# USAGE TIPS FOR WAYTEXT TOOL | This Article Updated 5-3-2024

## Introduction

The WayText application evolved from a need to enter boilerplate first-person narratives into webforms and emails. Working in the world of Special Education, I often need to write to one person (a parent), about another (their child), hence the need for first-person pronouns. As an itinerant staff person, I send documents from different offices, hence the desire to customize that information.



## Types of encoding/flags used by WayText

### INI File structure

Aka "Initialization file." aka "Config" or "Configuration file." Users of the WayText application need a basic understanding of ini 'Sections,' 'Keys,' and comments. See more about ini files below. See also: https://en.wikipedia.org/wiki/INI_file

### "MiniForm" key names

These are ini keys that have the word "MiniForm" in their key name. There should be a list of items under the key name. Each item should be on its own line. At runtime a form will pop up and offer those items as options to choose. "MiniForm" is not CaseSensitive.

### {Title} in INI key name

Allows users to specify a Title/Label at the top of the form, for example (Fruits} in the example on the left.



### Checked/Uncheck by default defined with 1/0 in key value.

The list of items in the MiniForm key may be "pre-checked" if the line is preceded with "1 " (one and space). An item will be unchecked if "0 " is used or if this level of encoding is skipped. Example:
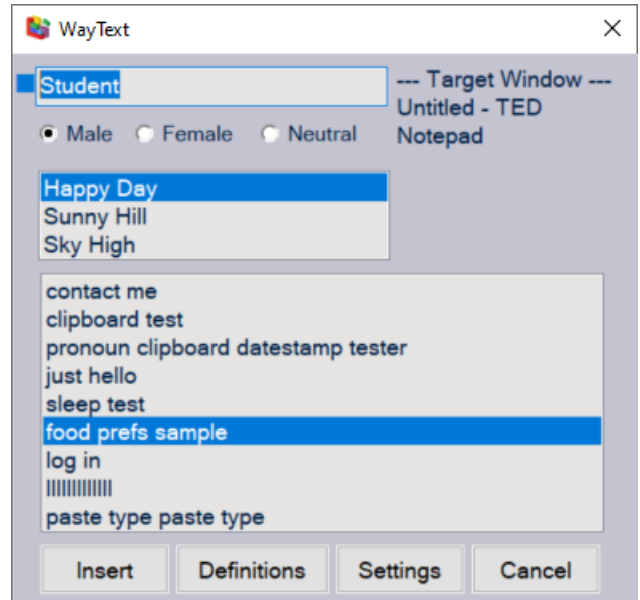
> *Sample MiniForm Key=*
> *0 item is unchecked by default*
> *1 item is checked by default*
> *this item is also unchecked*

### "Punct" in MiniForm key name

Tells the tool to use Oxford comma with "and" in the list of items chosen from the list of items in the MiniForm key. Can use "Punctuate" or "Punctuation." Really, it's checking for the string "punct". With punctuation flag, the information obtained from the Fruits example would be, "*apples, bananas, and oranges.*" Without "Punctuation" in the key name, the resulting text would be, "*apples bananas oranges*." Is not CaseSensitive. If the word is meant to be at the end of a sentence (such as the Fruit example; see TextDefinitions.ini item), you can put a period at the end of each MiniForm selection (see also MiniForm screenshot). WayText will attempt to detect the periods, and strip all but the last one, so that the punctuation will be correct. For this reason, be careful when using MiniForms for web URLs, email addresses, or other things with.embedded.dots.

### "Radio" in MiniForm key name

Tells the tool that only one item from this list will be chosen. Therefore, use round radio buttons rather than the normal square checkboxes. Since there can only be one choice, the Punctuate flag becomes moot. Again, the string "Radio" is not CaseSensitive.

## "Paste" or "Type" in INI key name

If a key name in TextDefinitions.ini contains the word Paste, the value of that key will get pasted. This mode of pasting, stays until there is a key with "Type" in it. If neither of these is present, then Type mode is the default.

Advantage of Paste mode:
   -It's way faster.

Disadvantage of Paste mode:
   -Keyboard commands like {Tab} or {Enter} don't work.
   -Several pastes in a row will confuse Windows. WayText tries to accommodate this by combining back-to-back paste keys, into one key, prior to pasting. Also, a "post-paste delay" is used.

