

概略

縦または横方向に表示できる、ボタン型コマンドランチャー

- ボタン押下時の実行内容をスクリプトで詳細に記述できます。
- ボタン押下時の実行内容を、左クリック/右クリック/シフトキー押下時など多様に制御できます。
- ファイルやフォルダをボタンにドロップして、ドロップしたパスを引数としてコマンドを実行できます。
- スクリプトで、条件分岐/ループ/メニューの作成を記述できます。
- 本プログラムのコマンドライン実行時に、スクリプトに記述した項目を実行することができます。
- Altキーを押下しつつプログラムをドロップするとボタンを追加できます。
- スクリプト内で変数を使用できます。またユーザー名やPC名によって制御を分ける事ができます。
- 縦表示の場合に、変数値を切り替えるセレクトを左側に表示することができます。
- セレクトは、フォルダを切り替えて、各フォルダに対して登録ツールを実行するような用途に適しています。

属性

\$Program:	SLauncher
\$Filename:	SLauncher.exe
\$Description:	Selector Launcher.
\$Company:	Nobuyuki Kobayashi
\$Copyright:	(C) Nobuyuki Kobayashi.

履歴

\$Version:
1.00.2018.0919: Created.
1.01.2018.1022: Updated.
1.02.2019.0106: Relased.
1.03.2019.0128: アイコンの遅延読み込み機能を導入。
1.04.2019.0205: Expand, Foreachコマンドを追加。
1.05.2019.0224: ライブラリを更新。
1.06.2019.0630: Stringライブラリを更新。
1.07.2019.1124: 変数セクションを初期化セクションの仕組みに変更。
1.08.2020.0227: ライブラリを更新。
1.09.2020.0427: ライブラリを更新。
1.10.2020.0609: Window変数とPopupWindowコマンドを追加。
1.11.2020.0715: LayerRuleの読み込み構成を変更。
1.13.2020.0810: 公開用にReadMe等を変更。
1.12.2020.0710: デフォルトの文字コードをUTF8に変更。
1.14.2020.0811: 設定ファイル読み込み時の文字コード変換を有効化。
1.15.2020.1019: OpenFolderコマンドでファイルを指定した場合に格納フォルダを開くように変更。
1.16.2020.1214: 設定ファイルのフォントにASCII以外が混じると文字化けする問題を修正。
1.17.2021.1222: MarginとFrameMarginを分けた。
1.18.2021.1224: InputText機能を追加。
1.19.2022.0302: システムメニュー選択によるヘルプ表示を追加。
1.20.2022.0506: ライブラリを更新。
1.21.2022.0808: IconDelayをデフォルト False に変更。
1.22.2022.0901: ツールチップのフォントをWindowFontからMenuFontに変更。Path変数にConfigFileを追加。オプション項目のHideSelectorをViewSelectorに変更。
1.23.2022.0902: Altキーによるボタン追加時の位置をドロップした位置に変更。ボタン追加時のダミー変数をAPPENDに変更。
1.24.2022.0906: IconFileでIconExtensionで指定した拡張子以外のファイルを指定した場合に、Windowsで関連付けられたアイコンを表示するようにした。
1.25.2022.0912: Foreachコマンドのリスト分解方法を変更した。
1.26.2022.0913: AskMessageでボタンに表示する文字列と中断する文字列を指定できるようにした。
1.27.2022.0914: ボタン押下でコマンドを実行する前に、ボタンを非押下状態にするようにした。ボタンの移動を可能にした。
1.28.2023.0527: フォントやアイコンサイズなど表示サイズに関する設定を GuiProfile での設定に変更した。
1.29.2023.0601: タイトルメニューにフォント選択等を追加。マニュアル指定方法を変更。
1.30.2023.0822: アイコン読み込みを遅延させた場合にメニューの再描画を可能にした。デフォルトを遅延読み込みに変更した。
1.31.2023.0927: 読み込んだアイコンをキャッシュファイルに保存可能にした (IconCacheを参照。デフォルトON)。
1.32.2023.1004: アイコン取り込み時の拡大縮小を実装した。
1.33.2023.1101: 画像拡大縮小ロジックを更新。
1.34.2024.0216: セレクト領域に右クリックメニューを追加。セレクト領域で複数選択できるようにして、移動&削除&ソートコマンドを追加/修正した。
1.35.2024.0219: セレクト領域にセバレータ機能を追加。設定ファイルの[WindowParameter]セクションの SeparatorHeightをButtonSeparatorに変更、SelectorSeparator項目を追加。
1.36.2024.0503: コンパイラを更新。ライブラリを更新。デバッグ用にログ記録機能を追加。
1.37.2024.0907: ライブラリを更新。タイトルメニューに言語選択を追加。

コマンドライン指定

次のコマンドライン引数を組み合わせて指定できます。

(未指定)	... 本プログラムを起動します。本プログラムと同じフォルダにある SLauncher.ini が設定ファイルとして使用されます。
-p xxxx	... xxxx を設定ファイルとして本プログラムを起動します。

-a xxxx	... Action定義 xxxx を実行して終了します。ウィンドウは表示しません。
-l xxxx	... Button定義 xxxx をマウスの左ボタン押下時と同様に実行して終了します。ウィンドウは表示しません。
-r xxxx	... Button定義 xxxx をマウスの右ボタン押下時と同様に実行して終了します。ウィンドウは表示しません。
-d xxxx yyyy	... Button定義 xxxx を yyyy がドロップされたのと同様に実行して終了します。ウィンドウは表示しません。

- -d 実行で yyyy の部分に複数のパスを記述した場合は、複数フォルダ/ファイルを選択してまとめてドロップした場合と同じ動作になります。

外観(ウィンドウの説明)

本プログラムでは、横方向表示、縦方向表示、セクタ付き表示、の 3 種類の表示の仕方を選択できます。

横方向表示

設定ファイルの[Option]セクションでHorizontal=Trueとした場合、横方向にボタンを連ねる表示になります。



- Sampleフォルダにある 横表示Sample.ini を本プログラムの exe ファイルにドロップすると、この画像の表示になります。
- 右端にある閉じる（×）ボタンの下側やボタン間のセパレータ部分など、ボタン以外の部分を掴んで（ドラッグして）ウィンドウを移動する事ができます。
- ボタンを左または右クリックすると、そのボタンに関連付けられたコマンドを実行できます。実行するコマンドは設定ファイルに記述します。
- ボタンを編集するためにボタンを選択したい場合は、Altキーを押下しながらクリックします。
- ボタンの 1 項目を選択し、Altキーを押下しながら上下カーソルを押下すると、選択したボタンが上下に移動します。
- ボタンの 1 項目を選択し、DELETEキーを押下すると、選択したボタンが削除されます。
- ボタン登録したいプログラム/ファイル/フォルダをデスクトップ等で選択し、Altキーを押下しながらボタン領域にドロップすると、ドロップした位置にボタンを追加できます。
- ボタンを編集するためのメニューを表示したい場合は、上または下カーソルキーを押下します。
- Altキーを押下しながら右クリックした場合も編集メニューを表示しますが、Windows制限により、右ボタンを離すとメニューが閉じます。

