つ目の対象アイテムが指定されます。
戻り値	アイテム名やウィンドウが見つからない : -1 アイテムは見つかったが、チェックが入っていない : 0 アイテムが見つかり、チェックが入っている : 1	
備考		

4.2.17. マウス操作の制御

4.2.17.1. BTN コマンド

コマンド名	BTN	
機能	マウスキーをクリックします。	
書式	BTN(X 座標[Short], Y 座標[Short], クリックキー[UInteger])	
引数の説明 ※[]で囲っている引数は省略可能です。	X 座標	クリックする X 座標を指定します。
	Y 座標	クリックする Y 座標を指定します。
	クリックキー	クリックするボタンを指定します。 1 : 左クリック 2 : 右クリック 3 : センターボタン
戻り値	クリック操作が行えた : TRUE クリック操作に失敗した : FALSE	
備考		

4.2.17.2. MMV コマンド

コマンド名	MMV	
機能	マウスカーソルを移動します。	
書式	MMV(X 座標[Short], Y 座標[Short])	
引数の説明	X 座標	マウスカーソルにおける移動先の X 座標を指定します。
	Y 座標	マウスカーソルにおける移動先の Y 座標を指定します。
戻り値	正常にマウスカーソルを移動できた : TRUE マウスカーソルの移動に失敗した : FALSE	
備考		

4.2.17.3. MDG コマンド

コマンド名	MDG	
機能	マウスカーソルをドラッグします。	
書式	MDG(X 座標[Short], Y 座標[Short], ドラッグするボタン [UInteger])	
引数の説明 ※[]で囲っている引数は省略可能です。	X 座標	マウスカーソルにおける移動先の X 座標を指定します。
	Y 座標	マウスカーソルにおける移動先の Y 座標を指定します。
	[ドラッグするボタン]	ドラッグ時に使用するボタンを指定します。 規定値は 1 (左クリック) です。 1 : 左クリック 2 : 右クリック 3 : センターボタン
戻り値	正常にマウスカーソルをドラッグできた : TRUE マウスカーソルのドラッグに失敗した : FALSE	
備考		

4.2.18. キーボードの制御

4.2.18.1. KBD コマンド

コマンド名	KBD	
機能	キーボードを操作します。	
書式	KBD(クリックキー[String], 挙動[UInteger])	
引数の説明 ※[]で囲っている引数は省略可能です。	クリックキー	操作するキーをキーコードで指定します。 キーコードは 6.1 キーコード一覧 を参照ください。
	[挙動]	キーの Updown 挙動を指定します。 省略時は 0(クリック)が指定されます。 0 : クリック 1 : キー押しっぱなし 2 : キーを離す
戻り値	キーボード操作が行えた : TRUE キーボード操作に失敗した : FALSE	
備考		

4.2.19. メッセージボックスの制御

4.2.19.1. MSGBOX コマンド

コマンド名	MSGBOX	
機能	メッセージボックスを表示します。	
書式	MSGBOX(タイトル[String], 表示形式[Integer], メッセージ[String])	
引数の説明	タイトル	メッセージボックスのタイトルを指定します。
	表示形式	メッセージボックスの表示形式を指定します。
	メッセージ	表示させたいメッセージを指定します。
戻り値	メッセージボックスが表示できた : TRUE メッセージボックスが表示できない : FALSE	
備考		

4.2.20. 文字列の制御

4.2.20.1. ASCII コマンド

コマンド名	ASCII
機能	文字を ASCII コードに変換します。
書式	ASCII(値[String])
戻り値	文字コードを変換できた : TRUE 文字コードを変換できない : FALSE
備考	

4.2.20.2. BETWEEN コマンド

コマンド名	BETWEEN	
機能	指定文字列間の指定文字を返します。	
書式	BETWEEN(値[String], 開始文字[String], 終了文字[String], カウント[Integer])	
引数の説明	値	取得したい文字が含まれている文字列を指定します。
	開始文字	取得したい文字の開始文字を指定します。
	終了文字	取得した文字の終了文字を指定します。
	カウント	
戻り値	文字列を取得できた : TRUE 文字列を取得できない : FALSE	
備考		

4.2.20.3. CHAR コマンド

コマンド名	CHAR	
機能	文字コードから文字を返します。	
書式	CHAR(値[Integer])	
引数の説明	値	文字コードを指定します。
戻り値	文字を取得できた : TRUE 文字を取得できない : FALSE	
備考		