## "{Sleep, N}" ini INI key name.

If an ini key name has {Sleep, N} then a sleep (a pause) of N milliseconds will happen after that key's value is typed or pasted. The coma and the space are optional, so {SleepN} also works. "Sleep" is not case sensitive.

## Additional {Curly items} not in the AHK docs

If {Clipboard} is in the boilerplate text, WayText will replace it with the contents of the clipboard. You can embed a date-stamp by including {now, M-d-yyyy}. It must start with exactly "now, " but replace "M-d-yyyy" with any common format. The clipboard replacement and the time-stamp both can be used regardless of the "paste vs. type" key setting.

## {Tab} and other {Curly Keyboard items}

Key names in curly braces simulate the given key. See also:
https://www.autohotkey.com/docs/commands/Send.htm#keynames


**Note:** *{Curly brace} keyboard key simulation in the INI key values must be typed to work. They cannot be pasted. The {Clipboard} insertion, and the {now …} timestamps can be pasted though.*


## [c]ontact information

[c] gets replaced with the chosen section from OfficeList.ini. Similarly, [p] gets replaced with just the phone number of the location.

## Other [bracket] replacement codes

[n] gets replaced with student name, [e][m][s][r] are changed to pronouns, as assigned in the gui form.

## Table of Gender Pronoun Replacements

See also: https://en.wikipedia.org/wiki/Gender_in_English

| [e] is next. | *He is next.* | *She is next.* | *They are next.* <br> *Note that WayText will automatically change "They is" to "They are."* |
|---|---|---|---|
| I will tell [m]. | *I will tell him.* | *I will tell her.* | *I will tell them.* |
| It is [s] book. | *It is his book.* | *It is her book.* | *It is their book.* |
| The book is [r]. | *The book is his.* | *The book is hers.* | *The book is theirs.* |

**Tip:** When adding boilerplate text to TextDefinitions.ini, say the text in your head as though you are writing about a *male*. This is recommended because the names of the letters in the first three pronoun codes [e][m][s] sound similar to their corresponding male replacements (e-he, m-him, s-his). This doesn't work for [r], but that code is rarely used anyway ([r] = his/hers/theirs).

**Summary of bracket item replacements:**
[n] = Name.
[e] [m] [s] [r] = gender pronouns.
For selected Office: [c] = complete contact info.  [p] phone only.

More Information about .ini file format.

An ini file has an extension of .ini.  Ini files don't support bold, tables, pictures, etc., only raw text.  They are similar to text files (.txt) in that regard.

Within ini files, lines that start with a semicolon
; like this
are 'comments' and are ignored by the computer.  You can use these to make personal (human-friendly) notes embedded amongst the technical parts.

The main structure of an ini file is this:

```
[Section1]
key1=Value1
key2=Value2
[Section2]
key1=Value1
key2=Value2
etc
; comment
```

The sections are like headers and the keys are like subcomponents.

The WayText uses three different ini files.  They must be kept in the *wtFiles* folder, and they must be named:

```
TextDefinitions.ini
Settings.ini
OfficeList.ini
```

The TextDefinitions.ini sections and keys can be changed/customized/rearranged.  Each time the WayText is launched/activated, the TextDefinitions file is read, and the list is updated.  Double clicking an entry from the list, tells the WayText to go read the keys from that section, then type the contents of the section from the ini file.   Here is an example:

```
[quick sample]
key=Hello [n].
```

When the "quick sample" item is double-clicked, the tool sees the [n] and replaces that with whatever is typed into the name box of the WayText main form, then it types out "Hello Student" or "Hello Franky" or whatever into the text field was active before the tool was launched.

With the other two ini files, the Key names must be specific, but for TextDefinitions, they get removed at runtime anyway, and can be anything.  For a simple text entry that has no popup mini forms the key names can be completely left out.  If the key name is left out, however, there would be no way to tell WayText to paste the content rather than typing it.