縦方向表示

設定ファイルの[Option]セクションでHorizontal=Falseとした場合、縦方向にボタンを連ねる表示になります。



- Sampleフォルダにある 縦表示Sample.ini を本プログラムの exe ファイルにドロップすると、この画像の表示になります。
- 上端にある閉じる（×）ボタンの左側やボタン間のセパレータ部分など、ボタン以外の部分を掴んで（ドラッグして）ウィンドウを移動する事ができます。
- ボタンを左または右クリックすると、そのボタンに関連付けられたコマンドを実行できます。実行するコマンドは設定ファイルに記述します。
- ボタンを編集するためにボタンを選択したい場合は、Altキーを押下しながらクリックします。
- ボタンの 1 項目を選択し、Altキーを押下しながら上下カーソルを押下すると、選択したボタンが上下に移動します。
- ボタンの 1 項目を選択し、DELETEキーを押下すると、選択したボタンが削除されます。
- ボタン登録したいプログラム/ファイル/フォルダをデスクトップ等で選択し、Altキーを押下しながらボタン領域にドロップすると、ドロップした位置にボタンを追加できます。
- ボタンを編集するためのメニューを表示したい場合は、左または右カーソルキーを押下します。
- Altキーを押下しながら右クリックした場合も編集メニューを表示しますが、Windows制限により、右ボタンを離すとメニューが閉じます。

ボタン領域の編集メニュー項目

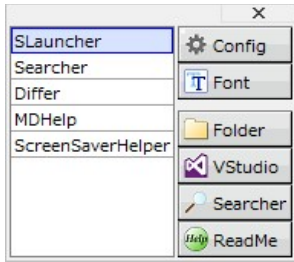
ボタン領域で上/下カーソルキー（横表示）または左/右カーソルキー（縦表示）を押下するか、Altキーを押下しながら右クリックした際に表示されるメニュー項目は下記の通りです。

Insert separator.	... 選択した位置にセパレータを追加します。
Delete selected buttons.	... 選択したボタンを削除します。
Move up selected buttons.	... 選択したボタンを上方に移動します。
Move down selected buttons.	... 選択したボタンを下方に移動します。

- Windows制限により、Altキーを押下しながら右クリックした場合は、マウスの右ボタンを離すとメニューが閉じます。右ボタンを離す前に項目を選択してください。

セクタ付き表示

設定ファイルの[Option]セクションでHorizontal=FalseかつViewSelector=Trueとした場合、縦方向でセクタ付きの表示になります。



- Sampleフォルダにある セレクタ付きSample.ini を本プログラムの exe ファイルにドロップすると、この画像の表示になります。
- セレクタ付き表示は、いくつかのフォルダを切り替えて、各フォルダに対して登録ツールを実行するような用途に適しています。
- セレクタの 1 項目を選択すると、本プログラム内部で Selector 変数に選択した項目(この例ではフォルダのパス)が格納されます。
- Sampleの例では、右側の ReadMe ボタンを押下すると、セレクタで選択したフォルダにある ReadMe_xxxx.txt が MDHelp.exe で表示されるようにスクリプトを記述しています。
- Sampleの例では、右側の VStudio ボタンを押下すると、セレクタで選択したフォルダにある xxxx.sln ファイルがVisualStudioで表示されるようにスクリプトを記述しています。
- セレクタの 1 項目を選択し、Altキーを押下しながら上下カーソルを押下すると、選択したセレクタが上下に移動します。複数選択も可能です。
- セレクタの 1 項目を選択し、DELETEキーを押下すると、選択したセレクタが削除されます。複数選択も可能です。
- セレクタを編集するためのメニューを表示したい場合は、左または右カーソルキーを押下します。
- セレクタ部分に登録したいフォルダをデスクトップ等で選択し、セレクタ領域にドラッグ＆ドロップすると、ドロップした位置にセレクタを追加できます。

セレクタ領域での右クリックメニュー項目

セレクタ領域で左または右カーソルキーを押下するか、マウスを右クリックした際に表示されるメニュー項目は下記の通りです。

Insert separator.	... 選択した位置にセパレータを追加します。
Delete selected items.	... 選択した項目を削除します。
Move up selected items.	... 選択した項目を上方に移動します。複数選択がばらけている場合は、1 番上の選択位置に項目をまとめます。
Move down selected items.	... 選択した項目を下方に移動します。複数選択がばらけている場合は、1 番下の選択位置に項目をまとめます。
Sort selected items.	... 選択した項目を昇順でソートします。複数選択がばらけている場合はソートできません。
Reverse selected selected items.	... 選択した項目を降順でソートします。複数選択がばらけている場合はソートできません。

サンプル設定ファイル共通の説明

Sampleフォルダにある 3 種類の設定ファイルの共通動作について説明します。



- このサンプルでは、左端にあるボタンをマウスの左ボタンでクリックすると、設定ファイルが開きます。
- このサンプルでは、左端にあるボタンをマウスの右ボタンでクリックすると、設定ファイルを再読み込みします。(自身を再起動します)
- このサンプルでは、デスクトップで何らかのファイルを選択し、右端にあるクリップ柄のボタンにドロップすると、ファイルのパスがクリップボードにコピーされます。
- このサンプルでは、Fontと書かれたボタンをマウスの左ボタンでクリックすると、フォント選択のメニューが表示されます。
- 上記のように、左右ボタンでの動作切り替え、ドロップしたパスの利用、メニュー表示などの動作は、設定ファイルに記述されたスクリプトで制御しています。
- 動作の詳細を変更したい場合、設定ファイル中のスクリプトを修正し、再読み込みする事で表示や動作を変更する事ができます。

設定ファイル

本プログラムと同じフォルダにある、<プログラム名>.ini という名称のファイルを設定ファイルとして読み込みます。

共通項目 (Commonセクション)

[Common]	
Program=SLauncher	... SLauncherの設定ファイルである事を指定します。
GuiProfile=xxxx	... 表示に関するパラメータやウィンドウの位置情報などを保存するファイルを、別のファイルから読み込みます。
ConfigFile=xxxx	... 動作設定情報を記述するファイルを、別のファイルから読み込みます。
LanguageFile=xxxx	... GUIの表示言語設定ファイルを指定します。未指定の場合は LanguageFile.ini が使用されます。
Language=xxxx	... LanguageFile内の使用するセクション名を指定します。ユーザー選択設定がある場合はそちらを優先します。