4.2.20.4. COPY コマンド

コマンド名	COPY	
機能	指定文字列内の指定箇所をコピーします。	
書式	COPY(指定文字列[String], 開始位置[UInteger], コピー文字数[UInteger])	
引数の説明	指定文字列	コピーしたい文字を含む文字列を指定します。
	開始位置	コピーしたい文字の開始位置を指定します。
	コピー文字数	コピーしたい文字の数を指定します。
戻り値	文字をコピーできた : TRUE 文字のコピーに失敗した : FALSE	
備考		

4.2.20.5. COPYB コマンド

コマンド名	COPYB	
機能	バイト単位で指定文字列の指定箇所をコピーします。	
書式	COPYB(指定文字列[String], 開始位置[UInteger], コピー文字数[Byte])	
引数の説明	指定文字列	コピーしたい文字を含む文字列を指定します。
	開始位置	コピーしたい文字の開始位置を指定します。
	コピー文字数	コピーしたい文字の数を指定します。
戻り値	文字をコピーできた : TRUE 文字のコピーに失敗した : FALSE	
備考		

4.2.20.6. LENGTH コマンド

コマンド名	LENGTH	
機能	文字列や配列の数を調べます。	
書式	LENGTH(値[String])	
引数の説明	値	数を取得したい文字列や配列を指定します。
戻り値	文字列や配列の数を取得できた : TRUE 文字列や配列の数を取得できない : FALSE	
備考		

4.2.20.7. LENGTHB コマンド

コマンド名	LENGTHB	
機能	バイト単位で文字列の数を調べます。	
書式	LENGTHB(値[String])	
引数の説明	値	数を取得したい文字列や配列を指定します。
戻り値	文字列や配列の数を取得できた : TRUE 文字列や配列の数を取得できない : FALSE	
備考		

4.2.20.8. LEFT コマンド

コマンド名	LEFT	
機能	左端から指定数分の文字を取得します。	
書式	LEFT(値[String], 文字数[Integer])	
引数の説明	値	取得したい文字を含んでいる値を指定します。
	文字数	取得したい文字の数を指定します。
戻り値	文字列を取得できた : TRUE 文字列を取得できない : FALSE	
備考		

4.2.20.9. MID コマンド

コマンド名	MID	
機能	開始位置から指定数分の文字を取得します。	
書式	MID(値[String], 開始位置[Integer], 文字数[Integer])	
引数の説明	値	取得したい文字を含んでいる値を指定します。
	開始位置	取得したい文字の開始位置を指定します。
	文字数	取得したい文字の数を指定します。
戻り値	文字列を取得できた : TRUE 文字列を取得できない : FALSE	
備考		

4.2.20.10. RIGHT コマンド

コマンド名	RIGHT	
機能	右端から指定数分の文字を取得します。	
書式	RIGHT(値[String], 文字数[Integer])	
引数の説明	値	取得したい文字を含んでいる値を指定します。
	文字数	取得したい文字の数を指定します。
戻り値	文字列を取得できた : TRUE 文字列を取得できない : FALSE	
備考		

4.2.20.11.REPLACE コマンド

コマンド名	REPLACE	
機能	指定文字列内にある指定文字を置換します。	
書式	REPLACE(値[String], 置換前文字[String], 置換後文字[String])	
引数の説明	値	置換したい文字を含んでいる値を指定します。
	置換前文字	置換する前の文字を指定します。
	置換後文字	置換した後の文字を指定します。
戻り値	文字列の置換に成功した : TRUE 文字列の置換に失敗した : FALSE	
備考		

4.2.20.12. SPLIT コマンド

コマンド名	SPLIT	
機能	文字列を区切り,配列を作成します。	
書式	SPLIT()	
戻り値	配列が作成できた : TRUE 配列を作成できない : FALSE	
備考		

4.2.20.13. STRCONV コマンド

コマンド名	STRCONV
機能	文字列を指定形式に変換します。
書式	STRCONV()
戻り値	指定形式に変換できた : TRUE 指定形式に変換できない : FALSE
備考	

4.2.20.14. TRIM コマンド

コマンド名	TRIM
機能	文字列の両端にある空白や制御文字を削除します。
書式	TRIM()
戻り値	削除できた : TRUE 削除できない : FALSE
備考	