**Disambiguation**:  ini file [Section Names] also get brackets (as discussed above).  To avoid confusion, <span style="color:red">don't start a boilerplate text definition on its own line if you also start the sentence with [n] [e] [m] [s] [r] [c] or [p].</span>
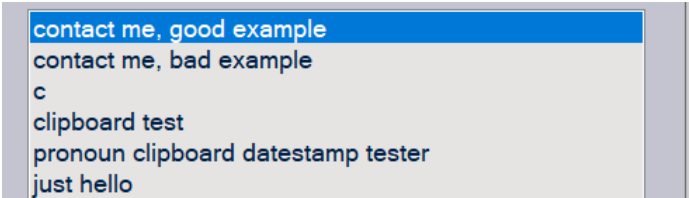
> **[contact me, good example]**
> paste key=Please contact me at your earliest convenience.  [c].
> ;=====================
> **[contact me, bad example]**
> [c] is my contact information.

With the above bad example, there is an (erroneous) attempt to insert my office contact information at the beginning of the line.  Windows (and AutoHotkey) interprets this as an ini section.  As such, (and as seen in the image on the right), WayText includes this in the list of TextDefinitions.  Only the part between the braces is displayed though (I.e. "c").  The rest is ignored.

contact me, good example
contact me, bad example
c
clipboard test
pronoun clipboard datestamp tester
just hello

## Additional information about each ini file.

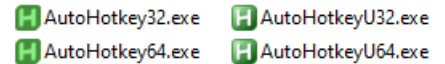| File | Comments |
|------|----------|
| *TextDefinitions.ini* | The largest one.  The sections show up as list items in the bottom part of the main tool.  The key *names* get deleted and can be can mostly be anything.  Key *values* are what get inserted (typed or pasted) by the tool.  Keys with the word "MiniForm" flag the tool that there is a list of optional popup mini form items.  The word 'Paste' means WayText will paste the content of that key.  Other replacement codes to use are [c], [p] (see OfficesList.ini below).  [n] = name.  [e], [m], [s], [r] = pronouns.  Keyboard items such as {Tab} can be used, but only if the key containing them is typed (not pasted).  This ini files is used in a non-traditional way.  It is used like a quasi database of boilerplate template items. |
| *OfficeList.ini* | Gets used for the smaller list, at the top of the Definitions tab.  The key names **must** be "ContactInfo" or "SchPhone."  If the TextDefinitions.ini text has [c] in it, the tool uses the ContactInfo as the replacement text.  A [p] gets replaced with just the phone number (SchPhone).   Don't use name or pronoun replacement codes in the OfficeList.ini file.  This ini files is used in a non-traditional way.  It is used like a quasi database of Office directory information. |
| *Settings.ini* | A file that holds several of the preferences.  Don't edit Settings.ini directly.  affect it by using the "wtSettings" tool.  Access wtSettings via the 'Settings' link on the bottom of the main tool or via right-clicking the Win SysTray iton.  The other two ini files get opened and edited directly.  This one does not.   This ini file is used in a traditional orthodox way.  This is what initialization files were actually designed for. |

WayText looks for these ini files, by name, in the sub folder called "wtFiles."  It looks for the wtFiles subfolder in the same location that the WayText.ahk and WayText.exe files are.

**Note about the exe files**

The files "WayText.exe" and "wtSettings.exe" are the executable files that get double-clicked at first run. They are not compiled versions of the corresponding (same-named) .ahk files. They are merely renamed copies of AutoHotkey.exe. More specifically, they are copies of the 64-bit "AutoHotkey64.exe." If a person has a 32-bit Windows PC, they could obtain a copy of AutoHotkey32.exe and make two copies of it, renamed "WayText.exe" and "wtSettings.exe." That should work fine. Please note that these are the version 2 of AutoHotkey. The code will not work with AHK v1.

Please note the different icon images that AutoHotkey v1 and 2 use… Version2 has a flatter look, a yellow/greenish H and a parallelogram shape, and is on the left side of the image. The v1 icon (right side of image, but not the right one to use here), has an embossed look, a white H, and is square with rounded corners.

**Some GUI Elements**

1) The little blue rectangle on the left edge is the "Marker." It indicates which of the four main input controls are focused. It is already easy to detect when the name box or the radio group is focused, because they become outlined. The two list boxes have no such indicator of being focused though. The marker addresses this. The rectangle is simply a Unicode character. Different characters can be assigned. Look for the **markerX := "█"** variable in the WayText.ahk code. The color can also be changed there.

