

検索支援ツール「かんざし」 AHK 版 v0.4 カスタマイズガイド

2023 年 8 月

この文書では、かんざし AHK 版 v0.4 の検索先や動作をカスタマイズする方法について説明します。
デフォルト状態のかんざしの操作を試したうえでご覧ください。

かなり長い文書ですが、すべてをお読みいただく必要はありません。目的に応じて、次のようにご利用ください。

- ◆ 検索先の追加やショートカットキーの変更などの基本的な修正については、「[1 基本的なカスタマイズ](#)」で説明しています。カスタマイズを初めて行う方は、まずはこのセクションに一通り目を通すことをお勧めします。
- ◆ それ以外の細かなカスタマイズが必要な場合は、「[2 目的別設定リファレンス](#)」を参照して、目的に合った箇所をお読みください。

なお、かんざしの実行中に Kanzasi.ahk を修正したときには、タスクトレイでかんざしのアイコンを右クリックし、「スクリプトの更新を反映」をクリックしてください。(これを行わないと修正が動作に反映されませんのでご注意ください)。

※スクリプトに手を加えるときには、修正を誤ると正しく動作しなくなる可能性があります。すぐに元に戻せるよう、Kanzasi.ahk のバックアップをとってから作業に入ることをお勧めします。

1 基本的なカスタマイズ

ここでは、デフォルト状態の Kanzasi.ahk に対する基本的なカスタマイズの方法として、以下の 6 つを説明します。

- ◆ [かんざしの入力ウィンドウやポップアップメニューを表示するためのホットキーを変更する方法](#)
- ◆ [検索先選択のショートカットキーや一発検索のホットキーを変更する方法](#)
- ◆ [新たな検索先として Web サイトを追加する方法](#)
- ◆ [新たな検索先として Windows 上の辞書ソフトを追加する方法](#)
- ◆ [Web サイトや辞書ソフトを串刺し検索する方法](#)
- ◆ [独自の検索処理を記述する方法](#)

かんざしの入力ウィンドウやポップアップメニューを表示するためのホットキーを変更する

かんざしの入力ウィンドウやポップアップを表示するためのホットキーは、Kanzasi.ahk の下記の部分で指定できます。(「InputBoxActivate」で検索すると見つかります)。

```
^+k::InputBoxActivate() ; 入力ウィンドウを表示
^+d::InputBoxActivate(SelGet(false)) ; 選択語句を取り込んで入力ウィンドウを表示
^+i::PopupMenuActivate(1, " ") ; 前回と同じ語句を検索するポップアップメニューをカーソル位置に表示
^+e::PopupMenuActivate(1) ; 選択語句を検索するポップアップメニューをカーソル位置に表示
MButton::PopupMenuActivate(0, , "{MButton}") ; 選択語句を検索するポップアップメニューをマウス位置に表示
!^MButton::KanzasiInfoGet() ; スクリプト修正用の情報を取得して表示
^+n::NextDictActivate(1) ; 次の辞書をアクティブ化
^+p::NextDictActivate(-1) ; 前の辞書をアクティブ化
```

キーを変更するには、上記の青色部分を修正してください。キーの指定方法は AHK の一般的な記述方法と同じです。詳しくは[こちら](#)をご覧ください。

検索先選択のショートカットキーや一発検索のホットキーを指定する

かんざしの入力ウィンドウで各検索先を選択するためのショートカットキーや、ホットキー一発検索で使用するホットキーは、Kanzasi.ahk の下記の部分で指定できます。(「ini_grpid」で検索すると見つかります)。

```
ini_grpid := { 3: "Google 日本      = (^g), ((^+g))"
              , 6: "Google 米国      = (^u), ((^+u))"
              , 9: "Google 英国      = (^k)"
              , 12: "Google 米画像    = (^i)"
              , 15: "Bing            = (^b)"
              , 18: "-----"
              , 21: "EBWin4          = (^q), ((^+q))"
              , 24: "-----"
              , ... (中略)
              , 57: "Wiktionary      = (^w)"
              , 60: "M-W            = (^s)"
              , 63: "AHD            = (^h)"
              , 66: "-----"
              , 69: "ロングマン英和  = (^;) )"
              , ... (以下略)
```

() 囲みの部分(青色マーカーの部分)がショートカットキー、(()) 囲みの部分(緑色マーカーの部分)がホットキーです。

※ショートカットキーは () 囲み、ホットキーは (()) 囲みで指定します。ホットキーは Windows 全体で幅広く有効なキーだからカッコ (()) も幅広い、と覚えるとよいかもしれません。

このカッコ囲みの表記法は、かんざしで独自に定めたもので、この定義部分でのみ有効です。スクリプト内の他の部分や、一般的な AHK スクリプトでは使えませんのでご注意ください。

キーを別のものに変更するには、() 囲みや (()) 囲みの中のキー指定を変更してください。キーの指定の仕方は AHK の一般的な記述方法と同じです。詳しくは[こちら](#)をご覧ください。

キーの追加や削除も可能です。たとえば、先ほどのコードで Bing の行を次のように変更すると、Ctrl + Shift + B というホットキーで Bing の一発検索が行えるようになります。

```
, 15: "Bing            = (^b), ((^+b))"
```

新たな検索先として Web サイトを追加する

新たな検索先を追加するための作業は、大きく3つに分かれます。

1. 検索先の名前、ショートカットキー、ホットキーを指定する
2. 検索処理を記述するための「枠組み」を作成する
3. 検索処理を記述する

ここでは例として、辞書検索サイト「Onelook Dictionary Search」(<https://onelook.com/>)を新たな検索先として追加する場合を想定して、この3つの作業を説明していきます。