- Common項目のパス指定で相対パスを指定した場合は、プログラムの置かれたフォルダからの相対に置き換えられます。
- Common項目のパス指定内で「\$(User)」を用いた場合は、ユーザの AppData¥Roaming¥プログラム名 に置き換えられます。

動作設定

プログラムの動作に関する設定を指定します。

オプションパラメータの設定 (Optionセクション)

各種オプションを設定します。表示サイズの設定は WindowParameterの項目を参照してください。

[Option]	
LogFolder=xxxx	... 本ツールのログを保管するフォルダを指定します。(デフォルトはTempフォルダの本ツール名サブフォルダのLogフォルダ)
IconCacheFolder=xxxx	... アイコンを保管する場合のフォルダを指定します。(デフォルトはTempフォルダの本ツール名サブフォルダ)
MaxLogFiles=200	... 指定のログフォルダに保管するログの最大数を指定します。0を指定すると無制限になります。(デフォルトは 200)

LogLevel=UCPV	... ログに記録する内容を指定します。U はユーザー操作、C はコマンド実行、P はコマンドパラメータ、V は変数展開をログに記録します。(デフォルトは UCPV)
ViewSelector=False	... Trueを設定すると、ウィンドウの左側のセレクト部分を表示します。(デフォルトはFalse)
Horizontal=False	... Trueを設定すると、ウィンドウが横方向のバーになります。横方向のバーの場合はセレクト部分も非表示になります。(デフォルトはFalse)
ViewTip=False	... Trueを設定すると、ボタンにツールチップを表示します。(デフォルトはFalse)
TopMost=False	... ウィンドウを画面の最上位に固定するかどうかを指定します。(デフォルトはFalse)
SysTray=False	... アイコンをシステムトレイに表示するかどうかを指定します。(デフォルトはFalse)
IconCache=True	... 読み込んだアイコンをファイルに保管するかどうかを指定します。(デフォルトはTrue)
IconDelay=True	... アイコンの遅延読み込みをするかどうかを指定します。(デフォルトはTrue)
IconExtension=ico,exe,dll	... ボタンのIconFile指定で、アイコンを直接取得する拡張子のリスト。(デフォルトはico,exe,dll)
IconUnusedCount=60	... 保管したアイコンがここで指定した起動回数分使用されなかった場合はキャッシュから削除します。(デフォルト60)、
ButtonAlign=Center	... ボタン内のアイコンやテキストの配置。Left, Center, Right が指定できます。(デフォルトはCenter)

- フォルダパス指定で、相対パスで記述した場合は設定ファイル格納フォルダからの相対として扱われます。
- フォルダパス指定で、「\$(User)」を含めた場合は、ユーザの AppData¥Roaming¥本ツール名 に置き換えられます。
- ボタンのIconFile指定が、IconExtensionで指定した拡張子以外の場合で、かつ、IconIndexを指定しなかった場合は、Windowsの関連付けられたアイコンを取得します。
- HDDを使用している場合など、アイコンを各ファイルから読み込むのに時間が掛かり、最初の表示が遅くなる事があります。アイコンの遅延読み込みの設定でこれを回避できます。
- アイコンの遅延読み込みでは、バックグラウンドでアイコンを順次読み込みます。読み込まれる前のアイコンは点線の四角枠のアイコンで表示されます。
- アイコンの保管ファイルは、システムの一時フォルダの本プログラム名のフォルダ内に生成されます。削除された場合は、次の起動時に再作成します。
- IconUnusedCountの判断では、メニューで使用するアイコンはメニュー表示されないと未使用と判断されるため、ある程度大きい数値であるべきです。

ボタンの設定 (Buttonセクション)

各ボタンの表示方法と、ボタンを押下した際のコマンド動作を設定します。Buttonセクションはボタンごとに記述します。

[Button:nnnn]	... nnnnの部分にボタンの定義名を指定します。
Title=xxxx	... ボタンに表示する文字列を指定します。本項目を指定するとボタンに文字列を表示します。
IconFile=xxxx	... アイコンファイルのパスを指定します。本項目を指定するとボタンにアイコンを表示します。
IconIndex=xxxx	... アイコンファイル中のアイコンインデックスを指定します。(デフォルトは 0)
DropRun=False	... Trueを指定した場合、ボタンにファイルやフォルダをドロップした時にコマンドを実行します。(デフォルトはFalse)
Hidden=False	... Trueを指定した場合、ボタンを非表示にします。(デフォルトはFalse)
TipText=xxxx	... ツールチップに表示する文字列を指定します。[Option]のViewTipがTrueの場合に表示されます。本項目を指定しない場合は Title 文字列が使用されます。
コマンド指定	... 実行するコマンドを指定します。複数のコマンドを記述した場合は順に実行します。

[Button:-]	... ボタン間の空白(セパレータ)を指定します。
Hidden=False	... Trueを指定した場合、セパレータを非表示にします。(デフォルトはFalse)

(例1)	
[Button:OpenBase]	... OpenBaseというボタンを作成します。
OpenFolder {	
Parameter=\$(Base:\$(Selector))	
}	

(例2)	
[Button:-]	... ボタン間の空白(セパレータ)を作成します。

- Buttonセクションには、ボタンの表示方法と、ボタンを押下した際のコマンド動作を記述します。
- Buttonセクションは、表示されるボタン 1 個ごとに 1 つのセクションを記述します。
- Buttonセクションが記述された順番に、ウィンドウにボタンが表示されます。
- Buttonセクションのコマンド実行が完了する際、実行時に変更された変数の値が実行前の状態に戻ります。
- TitleもIconFileも指定しなかった場合は、ボタン定義名の nnnn の文字列がボタンに表示されます。
- IconFile に xxx.exe や xxx.ico 以外を指定した場合、Windows でそのファイルに関連付けられたアイコンが表示されます。
- DropRunがTrueの場合、ドロップしたファイルやフォルダの列が変数DropListに、最初にドロップしたファイルやフォルダが変数Dropに格納されます。
- ボタンはシングルクリックで実行されます。ダブルクリックした場合は最初のクリックのみ認識されます。

ウィンドウ操作による新規ボタンの追加

Altキーを押下しながらウィンドウのボタン部分にファイルやフォルダをドロップすると、[Button:*]に記述した内容をテンプレートとして新規ボタンが追加されます。

[Button:*]	... ボタンが追加される際のテンプレートとなる内容を記述します。
:	... 追加される内容
(例1)	
[Button:*]	... ドロップしたファイルやフォルダの名前部分が * の記述箇所に埋め込まれます。
IconFile=\$(APPEND)	... ドロップしたファイルやフォルダのパスが \$(APPEND) の記述箇所に埋め込まれます。
Shell {	
Command=\$(APPEND)	... ドロップしたファイルやフォルダのパスが \$(APPEND) の記述箇所に埋め込まれます。
}	
(例2)	
[Button:NewProgram.exe]	... 例 1 の記述がされている場合に NewProgram.exe をドロップして追加される内容の例
IconFile=C:¥Program Files¥NewProgram¥NewProgram.exe	

```
Shell {
  Command=C:¥Program Files¥NewProgram¥NewProgram.exe
}
```