2) The Name Box. Student (patient?) name goes here. Shift+Left focuses the name box. "Student" is the default (can be changed) but the last used student info is remembered for "X" minutes (See Pre-Fill tab, below).

3) The Target Window indicator lets you know which app/window you are about to send your boilerplate text into. "Non/Preferred" windows can be listed in the Settings tool. If a non-preferred window is the target window, then the background color of the GUI will be the "warning color" (not seen in image).

4) The Gender radio group. "Neutral" attempts to automatically correct the grammar of the selected boilerplate text, to make the gender pronouns non-binary. It's not perfect but works pretty well.
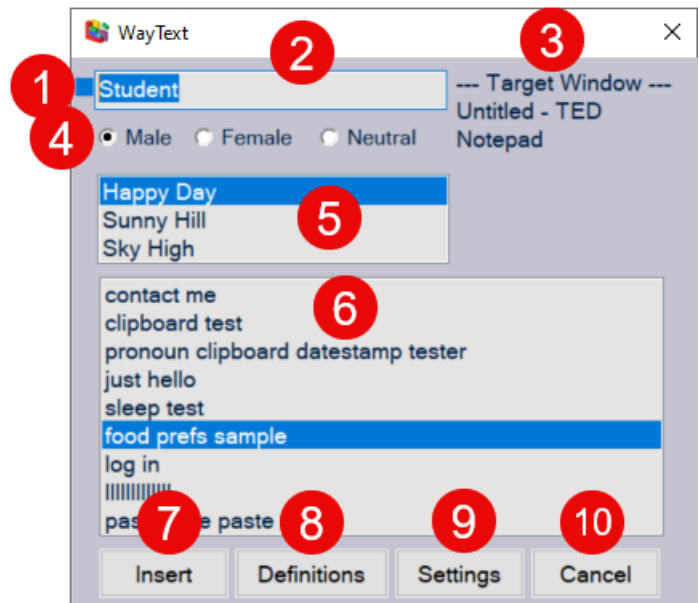
5) Office/Building/School list box. If the boilerplate text has [c] or [p], then office-specific information is used. If not, then the choice is moot.

6) Text Definitions/Boilerplate Library of Entries. Shift+Right focuses this listbox. Use Up/Down Arrows to navigate. The navigation "wraps around," which is to say, if you are at the top and go up, the selection jumps to the bottom. The same thing happens when going down past the bottom. To quickly navigate, press Shift+Up/Down to "skip jump" by X spaces. (X is defined in WayText code, with **skipNum := 5** variable.

7) The Insert button inserts selected Definition. Pressing Enter does the same. Shift+Click opens a "debug" window instead. The debug window shows the content of the ini key names and values.

8) Open the Definitions ini file in your assigned "Favorite Editor." Right-Clicking the button, attempts to open the ini file and search for the selected Definition item.

9) Run the wtSettings tool, where most of the user preferences are set. Settings can also be accessed via right-clicking the Windows SysTray icon (see img). The user can browse to a specific tab in this manner. Tip: Right-Clicking the Settings button is a shortcut to the settings submenu seen in the image on the right.

10) Press Cancel Button (or Esc key) to just hide the form.  Esc will also hide any MiniForms that are showing.

11) In the Windows SystemTray (bottom/right of screen) is the WayText icon.    Right-click that for the Menu.  At the top of the menu is the currently-assigned hotkey for the main form.  Clicking the top menu items pops up a message box with the following tips:

*Main Hotkey: Shift+Alt+W*
 *\*Other hotkeys (for when form is active) \**
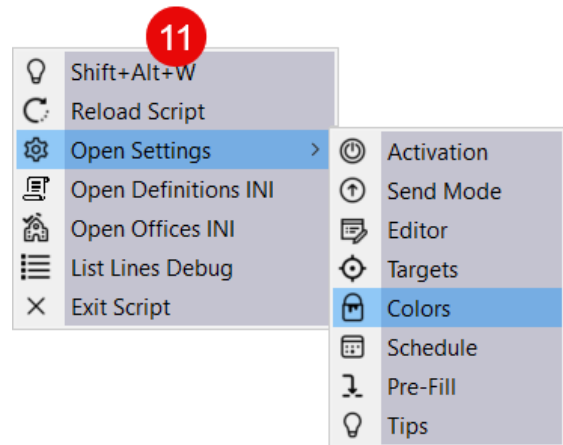*Shift+Up/Down in Definition List does "skip-jump."*
*Skip-jump number set with variable "skipNum", above*
*Right-Clicking Settings Button Attempts to open TextDefinitions.ini and use Find feature of editor to search for whatever Definition is selected in List.*
*Shift+Clicking Insert Button shows debug message of the selected ini Section.*
*Shift+Left to make Name box active.*
*Shift+Right to make Definitions listbox active.*

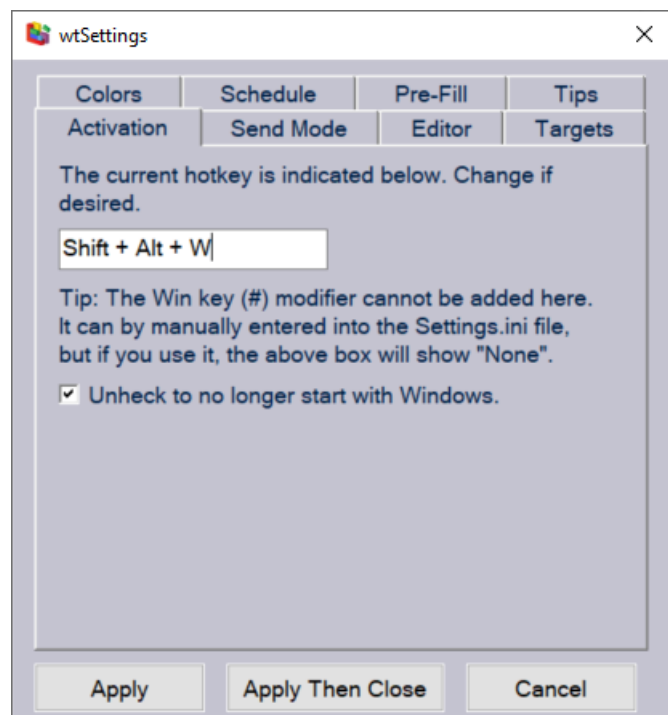| | |
|---|---|
| 🔆 Shift+Alt+W | |
| C⟳ Reload Script | |
| ⚙ Open Settings ＞ | ⊚ Activation |
| 🖥 Open Definitions INI | ⬆ Send Mode |
| 🏠 Open Offices INI | 🖥 Editor |
| ☰ List Lines Debug | ⊕ Targets |
| ✕ Exit Script | 🔏 Colors |
| | 🗓 Schedule |
| | ⅂ Pre-Fill |
| | 🔆 Tips |

**The WayText Settings Tool**

Most of the user preferences are stored in the wtFiles folder, in the Settings.ini file.  The WayText Settings tool (wtSettings) reads from, and writes to, the Settings.ini file, then the WayText Application reads from the same ini file to get the preferred user options.  As such, there are very few preferences that are manually set in the WayText code directly.

<u>Activation Tab</u>

Change the default hotkey if desired.

**Tip:** The Win key (#) modifier cannot be added here.  It can by manually entered into the Settings.ini file, but if you use it, the above box will show "None."  The assigned hotkey appears in the SysTray menu as well.

The check box adds/removes a link in the *Windows/Apps/Startup folder*.  This is the only system file that gets changed.  Neither WayText, nor wtSettings touches the Windows Registry.

wtSettings                                                    ✕

| Colors | Schedule | Pre-Fill | Tips |
|---|---|---|---|
| **Activation** | Send Mode | Editor | Targets |

The current hotkey is indicated below. Change if desired.

Shift + Alt + W

Tip: The Win key (#) modifier cannot be added here. It can by manually entered into the Settings.ini file, but if you use it, the above box will show "None".

☑ Unheck to no longer start with Windows.