1. 検索先の名前、ショートカットキー、ホットキーを指定する

Kanzasi.ahk の中に次のような箇所があります。(ファイル内を「ini_grpid」で検索すると見つかります)。

```
ini_grpid := { 3: "Google 日本      = (^g), ((^+g))"
              , 6: "Google 米国      = (^u), ((^+u))"
              , 9: "Google 英国      = (^k)"
              , 12: "Google 米画像     = (^i)"
              , 15: "Bing             = (^b)"
              , 18: "-----"
              , 21: "EBWin4          = (^q), ((^+q))"
              , 24: "-----"
              , …(中略)
              , 57: "Wiktionary      = (^w)"
              , 60: "M-W             = (^m)"
              , 63: "AHD             = (^h)"
              , 66: "-----"
              , 69: "ロングマン英和   = (^;)"
              , …(以下略)
```

ここが、検索先の名前とキーを指定している部分です。1行が1項目で、各行は「検索先 ID: "検索先名 = (ショートカットキー), ((ホットキー))"という形式になっています。

新たな検索先も同じ形式で追加します。この例では、次の青色部分のように記述してみましょう。

```
ini_grpid := { 3: "Google 日本      = (^g), ((^+g))"
              , 6: "Google 米国      = (^u), ((^+u))"
              , 9: "Google 英国      = (^k)"
              , 12: "Google 米画像     = (^i)"
              , 15: "Bing             = (^b)"
              , 18: "-----"
              , 21: "EBWin4          = (^q), ((^+q))"
              , 24: "-----"
              , …(中略)
              , 57: "Wiktionary      = (^w)"
              , 60: "M-W             = (^m)"
              , 63: "AHD             = (^h)"
              , 64: "Onelook         = (^l), ((^+l))"
              , 66: "-----"
              , …(以下略)
```

この行で記述した各項目の意味は次のとおりです。

- ◆ 「64」: 検索先の ID 番号。既存の番号と重複しない空き番号を指定してください。番号の並びが小さい順のままとなるような空き番号を選ぶことをお勧めします。また、ここで指定した値は後ほど使いますので、覚えておいてください。
- ◆ 「Onelook」: 検索先の名前。かんざしの入力ウィンドウやポップアップメニューでは、検索先としてこの名前が表示されます。
- ◆ 「(^1)」: ショートカットキー。かんざしの入力ウィンドウで、この検索先を選択するためのキーです。この例で記述した ^1 とは「Ctrl + 1」という意味です。
- ◆ 「((^+1))」: 一発検索のホットキー。選択語句をホットキーのみで検索するときに使用するキーです。この例で記述した ^+1 とは「Ctrl + Shift + 1」という意味です。

※^1 や ^+1 のようなキーの指定方法は、AHK の標準の指定方法と同じです。詳しくは[こちら](#)をご覧ください。

※検索先の名前と、その右のほうに出てくる「=」の間には、多数のスペースが入っていますが、このスペースの数はいくつでもかまいません。(デフォルトでは、各行の「=」の位置が揃うようにスペースの数を調整していますが、別に揃っていないなくても問題ありません)。

※ショートカットキーやホットキーが不要の場合は、「, 64: "Onelook"」のように、ID 番号と名前だけの指定でもかまいません。

2. 検索処理を記述するための「枠組み」を作成する

Kanzasi.ahk の中に次のような箇所があります。(ファイル内を「DictGrp」で検索すると見つかります)。

```
DictGrp3() ; Google 日本
{
    WebOpen(1, "https://www.google.co.jp/search?q=%s&hl=ja&gl=jp")
}

DictGrp6() ; Google 米国
{
    WebOpen(1, "https://www.google.co.jp/search?hl=en&gl=us&q=%s")
}

(中略)

DictGrp60() ; M-W
{
    WebOpen(1, "https://www.merriam-webster.com/dictionary/%s")
}

DictGrp63() ; AHD
{
    WebOpen(1, "https://www.ahdictionary.com/word/search.html?q=%s")
}

DictGrp69() ; ロングマン英和
```

```
{
  WebOpen (1, "https://www.ldoceonline.com/jp/dictionary/english-japanese/%s")
}
(以下略)
```

この部分で、それぞれの検索先に対する検索処理を記述しています。

個々の検索先に対する処理記述の枠組みは

```
DictGrpXX ()
{
  ; 何らかの処理
}
```

という形式です。

新しい検索先を追加するときも、まずはこの枠組みを作成します。このとき、「DictGrpXX」の「XX」の部分には、先ほど指定した検索先 ID 番号を記述してください。先ほどの例では「64」と指定しましたので、「DictGrp64」という名前の枠組みを、次のように作成します。

```
(前略)

DictGrp60() ; M-W
{
  WebOpen (1, "https://www.merriam-webster.com/dictionary/%s")
}

DictGrp63() ; AHD
{
  WebOpen (1, "https://www.ahdictionary.com/word/search.html?q=%s")
}

DictGrp64()
{
}

DictGrp69() ; ロングマン英和
{
  WebOpen (1, "https://www.ldoceonline.com/jp/dictionary/english-japanese/%s")
}
(以下略)
```

3. Web サイトに対する検索処理を記述する

上で作成した「DictGrpXX() { ... }」の枠組みの中に、Web サイトの検索処理本体を記述していきます。

Web サイトの検索処理は、上の例にある他サイトと同様に、

```
DictGrp64 ()
{
    WebOpen (1, "https://~")
}
```

という形式で、URL を埋め込む形で記述します。

URL の部分は、検索対象のサイトを Web ブラウザで開き、手動で検索して、その結果ページから取得してください。今回の例なら、目的の Onelook Dictionary Search (<https://onelook.com/>)を開いて、たとえば「example」という単語を検索してみると、検索結果ページの URL は次のようになっています。

```
https://onelook.com/?w=example&ls=a
```

この検索結果の URL をコピーして、先ほどのコードで青色になっていた "https://~" という部分に貼り付けたうえで、**検索語句の箇所**(上の例なら緑色の「example」の箇所)を「%s」に置き換えてください。