Bu セクションが未定義の場合は、デフォルトの設定でボタンが追加されます。

ウィンドウ操作によるボタン位置の移動

下記の手順で、ウィンドウ上のボタンの位置を移動できます。

- 1. Altキーを押下しながら特定のボタンを左クリック(もしくは右クリック)すると、ボタンが選択状態になります。
- 2. ボタンが選択状態の時、Altキーを押下しながら上下または左右のカーソルキーを押下すると、ボタンの位置が上下または左右に移動します。
- 3. ボタンが選択状態の時、Altキーを押下しながらDeleteキーを押下すると、ボタンが削除されます。
- 4. マウスをウィンドウ外に動かすと、ボタンの選択状態が解除されます。

- 上記の、特定ボタンが選択状態の間は、ツールチップが非表示になります。(ツールチップがウィンドウのフォーカスを奪うためです)

ウィンドウの左辺に表示されるセレクトアの設定 (Selectorセクション)

各セレクトア項目を選択またはダブルクリックした際のコマンド動作を設定します。Selectorセクションはセレクトア項目ごとに記述します。

[Selector:nnnn] Title=xxxx コマンド指定	... nnnn部分にセレクトア定義名を指定します。 ... セレクトアへの表示名を指定します。Titleを指定しない場合は、セレクトア定義名 nnnn が表示されます。 ... セレクトアを選択またはダブルクリックした際に実行するコマンドを指定します。
(例1) [Selector:SLauncher] Define { Selector=C:¥Home¥SLauncher } If { Condition=\$(IsKey:DClick) OpenFolder { Parameter=\$(Selector) } }	... SLauncherという名前のセレクトア項目をウィンドウに追加します。 ... Selectorを選択した時に、変数に C:¥Home¥SLauncher を格納します。 ... ダブルクリック時に動作するコマンド記述 ... ダブルクリック時にSelectorパスのフォルダを開きます。
(例2) [Selector:-]	... セレクトア領域に空白(セパレータ)を作成します。

- Selectorセクションには、各セレクトア項目を選択またはダブルクリックした際のコマンド動作を記述します。
- Selectorセクションは、表示されるセレクトア 1 項目ごとに 1 つのセクションを記述します。
- Selectorセクションが記述された順番に、ウィンドウにセレクトア項目が表示されます。
- Selectorセクションからのコマンド実行で変数の値を変更した場合、コマンド実行後も変更した値が保持されます (ボタン実行時とは異なります)。
- セレクトア項目をダブルクリックした場合、1クリック目で選択、2クリック目でDClick属性ONでコマンドが起動されます。(変数IsKeyを参照)
- セレクトア項目では、アイコンは表示されません。
- セレクトア領域でCtrlやShiftを押下しながらの選択操作をした場合は、選択解除をした際でもカーソル位置の項目を選択した場合の処理が動作します。

ウィンドウ操作による新規セレクトア項目の追加

Altキーを押下しながらウィンドウのセレクトア部分にファイルやフォルダをドロップすると、[Selector: *]に記述した内容をテンプレートとして新規セレクトア項目が追加されます。

[Selector: *] :	... セレクトア項目が追加される際のテンプレートとなる内容を記述します。 ... 追加される内容
(例1) [Selector: *] Define { Selector=\$(APPEND) }	... ドロップしたファイルやフォルダの名前部分が * の記述箇所に埋め込まれます。 ... ドロップしたファイルやフォルダのパスが \$(APPEND) の記述箇所に埋め込まれます。
(例2) [Selector:SLauncher] Define { Selector=C:¥Home¥SLauncher }	... 例 1 の記述がされている場合に、C:¥Home¥SLauncher をドロップして追加される内容の例

- ウィンドウ左辺のセレクトア部分にパスをドロップすると、[Selector: *]に記述した内容の項目が自動で追加されます。
- 上記の追加されるセレクトア項目は、ドロップしたパスの親フォルダまでを取り除いた部分がセレクトア名になります。
- 上記の追加されるセレクトア項目では、\$(APPEND) と記述した箇所が、ドロップしたパスに置き換えられます。
- ドロップの際にAltキーを押下していた場合は、最下部に項目が追加されます。

Se セクションが未定義の場合は、デフォルトの設定でセレクトア項目が追加されます。

ウィンドウ操作によるセレクトア位置の移動

下記の手順で、ウィンドウ上のセレクトの位置を移動できます。

1. 特定のセレクトを左クリック(もしくは右クリック)すると、セレクトが選択状態になります。
2. セレクトが選択状態の時、Altキーを押下しながら上下のカーソルキーを押下すると、セレクト位置が上下に移動します。
3. セレクトが選択状態の時、Altキーを押下しながらDeleteキーを押下すると、セレクトが削除されます。

変数の記述方法

変数は、よく使う文字列に名前を付けて定義しておく機能です。定義後は、\$(変数名) と記述すれば定義した文字列に置換できます。

```
(例1)
Define {
  ProgramX32=C:¥Program Files (x86)
  Chrome=$(ProgramX32)¥Google¥Chrome¥Application¥chrome.exe
  Edge=$(ProgramX32)¥Microsoft¥Edge¥Application¥msedge.exe
}
```

- 変数は Define コマンドで定義します。例1では、ProgramX32、Chrome、Edge の3つの変数を定義しています。
- Define コマンドはパラメータの各行がそれぞれ1つの変数定義になっていて、「=」の左側に変数名、右側に置換する文字列を記述します。
- 変数引用時は \$(と) の間に変数名を記述します。\$(から) までの部分が定義しておいた文字列に置換されます。
- 例1では、Chrome と Edge 変数の定義で、文字列中の \$(ProgramX32) の部分が、その前に定義した「C:¥Program Files (x86)」に置換されます。
- 変数は、前述の [Common] [Option] [xxxxFont] セクションを除く各セクションで使用できます。