4.2.20.15. VALUE コマンド

コマンド名	VALUE
機能	文字列を数値に変換します。
書式	VALUE()
戻り値	数値に変換できた : TRUE 数値に変換できない : FALSE
備考	

4.2.21. ループ処理

4.2.21.1. FOR コマンド

コマンド名	FOR	
機能	FOR ① TO ② STEP ③ から NEXT の間の処理を、①が②を満たすまでの間ループします。	
書式	FOR(変数[String] = 初期値[String] TO 終了値[String] STEP 刻み値[Integer])	
引数の説明 ※[]で囲っている引数は省略可能です。	変数	処理を繰り返したい間の値が格納されている変数を指定します。
	初期値	処理の開始値を指定します。
	終了値	処理の終了値を指定します。
	[刻み値]	処理のループを行う刻み値を指定します。
戻り値	-	
備考	詳細は 4.4.8. FOR-NEXT コマンド を参照ください。	

4.2.21.2. NEXT コマンド

コマンド名	NEXT
機能	FOR 処理を終了します。
書式	NEXT(変数[String])
戻り値	FOR の処理が行えた : TRUE FOR の処理が失敗した : FALSE
備考	詳細は 4.4.8. FOR-NEXT コマンド を参照ください。

4.2.21.3. REPEAT コマンド

コマンド名	REPEAT
機能	REPEAT から設定された処理を最低 1 回実行します。 UNTIL に設定した条件を評価します。 詳細は 4.2.21.4. UNTIL コマンド を参照ください。
書式	REPEAT
戻り値	-
備考	詳細は 4.4.10. REPEAT-UNTIL コマンド を参照ください。

4.2.21.4. UNTIL コマンド

コマンド名	UNTIL
機能	設定した条件を評価します。 TRUE の場合は終了します。 FALSE の場合は REPEAT～UNTIL 間の処理を繰り返します。
書式	UNTIL(条件式)
戻り値	REPEAT の処理を行えた : TRUE REPEAT の処理が失敗した : FALSE
備考	詳細は 4.4.10. REPEAT-UNTIL コマンド を参照ください。

4.2.21.5. WHILE コマンド

コマンド名	WHILE
機能	WHILE に設定した条件を評価します。 TRUE の間は WHILE-WEND 間の処理を最低 0 回繰り返します。 FALSE の場合は WEND に進みます。
書式	WHILE(条件式)
戻り値	-
備考	詳細は 4.4.9. WHILE-WEND コマンド を参照ください。

4.2.21.6. WEND コマンド

コマンド名	WEND
機能	WHILE 処理を終了します。
書式	WEND
戻り値	WHILE の処理が行えた : TRUE WHILE の処理が失敗した : FALSE
備考	詳細は 4.4.9. WHILE-WEND コマンド を参照ください。

4.2.21.7. CONT コマンド

コマンド名	CONT
機能	FOR, WHILE, REPEAT で行われるループの先頭へ移動します。
書式	CONT
戻り値	-
備考	詳細は 4.4.12. CONT コマンド を参照ください。

4.2.22.その他

4.2.22.1. ABS コマンド

コマンド名	ABS
機能	絶対値を返します。
書式	ABS(値[Integer])
戻り値	絶対値を取得できた : TRUE 絶対値の取得に失敗した : FALSE
備考	

4.2.22.2. CPUUSAGE コマンド

コマンド名	CPUUSAGE
機能	現在の CPU 使用率を取得します。
書式	CPUUSAGE()
戻り値	CPU 使用率を取得できた : TRUE CPU 使用率の取得に失敗した : FALSE
備考	

4.2.22.3. KINDOS コマンド

コマンド名	KINDOS	
機能	指定されたコンピューターの OS 情報を取得します。	
書式	KINDOS(コンピューター名[String], ユーザー名[String], パスワード[String])	
引数の説明 ※[]で囲っている引数は省略可能です。	[コンピューター名]	情報を取得するコンピューター名を指定します。 既定値は自分自身です。
	[ユーザー名]	情報取得時に使用するユーザー名を指定します。 既定値は MagicRPA が利用しているユーザー名です。
	[パスワード]	情報取得時に使用するユーザー名を指定します。 既定値は MagicRPA が利用しているユーザーのパスワードです。
戻り値	取得できた : OS 情報が記載された文字列 取得できない : ""(空文字)	
備考		