※検索結果の URL に検索語句が含まれていない場合、この形式のコードでは検索できません。対処方法は[こちら](#)をご覧ください。

つまり今回の例では、Onelook を検索する処理は、下記の青色部分のように記述することになります。

```
DictGrp64 ()
{
    WebOpen (1, "https://onelook.com/?w=%s&ls=a")
}
```

以下で検索先の登録は完了です。かんざしの入力ウィンドウやポップアップメニューを開いて、実際に検索できるかどうか確認してみてください。

新たな検索先として Windows 上の辞書ソフトを追加する

Windows 上の辞書ソフトを新たな検索先として追加する作業も、全体は大きく 3 つに分かれます。

1. 検索先の名前、ショートカットキー、ホットキーを指定する
2. 検索処理を記述するための「枠組み」を作成する
3. 検索処理を記述する

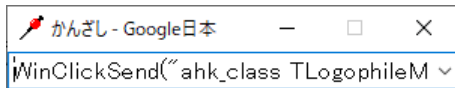
このうち 1. と 2. は、[前項](#)で説明した手順とまったく同じです。ここでは、3. の検索処理の記述方法のみ説明します。

検索処理の枠組みを次のように作ったとします。

```
DictGrp22 ()
{
}
}
```

この中に記述する、辞書ソフトの検索処理のコードは、以下の手順で取得できます。

- (1) Kanzasi.ahk をダブルクリックして、AHK 版かんざしを起動する。
- (2) 検索対象の辞書ソフトを起動する。
- (3) 辞書ソフトの語句入力エリアの上で、Ctrl キーと Alt キーを押しながらマウスの中ボタンをクリックする。(マウスのホイール部分をクリックして押し下げてください。ホイールの回転とは異なります)。すると、かんざしの入力ウィンドウがポップアップし、クリック先の辞書ソフトを検索するためのコードが次のような形で表示されます。



- (4) 表示された内容全体をクリップボードにコピーする (Ctrl+A で全選択し、Ctrl+C でコピー)。
- (5) 冒頭で示した DictGrpXX() { ... } の枠組みの中に、その内容を貼り付ける。

例: (実際のコードは検索先のソフトによって異なります)

```
DictGrp22 ()
{
    WinClickSend("ahk_class TLogophileMainForm", "TEdit2", "{Home}+{End}^v{Enter}") ; 「Oxford Learner's THESAURUS」, "x332 y69"
}
```

これで検索先の登録は完了です。実際に検索できるかどうか確認してみてください。(検索先の辞書ソフトを起動した状態でお試ください)

※この方法でうまく検索できないときには、貼り付けたコードに修正を加えることで検索できるようになる場合があります。詳しくは[こちら](#)をご覧ください。

※上記で貼り付けたコードで、カッコ内の 2 番目の項目が "xOO yOO" という形式の場合、辞書ソフトのウィンドウ内で入力エリアの位置が変わると、検索時に予期せぬ動作を起こす可能性が高くなります。ウィンドウのサイズや表示項目の配置を不用意に変更しないようご注意ください。

```
WinClickSend("ahk_class TMainForm", "x158 y624", "{Home}+{End}^v{Enter}") ; ...
```

↑ここが座標指定の場合は注意！

Web サイトや辞書ソフトを串刺し検索する

複数のソフトやサイトを 1 回の操作で串刺し検索するには、DictGrpXX() の枠組みの中に、目的のソフトやサイトに対する検索処理を並べて記述しておけば OK です。たとえば次のように記述すると、Google と Bing を 1 回の操作で串刺し検索できます。

```
DictGrp16 ()
{
    WebOpen (1, "https://www.google.co.jp/search?q=%s&hl=ja&gl=jp")
    WebOpen (2, "https://www.bing.com/search?q=%s")
}
```

※上の例で、WebOpen の直後の「1」「2」という数字は、検索結果を Web ブラウザのどのタブで開くかを表します。

上の例なら、Google の検索結果は先頭のタブで、Bing の結果は 2 番目のタブで、それぞれ開きます。

Web サイトに対する検索と、Windows 上の辞書ソフトに対する検索が混在していても問題ありません。

独自の検索処理を記述する

ここまでの例では、DictGrpXX() の枠組みの中で、WebOpen(...) や WinClickSend(...) など、かんざしが用意しているヘルパー関数を使って検索処理を記述してきました。このほか、DictGrpXX() の中では、AHK の一般的なコードと同じような形で、独自の検索処理のロジックを記述していただいてもかまいません。検索語句はクリップボードに入っていますので、Ctrl+V で検索語句の入力フィールドに貼り付けたり、Clipboard 変数を使って操作したりできます。

デフォルトのコードでは、EBwin を検索する DictGrp21() がそのような処理の例です。EBwin の exe ファイルに検索語句を起動パラメータとして渡すという方法で、検索処理を実行しています。

```
DictGrp21() ; EBwin
{
    ebwin64 := "C:\Program Files\EBWin4(x64)\EBWin4.exe"
    ebwin32 := "C:\Program Files (x86)\EBWin4\EBWin4.exe"

    ebwinexe := FileExist(ebwin64) ? ebwin64 : FileExist(ebwin32) ? ebwin32 : ""
    if (ebwinexe != "") {
        Run, % ebwinexe . " /S="" . StrRepl(Clipboard, "", "", "") . ""
    }
}
```

また、たとえば次のようなコードを記述すると、Google Chrome のウィンドウをアクティブ化して、Ctrl + F で検索ボックスを開き、ページ内検索を実行できます。

```
DictGrp100()
{
    if (WinExist("Google Chrome")) {
        WinActivate
        Sleep, 100
        Send, ^f
        Sleep, 100
        Send, ^v
    }
}
```