Apply            Apply Then Close            Cancel
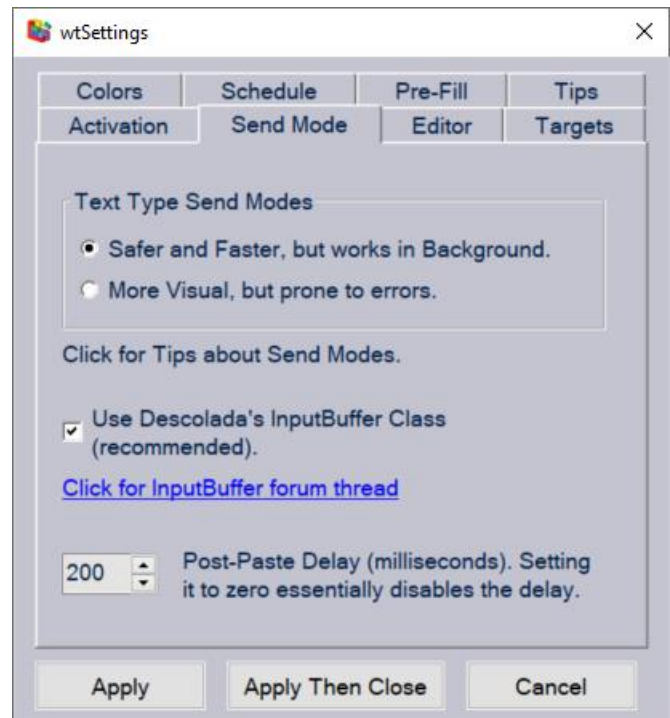
<u>Send Mode Tab</u>

Choose whether WayText should use AutoHotkey's "SendInput" or "SendEvent." From the Tips Message:

====*Text Type Send Modes:*====
*AutoHotkey's "SendInput" is Safer and Faster but works in Background. You might not see the fields in the webform getting updated, but Chrome will update the whole page after the text is sent.*

*"SendEvent" is More Visual, but prone to errors. The keys are send as separate events, so Chrome refreshes after each character.*

*When using SendInput, AutoHotkey tries to buffer any keypresses that occur during the auto-typing of the boilerplate text. This is an imperfect process in Windows though. Descolada's InputBuffer Class addresses this and "captures" extraneous keypresses during auto-typing, then places the captured text. Learn more on the forum thread at the AutoHotkey Forums.*

Windows takes time to send the content of its Clipboard, so AutoHotkey scripts must allow time for this. The "Post-Paste Delay" defines the amount of time used for this purpose. Note that pasting into MS Outlook Emails requires a great deal more time for this, so if the string "Message" is found in the Target Window, then 900 milliseconds are used. This is hardcoded into the WayText code.
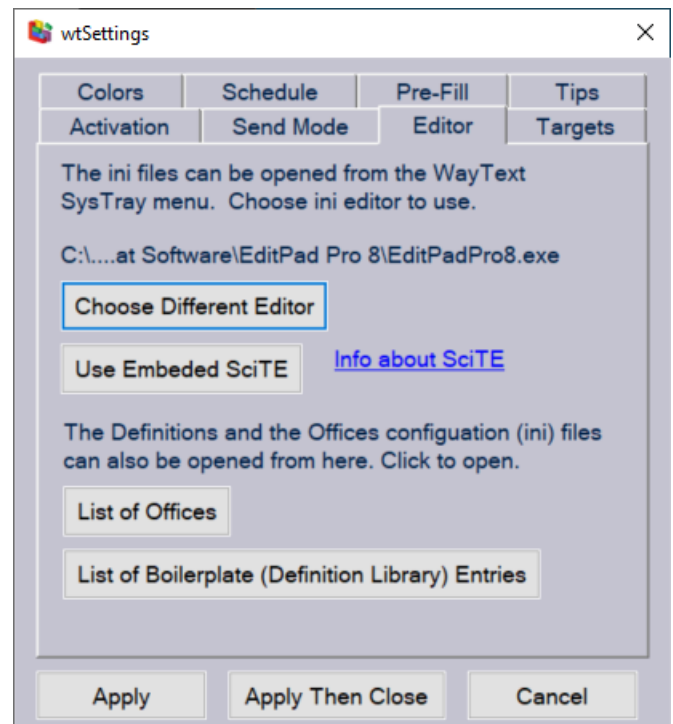
<u>Favorite/Preferred Editor Tab</u>