組み込み変数の種類

組み込み変数は下記の通りです。組み込み変数には引数を指定するものがあり、引数はコロン「:」で区切られます。

\$(Not:xxxx)	... xxxxが空文字列か False の場合は True、それ以外の場合は False に置換されます。
\$(Equal:xxxx:yyyy)	... xxxxとyyyyが同じ文字列の場合は True、それ以外の場合は False に置換されます。
\$(EqualNoCase:xxxx:yyyy)	... 英半角の大文字小文字を同一と見做す以外は Equal 変数と同じ動作です。
\$(=:xxxx:yyyy)	... xxxxとyyyyを数値と見做し、等しい場合は True そうでない場合は False に置換されます。
\$(!=:xxxx:yyyy)	... xxxxとyyyyを数値と見做し、等しくない場合は True そうでない場合は False に置換されます。
\$(<:xxxx:yyyy)	... xxxxとyyyyを数値と見做し、xxxxが小さい場合は True、そうでない場合は False に置換されます。
\$(>:xxxx:yyyy)	... xxxxとyyyyを数値と見做し、xxxxが大きい場合は True、そうでない場合は False に置換されます。
\$(<=:xxxx:yyyy)	... xxxxとyyyyを数値と見做し、xxxxが小さいか等しい場合は True、そうでない場合は False に置換されます。
\$(>=:xxxx:yyyy)	... xxxxとyyyyを数値と見做し、xxxxが大きいか等しい場合は True、そうでない場合は False に置換されます。
\$(&&:xxxx:yyyy:zzzz)	... 各引数の論理値の AND 結果に置換されます。引数は複数指定できます。論理値は、空文字列か False の場合は 偽、それ以外は 真 です。
\$(:xxxx:yyyy:zzzz)	... 各引数の論理値の OR 結果に置換されます。引数は複数指定できます。論理値は、空文字列か False の場合は 偽、それ以外は 真 です。
\$(Calc:xxxx)	... xxxxに計算式を指定します。精度は 32bit符号あり整数 で、C/C++の各2項演算子と () が使用できます。 ... (例) \$(Calc:(1+(2+\$(YYY))*4)/3) など。式にエラーがあると、空文字列に置換されます。
\$(Length:xxxx)	... xxxxの文字列長を返します。
\$(Body:xxxx)	... パス名xxxxの末尾の名前部分(拡張子なし)に置換されます。例えば C:¥Home¥Share.txt が引数であれば Share に置換されます。(変数内でのコロンの使用を参照)
\$(BodyEx:xxxx)	... パス名xxxxの末尾の名前部分に置換されます。例えば C:¥Home¥Share.txt が引数であれば Share.txt に置換されます。(変数内でのコロンの使用を参照)
\$(Base:xxxx)	... パス名xxxxの親フォルダのパスに置換されます。例えば C:¥Home¥Share.txt が引数であれば C:¥Home に置換されます。(変数内でのコロンの使用を参照)
\$(Extension:xxxx)	... パス名xxxxの最後の拡張子部分に置換されます。例えば C:¥Home¥Share.txt が引数であれば txt に置換されます。(変数内でのコロンの使用を参照)
\$(Before:xxxx:yyyy)	... 文字列xxxxの先頭からyyyy文字分の文字列に置換されます。
\$(After:xxxx:yyyy)	... 文字列xxxxの先頭からyyyy文字分以降の文字列に置換されます。指定文字数より短い場合は空文字列になります。
\$(File:xxxx)	... xxxxがファイルの場合パスを返し、そうでなければ\$(Missing)に置き換える。xxxxに検索式が使用できます。
\$(Folder:xxxx)	... xxxxがフォルダの場合パスを返し、そうでなければ\$(Missing)に置き換える。xxxxに検索式が使用できます。
\$(Process:xxxx)	... xxxxがプロセス名に合致した場合はプロセス名を返し、なければ\$(Missing)に置き換える。xxxxに検索式が使用できます。
\$(Window:xxxx:yyyy)	... xxxxがクラス名、yyyyがタイトル名の合致するウィンドウがあればハンドル値を返し、なければ\$(Missing)に置き換える。xxxx,yyyyに検索式が使用できます。
\$(FileCount:xxxx)	... xxxxに合致するファイルやフォルダの数に置き換える。xxxxには検索式が使用できます。
\$(FolderCount:xxxx)	... xxxxに合致するファイルやフォルダの数に置き換える。xxxxには検索式が使用できます。
\$(IsDefined:xxxx)	... xxxxで示す変数名の変数が、定義されている場合は True、未定義の場合は False に置換されます。
\$(IsMissing:xxxx)	... xxxxがMissingに定義されている文字列の場合は True、そうでない場合は False に置換されます。
\$(Cond:xxxx:yyyy:zzzz)	... xxxxの値が False でも 空文字列 でもない場合は yyyy、False か 空文字列の場合は zzzz に置換されます。
\$(CondDef:xxxx:yyyy:zzzz)	... xxxxで示す変数名の変数が、定義されている場合は yyyy、未定義の場合は zzzz に置換されます。
\$(ExitCode)	... ExitCodeには、Shellコマンドで実行した際のExitCodeが格納されています。終了を待ち合わせない場合は途中結果になります。
\$(Missing)	... File,Folder,Process,Window変数で使用するエラー時変換文字列。デフォルトでは空文字列が格納されています。
\$(Result)	... AskMessageコマンドで、ユーザの選択したYes/No結果がTrueかFalseとして格納されています。
\$(Pickup)	... Pickupコマンド、PickupMenuコマンドで選択した値が格納されています。
\$(Counter)	... Forコマンドのループ内で、Begin～End間の現在値が格納されています。
\$(Drop)	... ボタンにドロップした最初のファイルかフォルダのパスが格納されています。(二重引用符なし)
\$(DropList)	... ボタンにドロップした複数のファイルかフォルダのパスの列が格納されています。(それぞれ二重引用符を付加)
\$(IsKey:xxxx)	... ボタンからのコマンド起動時のマウスやシフト状態を調べる。引数で指定した状態がONの場合はTrueに置換されます。 ... 複数の引数を指定でき、複数指定した場合は各引数の結果のANDが全体の結果となります。

	... 引数は LButton, RButton, Button, LShift, RShift, Shift, LControl, RControl, Control, DClick, Dropのいずれかで す。
	... 引数の先頭文字が L か R の場合は、左 と 右 のキーを表し、L R のない場合は 左右のキーのどちらかのONを表します。
	... RETURNキーやコマンドライン呼び出しの場合は、マウスのLButtonがONになります。
\$(KeyState)	... ボタンからのコマンド起動時のマウスやシフト状態のリストが格納されています。上記 IsKey で使用されます。
\$(Clipboard)	... クリップボードテキストに置換されます。クリップボードにテキストが格納されていない場合は空文字列となります。
\$(Code:xxxx)	... 数値xxxxを文字コードとして1バイトの文字に置換します。
\$(StripQuote:xxxx)	... 文字列xxxxの両端の二重引用符を取り除きます。
\$(Replace:xxxx:yyyy:zzzz)	... 文字列xxxxに含まれる部分文字列yyyyをすべて文字列zzzzに置換します。(zzzzが空の場合は削除になります)
\$(Path:xxxx)	... 特殊なフォルダパスや名称に置き換わります。
	... xxxxが Desktop の場合は、デスクトップフォルダに置換されます。
	... xxxxが Temp の場合は、テンポラリフォルダに置換されます。
	... xxxxが User の場合は、ユーザの AppData¥Roaming に置換されます。
	... xxxxが HostName の場合は、PC名に置換されます。
	... xxxxが UserName の場合は、ユーザ名に置換されます。
	... xxxxが Launcher の場合は、SLauncher.exeのパスに置換されます。
	... xxxxが ConfigFile の場合は、読み込んでいる設定ファイルのパスに置換されます。