2 目的別 設定リファレンス

主な設定項目について、修正箇所を目的別にまとめました。

目的	設定項目
入力ウィンドウのデフォルトの位置とサイズを変更するには	ini_x, ini_y, ini_width を修正
「入力ウィンドウを常に表示」モードをデフォルトでオンにするには	ini_stayopen を修正
辞書ソフトやブラウザが未起動の場合に自動で起動してから検索させるには	ini_winexe を設定
検索先を新規作成・修正・削除するには	ini_grpid でID番号・検索先名・ショートカットキー等を定義し、 DictGrpXX() で検索処理を記述
検索先の名前を変更するには	ini_grpid でグループ名の定義を修正
入力ウィンドウ上で辞書グループを選択するためのショートカットキーを変更するには	ini_grpid で () 囲みのキー定義を修正
ホットキー一発検索のキーを変更するには	ini_grpid で (()) 囲みのキー定義を修正
Windows 上の辞書ソフトを検索先として追加するには	DictGrpXX() に追加
Windows 上の辞書ソフトが既製のコードでうまく検索できない場合は	DictGrpXX() 内の WinClickSend() の記述を修正
URL に検索語句が含まれるタイプの Web サイトを検索先として追加するには	DictGrpXX() に追加
URL に検索語句が含まれないタイプの Web サイトを検索先として追加するには	DictGrpXX() に追加
検索結果を Web ブラウザで開くとき、どのタブで開くかを指定するには	ini_browserwin でブラウザを指定したうえで、 DictGrpXX() 内の WebOpen() または WebOpenSend() に タブ番号を指定
入力ウィンドウやポップアップメニューを表示するホットキーを変更するには	ホットキー部のキー定義 を変更
入力ウィンドウで使用する短縮入力を指定するには	Kanzasi_hotstring.txt を開いて編集するか、 ショートカット部 または ホットキー部 で定義

次のページからは、スクリプト内の個別の構成要素の方法について説明していきます。

細かい話が延々と続きますので、上の表を参照のうえ、必要になったときに必要な箇所のみお読みいただくことをお勧めします。

GUI の設定

入力ウィンドウの表示などに関する設定を行うセクションです。

デフォルト設定

```
ini_x := -100, ini_y := 0, ini_width := 300

ini_stayopen      := false
ini_stayontop    := true
ini_fontname     := "MS Pゴシック"
ini_fontsize     := 13

ini_activate1stwin := true
```

設定項目

入力ウィンドウのデフォルトの位置とサイズ

項目	ini_x, ini_y, ini_width
指定値	入力ウィンドウの上で「 Ctrl+Alt+中ボタン 」し、表示されたコードをそのまま上書きで貼り付け
説明	<p>実行中に入力ウィンドウの位置とサイズを変更しても、自動では記憶されず、再起動するとデフォルトの状態に戻ります。その「デフォルトの状態」を指定するための場所がここです。</p> <p>入力ウィンドウを目的の位置とサイズに調整したうえで「Ctrl+Alt+中ボタン」し、表示された内容を貼り付けてください。</p> <p>なお、位置に関しては、特別な設定値として次の 2 つがあります。</p> <ul style="list-style-type: none">• ini_x := -100 …入力ウィンドウを画面中央に表示• ini_x := -200 …入力をアクティブウィンドウの中央に表示 <p>(これらの値を指定するときは、ini_y の値は無視されます)。</p>

「入力ウィンドウを常に表示」をデフォルトでオンにするかどうか

項目	ini_stayopen
指定値	true / false
説明	実行中にタスクトレイで設定を変更できる項目ですが、それを起動時点でオンにするかどうかをここで指定します。

入力ウィンドウを常に前面表示するかどうか

項目	ini_stayontop
指定値	true / false

入カウインドウのフォントの種類

項目	ini_fontname
指定値	フォント名を " " で囲んで指定

入カウインドウのフォントのサイズ

項目	ini_fontsize
指定値	フォントサイズを表す数値

送信完了後に先頭の辞書をアクティブ化するかどうか

項目	ini_activate1stwin
指定値	true / false
説明	false の場合は、送信前にアクティブだったウィンドウに戻ります。 (ただし作者の環境では、入カウインドウが閉じた直後になぜかフォーカスが別のウィンドウに移ってしまうことがあり、false の場合には予期したウィンドウとは異なるウィンドウに戻ってしまうことがあります)。

Web ブラウザの設定・自動起動するソフトの設定

Web 検索に使うブラウザと、未起動時に自動起動する辞書ソフト類を指定するセクションです。

設定例

```
ini_browserwin := "ahk_class MozillaWindowClass"

ini_logfilename := ""

ini_winexe := { "" : ""
, "ahk_class TJammingMainForm" : "C:¥Program Files (x86)¥Jamming¥Jamming.exe,1000" }
```

設定項目

Web 検索で使用するブラウザ

項目	ini_browserwin
指定値	Google Chrome の場合は"chrome"、Mozilla Firefox の場合は"firefox"、Microsoft Edge の場合は"edge"。それ以外のブラウザの場合は、 <u>ブラウザのウィンドウ上で「Ctrl+Alt+中ボタ</u> <u>ン</u> 」し、ドロップダウンの「〈ウィンドウクラス〉」の内容を " " で囲んで貼り付け
説明	デフォルトでは、Web 検索の結果は、Windows 上で「既定のブラウザ」として設定しているブラウザの新しいタブで表示されますが、ここでブラウザを明示的に指定しておくことをお勧めします。(別の場所の設定と併用することにより、検索結果をどのタブで開くかを指定できるようになるからです)。

ログファイル名

項目	ini_logfilename
指定値	ログファイル名を表す文字列
説明	ここにファイル名を指定すると、検索を実行するたびにそのファイルに履歴が記録されるようになります。ファイルが存在しない場合は自動で作成されます。 ※古い履歴を自動で削除する機能はありません。長い間使用していると、ログファイルがどんどん大きくなっていきます。お手数ですが、必要に応じてご自身で整理してください。(ログファイルをテキストエディタで開いて、不要な行を手動で削除すれば大丈夫です)。