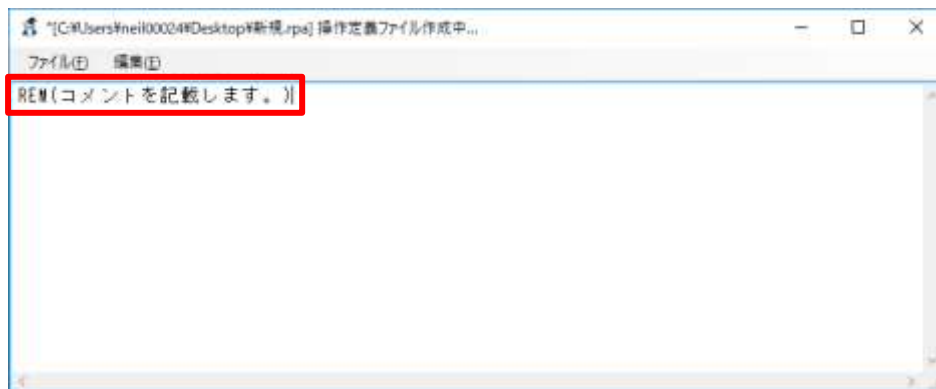
4.2.22.4. MEMUSAGE コマンド

コマンド名	MEMUSAGE
機能	現在のメモリ使用量を取得します。
書式	MEMUSAGE()
戻り値	メモリ使用量が取得できた : TRUE メモリ使用量が取得できない : FALSE
備考	

4.3. プログラムにコメントをする

行の先頭に「REM」を入力します。

例) REM(コメントを記載します。)



4.4. 特殊処理の特徴

4.4.1. DIM コマンド

機能	RPA のローカル変数を宣言します。
書式	DIM(変数名)
例	DIM(hoge = "test")
備考	

4.4.2. PUBLIC コマンド

機能	RPA のグローバル変数を宣言します。
書式	PUBLIC(変数名)
例	PUBLIC(hogehoge = "test2")
備考	

4.4.3. LET コマンド

機能	RPA 変数の演算を行います。
書式	LET(変数名/定数名 + 変数名/定数名 = 変数名/定数名)
例	LET(hogehoge + hoge = test)
備考	

4.4.4. REM コマンド

機能	該当行の処理を行いません。(コメント行となります。)
書式	REM(コメント)
例	REM(ほげほげ)
備考	

4.4.5. CONST コマンド

機能	RPA 定数を宣言します。
書式	CONST(定数名)
例	CONST(hogehogehoge = "test3")
備考	

4.4.6. TRY-ENDTRY コマンド

機能	TRY から EXCEPT の間の処理を実行します。 この間でエラーが発生したら、EXCEPT-ENDTRY 間の処理を実行します。 EXCEPT-ENDTRY 間の処理が完了するか EXIT した場合、エラーが発生した次の行から処理を再開します。
書式	TRY 処理 EXCEPT TRY～EXCEPT 間でエラーが発生した場合の処理 ENDTRY
例	TRY LET(hogehoge + hoge = test) EXCEPT MSGBOX(“エラーが発生しました。”) ENDTRY
備考	

4.4.7. IF-FI コマンド

機能	IF に設定した条件に応じて、IF から ELSE 間、もしくは ELSE から FI 間の処理を実行します。
書式	IF 真の場合の式 真の場合の処理 ELSE 偽の場合の処理 FI
例	IF hogehoge = 0 Then hoge = hoge + 1 ELSE hoge = hoge - 1 FI
備考	

4.4.8. FOR-NEXT コマンド

機能	FOR ① TO ② STEP ③から NEXT の間の処理を、①が②を満たすまでの間ループします。
書式	FOR ①変数 = 初期値 TO ②終了値 {STEP ③刻み値} 処理 NEXT ①変数
例	FOR hoge = 1 TO 10 {STEP 1} hoge = hoge_text.ReadLine() NEXT hoge
備考	

4.4.9. WHILE-WEND コマンド

機能	WHILE に設定した条件を評価し、TRUE の間は WHILE-WEND 間の処理を繰り返します。 (最低 0 回処理実行)
書式	WHILE 条件式 処理 WEND
例	WHILE hoge < 10 hoge = hoge + 1 WEND
備考	

4.4.10. REPEAT-UNTIL コマンド

機能	REPEAT から実行し、最後に UNTIL に設定した条件を評価し、FALSE の場合は REPEAT~UNTIL 間の処理を繰り返します。(最低 1 回処理実行)
書式	REPEAT 処理 UNTIL 条件式
例	REPEAT hoge = hoge + 1 UNTIL hoge = 10
備考	

4.4.11. BREAK コマンド

機能	TRY-ENDTRY,IF-FI,FOR-NEXT,WHILE-WEND,REPEAT-UNTIL を処理している際に、プログラム終了までプログラム処理をスキップ します。 プログラムを再開する際は、FI や NEXT 等処理を終了するコマンド の次の行から行われます。
書式	BREAK
例	IF hogehoge = 1 Then BREAK ELSE hoge = hoge + 1 FI
備考	