When opening one of the ini files, WayText or wtSettings will attempt to open the ini in the editor that is assigned here. The path to the currently assigned editor (if there is one) is shown above the "Choose Different Editor" button. Below that is the "Use Embedded SciTE" button. If neither of those are found, Windows will just open the ini as a normal file, prompting the user to choose an app. I recommend using an Editor with syntax highlighting for INI files. SciTE has this. From SciTE help file: "*SciTE distribution designed for AutoHotkey - made by fincs - Original SciTE made by Neil Hodgson*." You can get the excellent *Scintilla Text Editor for AHK (SciTE4AutoHotkey)* from the AutoHotkey website.

At the bottom of the tab are links to the two "database" ini files:

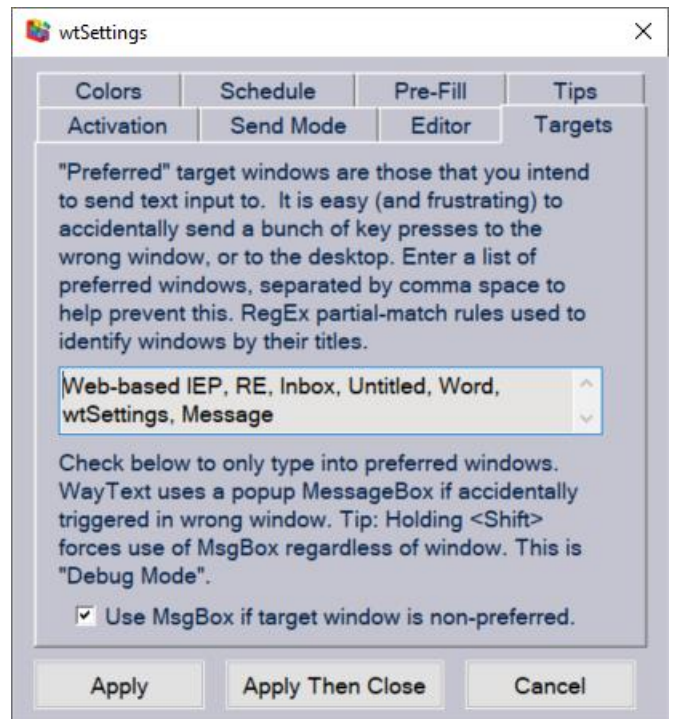-OfficeList.ini
-TextDefinitions.ini

These can also be accessed via Right-Clicking the SysTray icon. The Definitions ini gets accessed far more often, and so has a button on the bottom of the main WayText application window too.

## Target Windows Tab

As indicated in the tab: *"Preferred" target windows are those that you intend to send text input to. It is easy (and frustrating) to accidentally send a bunch of key presses to the wrong window, or to the desktop. Enter a list of preferred windows, separated by comma space. Regular Expression partial-match rules are used to identify windows by their titles.* As seen in the image, the items are separated with " ," (space comma), but the " ," gets saved as pipe characters in the Settings.ini file.

Also from the tab: *Check below to only type into preferred windows. WayText uses a popup Message Box if accidentally triggered in wrong window. Tip: Holding <Shift> forces use of MsgBox regardless of window.*

Related: See Color Tab, below, for visual indicator (I.e. gui warn color) for non-preferred windows.

## Color Tab

Several custom colors can be assigned to the main WayText form. With the four color assignments, a preview of the colors is given on-change.