自動起動するソフトの指定

項目	ini_winexe
指定値	対象の辞書ソフトやブラウザのウィンドウ上で「 Ctrl+Alt+中ボタン 」し、ドロップダウンの「〈EXE 自動起動用コード〉」の内容を貼り付けて、行末に「}」を追加（この後の説明を参照）
説明	語句を送信しようとした時点で辞書ソフトやブラウザが未起動だった場合に、自動で起動させたい場合は、そのソフトをここで指定しておく必要があります。

ini_winexe は「[Ctrl+Alt+中ボタン](#)」からのコピーで指定できます。「〈EXE 自動起動用コード〉」の内容を、下記の青色の部分に貼り付けてください。

```
ini_winexe := { "" : "" ; ウィンドウクラス(and/or タイトル)と EXE ファイルの対応付け(先頭行はダミー)  
, "この行にコードを貼り付け" : "" }
```

自動起動させたいソフトが複数ある場合は、それぞれに対する「Ctrl+Alt+中ボタン」の「〈EXE 自動起動用コード〉」を、1 行に 1 つずつ、並べて貼り付けてください。

そのうえで、最後の行の末尾にのみ「}」を加えて、次のような形としてください。

```
ini_winexelist := { "" : "" ; ウィンドウクラス(and/or タイトル)と EXE ファイルの対応付け(先頭行はダミー)  
, "ahk_class TJammingMainForm" : "C:\Program Files (x86)\Jamming\Jamming.exe,1000"  
, "ahk_class Chrome_WidgetWin_1" : "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe,1000" }
```

各行の先頭はカンマ

最終行のみ、末尾に中カッコ閉じを付ける

貼り付けたコードの exe のフルパスが間違っていないかどうか、念のためご確認ください。

また、exe のパスの末尾に「,1000」とあるのは、その exe を起動した後で何ミリ秒待ってから検索語句を送るかを表しています。起動の後の初期処理に時間がかかってすぐには検索できないソフトの場合は、この数字を大きくしてください。

検索先の ID 番号・名前・ショートカットキー・ホットキーの設定

検索先の ID 番号・名前・ショートカットキー・ホットキーを設定するセクションです。

デフォルト設定

```
ini_grpid := { 3: "Google 日本"           = (^g), ((^+g))"
              , 6: "Google 米国"         = (^u), ((^+u))"
              , 9: "Google 英国"         = (^k)"
              , 12: "Google 米画像"      = (^i)"
              , 15: "Bing"                = (^b)"
              , 18: "-----"
              , 21: "EBWin4"              = (^q), ((^+q))"
              , 24: "-----"
              , ... (以下略)
```

設定項目

1 行が 1 項目で、「検索先 ID 番号: "検索先名 = (ショートカットキー), ((ホットキー))"」という形式です。

- ◆ 既存の検索先の名前やキーを修正する場合は、該当する箇所を直接書き換えてください。
- ◆ 新しい検索先を作成する場合は、デフォルトの記述をコピーする形で行を追加し、各項目を書き換えてください。検索先はいくつでも作成できます。
- ◆ 検索先を削除する場合は、行ごと削除してください。

それぞれの項目の内容と指定値は次のとおりです。

検索先 ID 番号

内容	スクリプト内で検索先を区別するために付ける番号
指定値	正の整数値
説明	他と重複しない値を自由に付けてかまいません。ポップアップメニューの検索先名はこの番号順に表示されます。 ここで指定した番号は、実際の検索処理を記述する箇所でも使います。

検索先名

内容	入力ウィンドウやポップアップメニューで表示される検索先名
指定値	任意の文字列

ショートカットキー

内容	入力ウィンドウ上で検索先を選択するためのショートカットキー
指定値	キーを表す文字列を () [半角カッコ 1 つの組] で囲んで指定。

	カンマ区切りでいくつでも指定可能
説明	キーの文字列は AHK の書式どおりです。「ホットキー設定ヘルパー」を使うと簡単です。 ショートカットキーが不要な検索先は、指定を省略してかまいません。

ホットキー

内容	検索先を一発検索するためのホットキー
指定値	キーを表す文字列を (()) [半角カッコ 2 つの組] で囲んで指定。 カンマ区切りでいくつでも指定可能
説明	キーの文字列は AHK の書式どおりです。「ホットキー設定ヘルパー」を使うと簡単です。 ホットキー一発検索が不要な検索先は、指定を省略してかまいません。

※ショートカットキーは () で囲み、ホットキーは (()) で囲みます。ホットキーは幅広いウィンドウで有効なキーだからカッコ (()) も幅広い、と覚えるとよいかもかもしれません。

なお、このカッコ囲みの表記法は、かんざしで独自に定めたものであり、この部分でのみ有効です。スクリプト内の他の部分や、一般的な AHK スクリプトでは使えませんのでご注意ください。

記述方法に関する注意・補足

- "検索先名 = (ショートカットキー), ((ホットキー))" という部分は、この部分の文字列全体を " " で囲んでください。(左辺と右辺の各項目を " " で囲む形ではありません)。
- ショートカットキーとホットキーの組み合わせは、すべての検索先を通して一意でなくてはなりません。重複がないように指定してください。
- ショートカットキーもホットキーも不要な検索先は、たとえば「15: "Bing"」のように、検索先名のみを " " で囲んで指定する形がかまいません。
- ポップアップメニューで区切り線を入れたい箇所には、検索先名に "-----" (半角のマイナス 8 個) を指定してください。その項目にも ID 番号は必要ですが、DictGrpXX は不要です。
- 定義の修正が終わったら、次のような形式になっているかどうかご確認ください。

```
ini_grpid := {
10: "ローカル辞書 = (^d), (F1), ((^+d))"
, 20: "Google 日本 = (^g), (F2), ((^+g))"
, 30: "Google US = (^u), (F3)"
, 40: "オンライン英英 = (^e), (F4)"
, 50: "オンライン類語 = (^r), (F5)" }
```

先頭に中カッコ開きがある

最終行のみ、末尾に中カッコ閉じがある

2 行目以降の各行の先頭にカンマがある

なお、この ini_grpid で検索先を追加した場合は、この後の説明に従って、DictGrpXX の検索先も指定してください。(区切り線の "-----" の項目には DictGrpXX は必要ありません)。