- 文字列を数値として扱う場合、10進数表記以外に、16進数(0xFFFE等)と2進数(0b1010等)での表記が使用できます。
- 本プログラムの論理判定は、文字列が、空文字列か False の場合は「偽」、それ以外の場合は「真」です。

変数内でのコロンの使用

コロンは変数記述内では引数の区切り文字として扱われるため、コロンを含む文字列を引数に指定できませんが、下記の例のような指定方法で回避できます。

(例1)	
Define {	
Work-C=C:¥Work	
Work-D=D:¥Work	
WorkFolder=\$(Cond:\$(EqualNoCase:\$(Path:HostName):MyHost):\$(Work-C):(Work-D))	
}	
(例2)	
Define {	
Collon=:	
WorkFolder=\$(Cond:\$(EqualNoCase:\$(Path:HostName):MyHost):C\$(Collon)¥Work:D\$(Collon)¥Work)	
}	

- 例1、例2共に、MyHostというPCでは WorkFolder 変数を C:¥Work と定義し、その他のPCでは D:¥Work と定義する例です。

変数の展開

変数は定義時 (Defineコマンド) には展開されず、使用時 (引用時) に展開されます。

(例1)	
Define {	
Temp1=XXX	
Temp2=\$(Temp1)	
}	
Define {	
Temp1=YYY	
}	
PutMessage {	
Parameter=\$(Temp2)	... 表示されるのは YYY になります。
}	

Expandコマンドでの変数の展開

Expandコマンドを用いると変数を事前展開する事ができます。下記は、Testボタンにドロップしたファイルに対してメニュー項目を作成する例です。

(例1)	
[Button:Test]	
DropRun=True	
Menu {	... Testボタンを最初に実行した時にメニューが作成されます。
Foreach {	
Parameter=\$(DropList)	... Testボタンを最初に実行した時にドロップしたファイルのリストが渡されます。
MenuItem {	
Title=\$(Body:\$(Each))	... Testボタンを最初に実行した時にドロップしたファイルの名前でメニュー項目が作成されます。
OpenFile {	
Parameter=\$(Each)	... 表示されるメニューを選択した時に変数が展開されるため、2 回目以降は空になります。
}	
}	
}	
}	

}	
<div> <div>(例2)</div> <div>[Button:Test]</div> <div>DropRun=True</div> <div>Menu {</div> <div> <div>Foreach {</div> <div> <div>Parameter=\$(DropList)</div> <div>Expand {</div> <div>File=\$(Each)</div> <div>MenuItem {</div> <div>Title=\$(Body:\$(Each))</div> <div>OpenFile {</div> <div>Parameter=\$(File)</div> <div>}</div> <div>}</div> <div>}</div> <div>}</div> <div>}</div> <div>}</div> </div> </div> </div>	
...	Testボタンを最初に実行した時にメニューが作成されます。
...	Testボタンを最初に実行した時にドロップしたファイルのリストが渡されます。
...	変数 File については Expand 記述内で事前展開することを指定している。
...	Testボタンを最初に実行した時にドロップしたファイルの名前でメニュー項目が作成されます。
...	メニューを作成した時に\$(File)が展開されるので、表示されるメニューを選択した時にファイル名が渡されます。

- 例1/例2は、メニューが最初の実行時に作成される仕様のため、最初にドロップした個々のファイルに対してメニュー項目が作成されます。
- 例1では、2回目以降のTestボタン押下で表示されたメニューで項目を選択した際、OpenFile のパラメータの \$(Each) がその時点で空なので、動作しません。
- 例2では、OpenFile のパラメータの \$(File) が、Expandコマンド呼び出し時にファイル名に事前展開されるため、メニュー表示からの選択でも動作します。

よく使うコマンド記述に名前を付けて定義しておくための設定 (Actionセクション)

Actionセクションでは、よく使うコマンド記述に、名前を付けて定義します。

<div> <div>[Action:nnnn]</div> <div>コマンド指定</div> <div>:</div> </div>	
...	Action名 nnnn を指定します。
...	実行するコマンドを指定します。複数のコマンドを連続して記述できます。
<div> <div>(例1)</div> <div>[Action:IE]</div> <div>Shell {</div> <div> <div>Command=C:¥Program Files¥Internet Explorer¥iexplore.exe</div> <div>Directory=\$(Path)</div> </div> <div>}</div> </div>	
...	IEという名前のActionセクションの定義例です。
<div> <div>(例2)</div> <div>[Button:MyHelp]</div> <div>Action {</div> <div> <div>Name=IE</div> <div>Path=¥¥129.249.0.10¥Common¥MyHelp.html</div> </div> <div>}</div> </div>	
...	例1のActionを呼び出すボタン定義の例です。
...	呼び出すAction定義名を指定します。以降は呼び出す際の変数設定になります。

- Actionセクションで定義したコマンド記述は、Actionコマンドによって呼び出して実行できます。
- Actionセクションは複数記述することができます。
- ActionコマンドのName/パラメータで、呼び出すActionセクション名を指定します。
- ActionコマンドのName/パラメータ以外のパラメータは、Actionセクションを呼び出す際の変数設定になります。(Defineと同様の効果)

初期化の設定 (Initializeセクション)

Initializeセクションは、本プログラム起動時に 1 度だけ実行されます。広範囲に使用する変数を事前定義しておく用途などに使用します。

<div> <div>[Initialize]</div> <div>コマンド指定</div> </div>	
...	実行するコマンドを指定します。複数のコマンドを連続して記述できます。
<div> <div>(例1)</div> <div>[Initialize]</div> <div>Define {</div> <div> <div>ProgramX64=C:¥Program Files</div> <div>IE=\$(ProgramX64)¥Internet Explorer¥iexplore.exe</div> </div> <div>}</div> </div>	

- Initializeセクションは 1 つだけ記述することができます。
- Initializeセクションは本プログラムのウィンドウ表示より前に実行されます。

ヘルプ (マニュアル)表示の指定 (Manual セクション)

タイトルメニューの「View help.」または「Version: バージョン番号」項目を選択した際のマニュアル表示方法を指定します。

<div> <div>[Manual]</div> <div>Manual=xxxx,yyyy</div> </div>	
...	マニュアルファイルを指定します。カンマ区切りで複数のファイルを指定可能です。ファイル名部分には * と ? が使用できます。