4.4.12. CONT コマンド

機能	FOR, WHILE, REPEAT で行われるループの先頭へ移動します。
書式	CONT
例	REPEAT ←CONT でここに戻ります。 hoge = hoge + 1 CONT UNTIL hoge = 10
備考	

4.4.13. QUIT コマンド

機能	プログラムそのものを終了します。
書式	QUIT
例	QUIT
備考	

《補足》 使い方の例は [6.2.参考プログラム](#)を参照ください。

5. 付録

5.1. キーコード一覧

キーコード名	押下するキー
VK_BACKSPACE	BackSpace
VK_TAB	Tab
VK_NUMLOCKOFF_T5	NumLock オフ時のテンキー「5」
VK_ENTER	Enter / テンキーの Enter
VK_SHIFT	Shift
VK_CTRL	Ctrl
VK_ALT	Alt
VK_PAUSE	Pause
VK_HENKAN	変換
VK_MUHENKAN	無変換
VK_SPACE	スペース
VK_PAGEUP	PageUp
VK_PAGEDOWN	PageDown
VK_END	End
VK_HOME	Home
VK_LEFT_ARROW	←
VK_UP_ARROW	↑
VK_RIGHT_ARROW	→
VK_DOWN_ARROW	↓
VK_INSERT	Insert
VK_DELETE	Delete
VK_0	0
VK_1	1
VK_2	2
VK_3	3
VK_4	4
VK_5	5
VK_6	6
VK_7	7
VK_8	8
VK_9	9
VK_A	A
VK_B	B
VK_C	C

VK_D	D
VK_E	E
VK_F	F
VK_G	G
VK_H	H
VK_I	I
VK_J	J
VK_K	K
VK_L	L
VK_M	M
VK_N	N
VK_O	O
VK_P	P
VK_Q	Q
VK_R	R
VK_S	S
VK_T	T
VK_U	U
VK_V	V
VK_W	W
VK_X	X
VK_Y	Y
VK_Z	Z
VK_WINDOWS	Windows キー
VK_APPS	App キー
VK_T_0	NumLock オン時のテンキー「0」
VK_T_1	NumLock オン時のテンキー「1」
VK_T_2	NumLock オン時のテンキー「2」
VK_T_3	NumLock オン時のテンキー「3」
VK_T_4	NumLock オン時のテンキー「4」
VK_T_5	NumLock オン時のテンキー「5」
VK_T_6	NumLock オン時のテンキー「6」
VK_T_7	NumLock オン時のテンキー「7」
VK_T_8	NumLock オン時のテンキー「8」
VK_T_9	NumLock オン時のテンキー「9」
VK_T_ASTERISK	NumLock オン時のテンキー「*」
VK_T_PLUS	NumLock オン時のテンキー「+」
VK_T_MINUS	NumLock オン時のテンキー「-」

VK_T_DOT	NumLock オン時のテンキー「.」
VK_T_SLASH	NumLock オン時のテンキー「/」
VK_F1	F1(ヘルプ)
VK_F2	F2
VK_F3	F3(検索)
VK_F4	F4(アドレスバー)
VK_F5	F5(更新)
VK_F6	F6(フォーカス)
VK_F7	F7
VK_F8	F8
VK_F9	F9
VK_F10	F10(Alt)
VK_F11	F11(全画面)
VK_F12	F12
VK_NUMLOCK	NumLock
VK_SCROLLLOCK	ScrollLock
VK_COLON	: *
VK_SEMICOLON	; +
VK_COMMA	, <
VK_MINUS	- =
VK_DOT	. >
VK_SLASH	/ ?
VK_ATMARK	@ `
VK_BRACKETS_START	[{
VK_BACKSLASH	¥
VK_BRACKETS_END] }
VK_CARET	^ ~
VK_UNDERBAR	¥ _
VK_CAPSLOCK	英数
VK_KATAKANA	カタカナ/ひらがな
VK_ESC	Esc
VK_HANKAKU	半角/全角

6. 商標に関する表記

- ・ Microsoft, Windows, Microsoft .NET Framework は、米国 Microsoft Corporation の米国およびその他の国における登録商標です。
- ・ Windows®の正式名称は Microsoft® Windows® Operating System です。

その他、本マニュアルに記載されている会社名並びに商品名は、それぞれ各社の登録商標です。本マニュアルにおいて、必ずしも商標表示（®、™）を付記していません。