各検索先に対する検索処理の設定

それぞれの検索先について、辞書ソフトや Web サイトに対する検索処理を記述するセクションです。

```
DictGrp3() ; Google 日本
{
    WebOpen(1, "https://www.google.co.jp/search?q=%s&hl=ja&gl=jp")
}

DictGrp6() ; Google 米国
{
    WebOpen(1, "https://www.google.co.jp/search?hl=en&gl=us&q=%s")
}

DictGrp9() ; Google 英国
{
    WebOpen(1, "https://www.google.co.jp/search?hl=en&gl=uk&q=%s")
}
(以下略)
```

前述の ini_grpid で指定した検索先 ID 番号ごとに、DictGrpXX() { ... } という枠組みを用意し、それぞれの { } 囲みの中に検索先を記述していきます。複数の検索先を 1 回の操作で検索させたい場合は、{ } 囲みの中に複数の検索処理を、たとえば下記のように記述していきます (Windows 上の辞書ソフトに対する検索と Web 検索が混在していても問題ありません)。

```
DictGrp60()
{
    WinClickSend(....) ; ローカル検索
    WinClickSend(....) ; ローカル検索
    WebOpen(0, "http://www.google.com/search?hl=en&q=%s") ; Web 検索
    WebOpen(0, "http://www.onelook.com/?w=%s&ls=a") ; Web 検索
}
```

検索先の種類

検索先は、主に次の 3 種類に分かれます。

- ◆ **ローカル辞書ソフト**: Windows 上の辞書ソフトやアプリケーションの入力エリアに語句を送信する検索です。(設定方法は[こちら](#))
- ◆ **URL 型検索サイト**: Web 上の検索サイトや辞書サイトのうち、検索結果の URL の中に検索語句が含まれているタイプのサイトに対する検索です。(設定方法は[こちら](#))
- ◆ **非 URL 型検索サイト**: Web 上の検索サイトや辞書サイトのうち、検索結果の URL の中に検索語句が含まれていないタイプのサイトに対する検索です。できれば、該当の検索ページを開いた時点で、検索語句の入力フィールドにカーソルが置かれるページのほうが好ましいです。(設定方法は[こちら](#))

※スクリプトにデフォルトで登録してある Web 検索先は、いずれも URL 型の検索サイトです。

※非 URL 型サイトへの検索処理は、多少ぎくしゃくした動きとなり、少し時間がかかります。URL 型と非 URL 型のどちらの方法でも検索できるサイトは、URL 型として検索することをお勧めします。

以下、それぞれの設定方法を説明します。

ローカル辞書ソフトを検索先として指定する方法

ローカル辞書ソフトを検索先に追加する場合は、検索対象のソフトの入力エリア上で「Ctrl+Alt+中ボタン」し、表示されたコードを、DictGrpXX() のブロックの中に貼り付けてください。

貼り付けたコードで正しく検索できない場合の対処法

辞書ソフトによっては、「Ctrl+Alt+中ボタン」から貼り付けたコードで正しく検索できない場合がありますが、次に示す修正方法により検索できる可能性があります。

○ インクリメンタルサーチの辞書ソフトの場合

キー入力と並行して検索が実行される「インクリメンタルサーチ」型の辞書ソフトの場合は、最後に送信する Enter キーが不要となります。貼り付けたコードで、カッコ内の末尾にある項目から{Enter}を削除してください。

```
WinClickSend(..., "{Home}+{End}^v{Enter}")  
↑この部分を削除
```

○ 前の検索語句が残ってしまう場合

前の検索語句が消えずに次の語句が送られている場合は、貼り付けたコードでカッコ内の末尾にある {Home}+{End}^v{Enter} という部分を修正すると直る可能性があります。

```
WinClickSend(..., "{Home}+{End}^v{Enter}")  
↑この部分を修正
```

- 入力エリアを Ctrl+A で全選択できる辞書ソフトの場合は、"^a^v{Enter}" に変更してみてください。
- 入力エリアの右クリックメニューに「すべて選択(A)」という項目がある辞書ソフトの場合は、"+{F10}^v{Enter}" に変更してみてください。
- ウィンドウがアクティブ化した時点で入力エリアが全選択状態になる辞書ソフトの場合は、"^v{Enter}" に変更してみてください。

○ ウィンドウ自体がアクティブにならない場合

ウィンドウ自体がアクティブにならない場合は、貼り付けたコードのカッコ内の先頭項目をウィンドウのタイトルに変更してみてください。(タイトルの一部のみでもかまいません)。

タイトルの内容は、貼り付けたコードの末尾のほうに「」囲みで記載してあります。

```
WinClickSend("ahk_class olt1UIWindowClass", ... ) ; 「〜」  
↑この部分をウィンドウのタイトルに変更 ↑タイトルはここにあり  
(実際の内容はプログラムにより異なります)
```

○ ウィンドウはアクティブになるがキーが送られない場合

貼り付けたコードでカッコ内の 2 番目にある項目を、入力エリアの座標による指定に変更すると、検索できる可能性があります。

座標は、貼り付けたコードの末尾に "x000 y000" のような形式で記述してあります。その形式のままコピーしてください。

```
WinClickSend("ahk_class ~", "olt1WindowClass13" ... ) ; 「～」, "x130 y112"  
↑この 2 番目の項目を座標に変更      ↑座標はここにあり  
(実際の内容はプログラムにより異なります)
```

※このように座標を直接指定する形のコードにした場合、辞書ソフトのウィンドウ内で入力エリアの位置が変わったときに予期せぬ動作となる可能性がありますのでご注意ください。