- MDHelp=txt,md

... MDHelpを使って表示する拡張子を指定します。デフォルトは txt と md です。その他の拡張子は Windows 設定に従います。
- 「Version: バージョン番号」項目を選択した場合は、ReadMe_<本プログラム名>.txt を表示します。

■ 「Version: バージョン番号」項目を選択した場合で、[Common]のLanguage項目を設定している場合は、ReadMe_<本プログラム名>_<言語名>.txt を先に検索します。

■ 「View help.」項目では、複数のマニュアルが合致する場合に、サブメニュー形式でリストアップします。

■ マニュアルファイルの指定では、プログラムフォルダからの相対指定も可能です。

■ MDHelp.exeは、本プログラムのフォルダ、または MDHelpサブフォルダの下、または本プログラムフォルダと同じ並びのMDHelpフォルダの下、から検索します。

コマンドの記述方法

コマンドの記述方法は下記の通りです。複数のコマンドを連続実行する場合は、次の行に続けてコマンドを記述する事ができます。

コマンド名	... パラメータのないコマンドの記述文法です。
コマンド名 { 引数=xxxx }	... パラメータのあるコマンドの記述文法です。 ... パラメータは { } ブロック内に複数行記述できます。
(例1) QuitLauncher	... パラメータのないコマンドの記述例です。
(例2) Action { Name=IE }	... パラメータのあるコマンドの記述例です。 ... Actionコマンドの Name パラメータの記述です。

組み込みコマンドの種類

組み込みコマンドには、下記のものがあります。

QuitLauncher	... SLauncherを終了します。
RebootLauncher	... SLauncherを再起動します。
AbortExecution	... その時点の実行を中止します。
Define { 変数名=置換後の文字列 : }	... 変数を定義します。 ... \$(変数名)の部分を置き換える文字列を指定します。 ... 複数指定できます。
Expand { 変数名=置換後の文字列 : コマンド記述 : }	... パラメータで指定した変数についてのみ、指定したコマンド記述内で事前に展開します。 ... \$(変数名)の部分を事前に置き換える文字列を指定します。 ... 複数指定できます。 ... 対象となるコマンド記述です。 ... 複数指定できます。
Action { Name=xxxx 変数名=置換後の文字列 : }	... Actionセクションで定義したアクションを呼び出して実行します。 ... 実行するアクション名を指定します。 ... アクションを実行する前に変数を定義できます。\$(変数名)の部分を置き換える文字列を指定します。 ... 複数指定できます。
Menu { コマンド記述 : }	... メニューを構築し表示します。最初の実行時にコマンド記述を実行してメニューを構築します。2回目以降の実行時はコマンド記述を実行しません。 ... メニューを構築するコマンド記述です。Menu (サブメニュー)、MenuItem、Separator コマンドで要素を作成しますが、それ以外のコマンドも使用できます。 ... 複数指定できます。
Menu { Title=xxxx IconFile=xxxx IconIndex=xxxx コマンド記述 : }	... Menuコマンドのブロック内で記述した場合は、サブメニュー項目を作成します。 ... メニューに表示する項目名を指定します。 ... 項目名の右側にアイコンを表示する場合は、アイコンファイルのパスを指定します。 ... アイコンファイル中のアイコンインデックスを指定します。(デフォルトは 0) ... サブメニューを構築するコマンド記述です。 ... 複数指定できます。
MenuItem { Title=xxxx IconFile=xxxx	... Menuコマンドのブロック内で記述した場合は、メニュー項目を作成します。 ... メニューに表示する項目名を指定します。 ... 項目名の右側にアイコンを表示する場合は、アイコンファイルのパスを指定します。

IconIndex=xxxx コマンド指定 : }	... アイコンファイル中のアイコンインデックスを指定します。(デフォルトは 0) ... メニュー項目を選択した場合に実行されるコマンドを記述します。 ... 複数指定できます。
Separator	... Menuコマンドのブロック内で記述した場合は、メニューのセパレータを作成します。
OpenFolder { Parameter=xxxx }	... フォルダを開きます。ファイルを指定した場合は、そのファイルの置かれたフォルダを開きます。 ... フォルダのパスを指定します。
OpenFile { Parameter=xxxx }	... ファイルを開きます(起動します)。フォルダを指定した場合はフォルダを開きます。 ... ファイルのパスを指定します。
Pickup { Parameter=xxxx xxxx1=zzzz1 : }	... Pickup 変数に格納する文字列を、Parameterで指定される文字列によって切り替えます。 ... 他のパラメータのパラメータが xxxx で指定した文字列と一致するものを選びます。xxxx に変数を含む使用方法を想定しています。 ... xxxx が xxxx1 と同一の場合に、Pickup 変数に zzzz1 が格納されます。 ... 複数の選択肢が記述できます。
PickupMenu { xxxx1=zzzz : Separator }	... メニューを表示し、Pickup 変数に選択される項目の値となる文字列を格納します。 ... xxxx1をメニュー項目として表示します。この項目を選択した場合は Pickup 変数に zzzz1 が格納されます。 ... 複数のメニュー項目が記述できます。 ... セパレータを追加する場合の記述です。
Shell { Command=xxxx Parameter=xxxx Directory=xxxx ShowStatus=xxxx Verb=xxxx Wait=xxxx Timeout=xxxx }	... 外部ツールを実行します。 ... ツールのパスを指定します。 ... ツール起動時のパラメータを指定します。 ... ツール起動時のワークフォルダを指定します。指定しない場合は Windows 仕様に従います。 ... 起動時のツールの状態として Normal, Maximize, Minimize, Hide のいずれかを指定します。指定しない場合は Windows 仕様に従います。 ... open, print, properties, find, edit, explore、runas 等が指定できます。それぞれの動作は Windows 仕様に従います。指定しない場合も Windows 仕様に従います。 ... WaitForDone か WaitForIdle が使用できます。指定しない場合は、ツールを起動したら Shell コマンドを終了します (待ち合わせをしません)。 ... WaitForDone は起動したツールの終了まで待ち合わせます。WaitForIdle は起動したツールが Idle 状態になるまで待ち合わせます。 ... Waitパラメータを指定した場合のタイムアウト時間(ミリ秒)を指定します。(デフォルトは 5000 ミリ秒)
PopupWindow { Parameter=xxxx }	... ウィンドウが最小化されていれば開きます。 ... ウィンドウのハンドル値を指定します。Window 変数を参照してください。
ToClipboard { Parameter=xxxx }	... クリップボードにテキストを格納します。 ... クリップボードに格納する文字列を指定します。
PutMessage { Parameter=xxxx }	... メッセージを表示します。 ... 表示するメッセージを指定します。
AskMessage { Parameter=xxxx Button=xxxx AbortBy=xxxx }	... メッセージを表示しボタンの選択肢をユーザーに選択させます。選択した結果が Result 変数に格納されます。 ... 表示するメッセージを指定します。 ... ボタンの選択肢をカンマからスラッシュ区切りのリストで指定します。(デフォルトは Yes,No) ... ユーザーが選択したボタンと、AbortByで指定した文字列が一致した場合は、スクリプトの実行を中断します。
InputText { Parameter=xxxx NotAbort=xxxx 0:Variable=xxxx 0:Label=xxxx 0:Initial=xxxx 0:Hints=xxxx,yyyy 0:NoHint=True 0:ReadOnly=True	... メッセージを表示し文字列を入力させます。入力結果は指定した各変数と、Finish選択時は True、Cancel選択時は False が Result 変数に格納されます。 ... 表示するメッセージを指定します。 ... ユーザーが Cancel を選択した場合に、スクリプト全体の実行を中止しない場合は True を指定します。 ... 1 つ目のボックスで入力した文字列を格納する変数名を指定します。必ず指定する必要があります。 ... 1 つ目のボックスの左側に表示するラベル文字列を指定します。必ず指定する必要があります。 ... 1 つ目のボックスの初期状態で入力されている文字列を指定します。指定しない場合は初期状態では空になります。 ... 1 つ目のボックスのヒント文字列をカンマ区切りで複数指定します。個々の項目を二重引用符で括弧することができます。 ... 1 つ目のボックスのヒントのドロップダウンを表示しないようにできます。NoHint も Hints も指定しない場合は、過去の選択結果をヒント表示します。 ... 1 つ目のボックスを編集不可にできます。ReadOnly でもヒント選択によって変更する事はできます。