○ 登録した直後は検索できたが、辞書ソフトを再起動したら検索できなくなった場合

辞書ソフトによっては、送信先のウィンドウと入力エリアを特定するための情報が、起動のたびに変わってしまうものがあります。その場合にこの現象が発生します。

上で説明した「ウィンドウ自体がアクティブにならない場合」「ウィンドウはアクティブになるがキーが送られない場合」の内容にならって、貼り付けたコードのカッコ内の 1 番目の項目をウィンドウのタイトルに、2 番目を入力エリアの座標に、それぞれ変更してみてください。(どちらか一方の修正のみで済む場合もあります)。

URL 型検索サイトを検索先として指定する方法

検索結果の URL に検索語句が含まれている「URL 型」の検索先を追加するときには、DictGrpXX() のブロックの中に、「WebOpen(0, "http～")」という形式でコードを記述します。

URL の部分は、検索語が入る位置を「%s」で置き換えた形で指定します。目的の Web サイトで、たとえば「sample」といった適当な単語を検索し、検索結果ページの URL をコピーしてから、「sample」とある部分を「%s」に置き換えれば、たいていは動作すると思います。

%s の部分には、文字コード「UTF-8」で検索語句が挿入されます。もし文字化けが生じる場合、URL に文字コードの設定オプションがあれば、UTF-8 となるよう設定してみてください。

非 URL 型検索サイトを検索先として指定する方法

検索結果の URL に検索語句が含まれていない「非 URL 型」の検索サイトも、かんざしから検索可能です。できれば、該当の検索ページを開いた時点で、検索語句の入力フィールドにカーソルが置かれるタイプのページのほうが好ましいです。

○ 非 URL 型で、検索ページを開いたときに入力フィールドにカーソルが置かれるサイトの場合

Web ページ内の検索語句入力フィールドの上で「Ctrl+Alt+中ボタン」し、ドロップダウンの「〈非 URL 型 Web 検索用コード〉」を DictGrpXX() の中に貼り付けたうえで、URL を指定してください。URL は「%s」のないものをそのまま記述してください。

なお、非 URL 型検索サイトの検索は、処理の都合上、多少ぎくしゃくして時間がかかる形の実装となっております。ご了承ください。

○ 非 URL 型で、検索ページを開いたときに入力フィールドにカーソルが置かれられないサイトの場合

こうした形式のサイトも一応は検索可能です。「[Ctrl+Alt+中ボタン](#)」の「〈非 URL 型 Web 検索用コード〉」からコードを貼り付けたうえで、カッコ内の 5 番目の項目を、末尾にある "xOO yOO" という座標に変えてください。

例:

```
WebOpenSend(0, "http://kod.kenkyusha.co.jp/demo/form.jsp", 10, "研究社辞書検索", "", "^a^v{Enter}"); "x796 y259"
```

↑この青色の部分を、
行末にある緑色の内容に変更

※ただし、このように座標を直接指定する形のコードにした場合、ページのサイズや配置が変わったときに予期せぬ動作となる可能性がありますのでご注意ください。

Web ページを開くタブ番号の指定

デフォルトでは、Web 検索の結果は、Windows 上で「既定のブラウザ」として設定しているブラウザの新しいタブで表示されますが、Google Chrome、Mozilla Firefox、Microsoft Edge の場合は、検索結果を開くタブを指定できます。たとえば、Google の検索結果は常にブラウザの 1 番左のタブで開く、といったように、何をどのタブを開くか固定しておくくと便利です。

この機能を使うためには、次の 2 つの両方を行う必要があります。

- ◆ 検索に使うブラウザの設定。
「[ブラウザの設定](#)」セクションで説明した「ini_browserwin」という項目で設定してください。
- ◆ 検索を実行するコードでのタブ番号の指定。
検索用コードのカッコ内の先頭項目(下記の青色の部分)に、目的のタブ番号を表す数字を指定してください。
 - ・ URL 型検索サイトの場合: `WebOpen(0, "http://~")`
 - ・ 非 URL 型検索サイトの場合: `WebOpenSend(0, "http://~", ...)`

タブ番号として指定できる数字と動作は次のとおりです。

タブ番号の指定	動作
0	新しいタブで開く
1~8	左から n 番目のタブで開く(1=一番左のタブ、2=左から 2 番目のタブ、…)
9	一番右のタブで開く
その他	現在のタブで開く

※1~9を指定した場合、該当するタブがないときにどうなるかはブラウザ次第です。なるべく、常に存在するタブの番号を指定してください。

入力ウィンドウやポップアップメニューを表示するホットキーの設定

入力ウィンドウやポップアップメニューを表示するホットキーを設定するセクションです。

※一発検索のホットキーは、「[検索先の ID 番号・名前・ショートカットキー・ホットキーの設定](#)」セクションで定義します。

```
^+k::InputBoxActivate() ; 入力ウィンドウを表示
^+d::InputBoxActivate(SelGet(false)) ; 選択語句を取り込んで入力ウィンドウを表示
^+i::PopupMenuActivate(1, " ") ; 前回と同じ語句を検索するポップアップメニューをカーソル位置に表示
^+e::PopupMenuActivate(1) ; 選択語句を検索するポップアップメニューをカーソル位置に表示
MButton::PopupMenuActivate(0, , "{MButton}") ; 選択語句を検索するポップアップメニューをマウス位置に表示
!^MButton::KanzasiInfoGet() ; スクリプト修正用の情報を取得して表示
^+n::NextDictActivate(1) ; 次の辞書をアクティブ化
^+p::NextDictActivate(-1) ; 前の辞書をアクティブ化
```

どのホットキーに対して何をするか、それぞれ 1 行ずつで記述した形となっています。

キーを変更するには、上記の青色で示した部分を、目的のホットキーを表す文字列に変更してください。キーの指定方法は AHK の書式どおりで、「[ホットキー設定ヘルパー](#)」を使うと簡単です。

※上の設定例で、3 行目の「MButton」のみ、行頭と行中の 2 カ所に出てきます。この機能を別のマウスボタンに変える場合は、同じように、行頭と行中の 2 カ所に記述してください。

また、短縮入力のホットストリングをここで定義してもかまいません(その場合、かんざしの入力ウィンドウ以外も含め、デスクトップ全体で有効な短縮入力となります)。指定方法の詳細については、[AHK のヘルプ](#)などを参照してください。

入カウィンドウ上のショートカットキーの設定

入カウィンドウ上でのみ有効なショートカットキーを設定するセクションです。

```
#IfWinActive かんざし - ahk_class AutoHotkeyGUI

^2::                                ; 全体の " " 囲みのオン/オフの切り替え
    hk_text := InputBoxGetText()
    hk_text := InStr(hk_text, "\"") ? StrRepl(hk_text, "\"") : "\"" . hk_text . "\""
    InputBoxSetText(hk_text, 2)
    return

^3::                                ; 各単語の " " 囲みのオン/オフの切り替え
    hk_text := InputBoxGetText()
    hk_text := InStr(hk_text, "\"") ? StrRepl(hk_text, "\"") : RegExReplace(hk_text, "(*UCP)(¥S+)", "\"$1\"")
    InputBoxSetText(hk_text, 2)
    return

#IfWinActive
```

デフォルトで設定されている「Ctrl+2」「Ctrl+3」というキーを変更したい場合は、上記の青色の部分を変更してください。

また、短縮入力のホットストリングをここで定義してもかまいません。指定方法の詳細については、[AHK のヘルプ](#)などを参照してください。たとえば、「,a」「,m」「,g」という 3 種類の短縮入力を指定する場合、次のような 3 行を記述します。

```
:*,,a::site:apple.com
:*,,m::site:microsoft.com
:*,,g::site:google.com
```

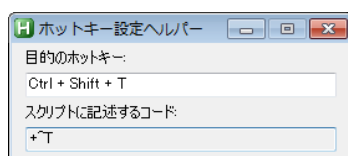
4 スクリプト修正のヘルパー機能

スクリプトの修正作業全般で利用する2つのヘルパー機能について説明します。

ホットキー設定ヘルパー

スクリプトの中には、ホットキーやショートカットキーを指定する箇所がいくつかあります。こうした所では、たとえば Ctrl+V は「^v」、Alt+F1 は「!F1」というように、AHK で定められた書式に従って、キーの組み合わせを表す文字列を記述する必要があります。

AHK 版かんざしに同梱の「ホットキー設定ヘルパー」を使うと、キーの記述方法を簡単に確認できます。



使い方は簡単です。

- (1) エクスプローラーで、Kanzasi.ahk と同じフォルダにある「ホットキー設定ヘルパー.ahk」をダブルクリックしてください。上のような画面が表示されます。
- (2) 「目的のホットキー」の欄で、指定したいホットキーを押してください。下の欄に、そのキーを表すコードが表示されます。

スクリプトの中でキーを指定する部分には、このコードを貼り付けてください。

ただしこのツールでは、一部の特殊なキーや、マウスのボタンを使うホットキーのコードは確認できません。AHK のヘルプでご確認ください。

※このツールで表示されるコードでは、a~z の大文字・小文字が、Kanzasi.ahk のデフォルトの記述と異なる場合がありますが、通常はどちらでも問題ありません。

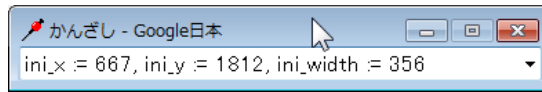
「Ctrl+Alt+中ボタン」

スクリプトを修正するときには、「Ctrl+Alt+中ボタン」で表示されたコードを貼り付けるという形で対応する部分があります。

この「Ctrl+Alt+中ボタン」には、次のような機能があります。

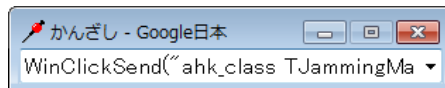
入力ウィンドウ上での「Ctrl+Alt+中ボタン」

入力ウィンドウ上で「Ctrl+Alt+中ボタン」をクリックすると、ウィンドウの x / y 座標と幅の情報が表示されます。

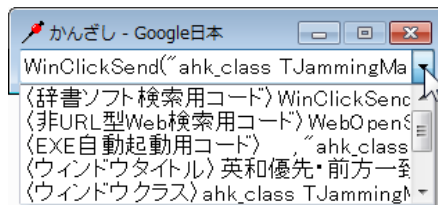


「Ctrl+Alt+中ボタン」の後のドロップダウンリスト

アプリケーションのウィンドウや入力エリアの上で「Ctrl+Alt+中ボタン」すると、ローカル辞書ソフトとして検索するためのコードが表示されます。これは冒頭のローカル辞書登録で行ったとおりです。



ここで、入力ウィンドウの右端の[▼]ボタンをクリックすると、ドロップダウン部分に、別のコードやデータがいくつか入っています。スクリプトの中では、こうしたコードやデータを貼り付けて使う箇所もあります。



リスト内の項目をクリックすると入力エリアに表示されますので、それをコピーして貼り付けてください。

ただし、行頭にある `< >` 囲みの項目名はスクリプトでは不要ですので、その部分を取り除いた形で貼り付けてください。

ドロップダウンリストに入っている項目は次のとおりです。

項目	意味
辞書ソフト検索用コード	ローカル辞書ソフトに検索語句を送信するコード (入力エリアに表示されるコードと同一)
非 URL 型 Web 検索用コード	URL に検索語句が含まれないタイプのサイトを検索するためのコード
EXE 自動起動用コード	自動起動させる辞書ソフトやブラウザを設定するためのコード
ウィンドウタイトル	マウス位置のウィンドウのタイトル部の内容
ウィンドウクラス	マウス位置のウィンドウのクラス*
マウス位置相対座標	ウィンドウの左上を基点とした、マウスの位置
マウス位置クラス	マウス位置のコントロールのクラス*
EXE フルパス	マウス位置のウィンドウの EXE ファイルのフルパス

* 「クラス」とは、プログラムの中でウィンドウやコントロールを指定するための識別子の一種です。