1:Variable=xxxx : }	... 2つ目のボックスで入力した文字列を格納する変数名を指定します。指定しない場合は、2つ目のボックス以降は表示されません。 ... 1つ目のボックスのパラメータと同様に、2つ目以降のボックスのパラメータを指定できます。ボックスは最大20個まで作成できます。
If { Condition=xxxx コマンド記述 : }	... 条件によってコマンドを実行するかどうかを切り替えます。 ... xxxxの部分が「真」の場合に (空文字列でも False でもなければ)、ブロック内のコマンド記述を実行します。 ... 実行するコマンドを記述します。 ... 複数指定できます。
Else { コマンド記述 : }	... 直前のIfの条件が「偽」の場合に、ブロック内のコマンド記述を実行します。 ... 実行するコマンドを記述します。 ... 複数指定できます。
While { Condition=xxxx コマンド記述 : }	... Conditionで指定した条件に従い、ブロック内のコマンド記述を繰り返し実行します。ループ中にESCAPEキーを押下すると、強制的に実行を停止します。 ... xxxxの部分が「真」の場合に (空文字列でも False でもなければ)、ブロック内のコマンド記述を繰り返し実行します。 ... 実行するコマンドを記述します。 ... 複数指定できます。
For { Begin=xxxx End=yyyy Add=zzzz コマンド記述 : }	... Begin,End,Addの条件に従い、ブロック内のコマンド記述を繰り返し実行します。ループ中にESCAPEキーを押下すると、強制的に実行を停止します。 ... xxxxからyyyyになるまで増分zzzzごとに、コマンド記述を繰り返し実行します。xxxx,yyyy,zzzzは32ビットの整数値(-32768～32767)です。 ... Begin,End,Addの条件で決定される現在値は Counter 変数に格納されます。 ... 実行するコマンドを記述します。 ... 複数指定できます。
Foreach { Parameter=xxxx コマンド記述 : }	... Parameterで指定した各文字列ごとに、ブロック内のコマンド記述を実行します。ループ中にESCAPEキーを押下すると、強制的に実行を停止します。 ... カンマ区切りで複数の文字列を指定します。指定した各文字列ごとに、文字列を Each 変数に格納して、コマンド記述を繰り返し実行します。 ... 各文字列はそれぞれ二重引用符付きでの記述も可能です。 ... 実行するコマンドを記述します。 ... 複数指定できます。
Break	... While、For、Foreachブロックを抜けます。

Gui関連設定

ウィンドウ表示に関するパラメータを指定します。プログラムによって書き換えられるため設定ファイルと分離した方が良いです。

ユーザー選択設定

ユーザーが選択した内容を保存します。

[UserSelection] Language=xxxx	... タイトルメニューの Select language 項目で選択した設定を保存します。xxxx が空の場合は自動選択となります。
----------------------------------	---

- LanguageFile が存在しない場合は、タイトルメニューの Select language 項目は選択不可となります。

オプションパラメータの設定 (WindowParameterセクション)

表示に関する各種パラメータを設定します。

[WindowParameter] Margin=4 FrameMargin=2 CaptionHeight=12 ButtonSeparator=7 SelectorSeparator=5 ButtonIconSize=16 MenuIconSize=16 BaseIconSize=32	... フォントダイアログやテキスト表示ダイアログのアイテム間やボーダーの幅を指定します。(デフォルトは 4) ... ウィンドウのボーダーの幅を指定します。(デフォルトは 2) ... ウィンドウ上部の空間の幅を指定します。(デフォルトは 12) ... ボタンのセパレータの高さ(幅)を指定します。(デフォルトは 7) ... セレクターのセパレータの高さを指定します。(デフォルトは 5) ... アイコンボタンのアイコンのサイズを指定します。(デフォルトは16) ... メニューのアイコンのサイズを指定します。(デフォルトは16) ... 内部でキャッシュするアイコンの基準サイズを指定します。(デフォルトはWindows設定と同じ)
---	---

ウィンドウの表示に使用するフォントの設定 (WindowFontセクション)

ウィンドウの表示に使用するフォントを設定します。タイトルメニューの「Select window font.」項目から変更できます。

```
[WindowFont]
Height=12
Width=0
Escapement=0
Orientation=0
Weight=0
Italic=0
Underline=0
StrikeOut=0
Charset=1
OutPrecision=0
ClipPrecision=0
Quality=0
PitchFamily=0
FaceName=xxxx
```

メニューの表示に使用するフォントの設定 (MenuFontセクション)

メニューの表示に使用するフォントを設定します。タイトルメニューの「Select menu font.」項目から変更できます。

```
[MenuFont]
:
... パラメータはWindowFontと同じです。
```

ツールチップの表示に使用するフォントの設定 (TipFontセクション)

ツールチップの表示に使用するフォントを設定します。タイトルメニューの「Select tip font.」項目から変更できます。

```
[TipFont]
:
... パラメータはWindowFontと同じです。
```

ダイアログの表示位置の記録

ダイアログの表示位置を保存します。

```
[MainDialog]
WindowRect_L=133      ... ダイアログの左端座標です。
WindowRect_T=373      ... ダイアログの上端座標です。
WindowRect_R=1217     ... ダイアログの右端座標です。
WindowRect_B=976      ... ダイアログの下端座標です。
```

End