**Form**: The main Gui Form background color.
**List Boxes**: No list boxes are present in the wtSettings gui, but this color will be applied in the WayText form, in the Offices List and the Definitions List. This is also applies to the background of edit fields, combo boxes, and the "Sample" rectangle seen in the image.
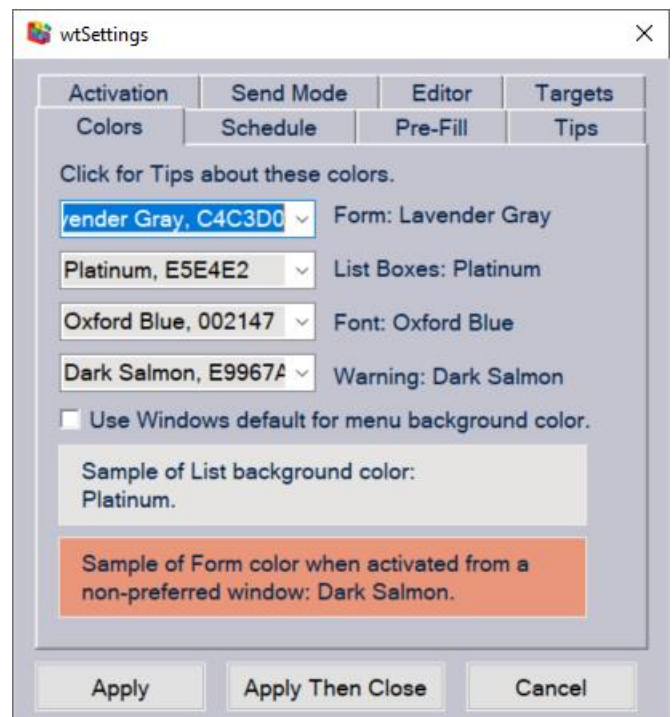**Font:** The color of the words on the gui forms.
**Warning**: The main Gui form color if/when a non-preferred window (as defined in previous screenshot) is the target window.

Scrolling the list of colors will instantly update the preview colors seen in the image. The colors won't be saved to the ini file until one of the "Apply" buttons is pressed though.

Notes: The big list of colors hex codes and names goes from *lighter* at the top, to *darker* at the bottom. The list is at the bottom of the Settings.ini file in its own ini section. If you want a color that is not already there, you can add it manually to the Settings.ini file (carefully using the same format as the others). You can also just paste a custom hex code into one of the four ComboBoxes.

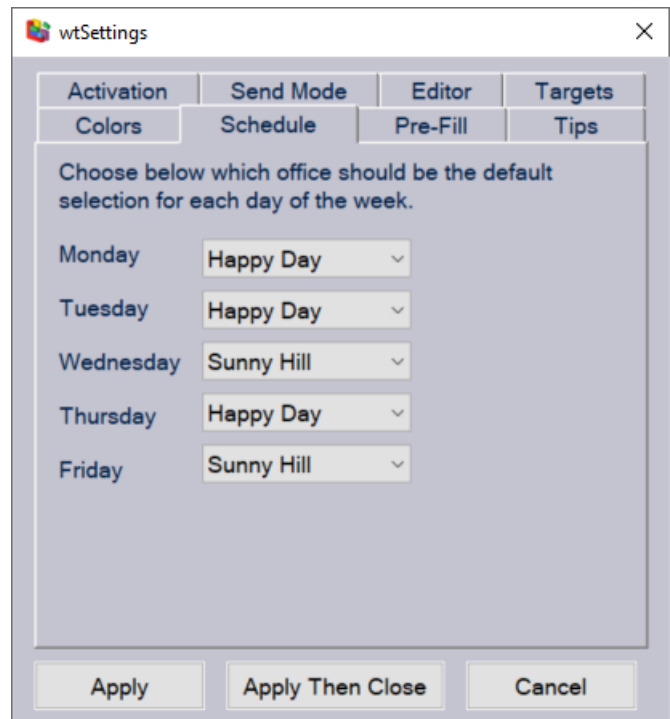The WayText app doesn't actual use the color names… It only uses the hex codes.

## Schedule Tab

If your weekly work schedule has you at particular offices on certain days of the week, you may wish for the relevant office to be preselected in the Offices listbox of the WayText form. Set those here. The wtSettings GUI reads from the OfficeList.ini file for the list items.

Note that the Pre-Fill mechanism (see below screenshot) will take precedence over the default Office that is assigned here.

The Schedule tab only has Monday-Friday, but the code in ini keys exist for Saturday and Sunday too. Just Uncomment-out the pertinent lines of code in wtSettings.ahk. There is a spot at the top (ini reads), two spots in the Schedule Tab code, and one spot at the bottom (the ini writes). The WayText.ahk file doesn't need to be changed at all.
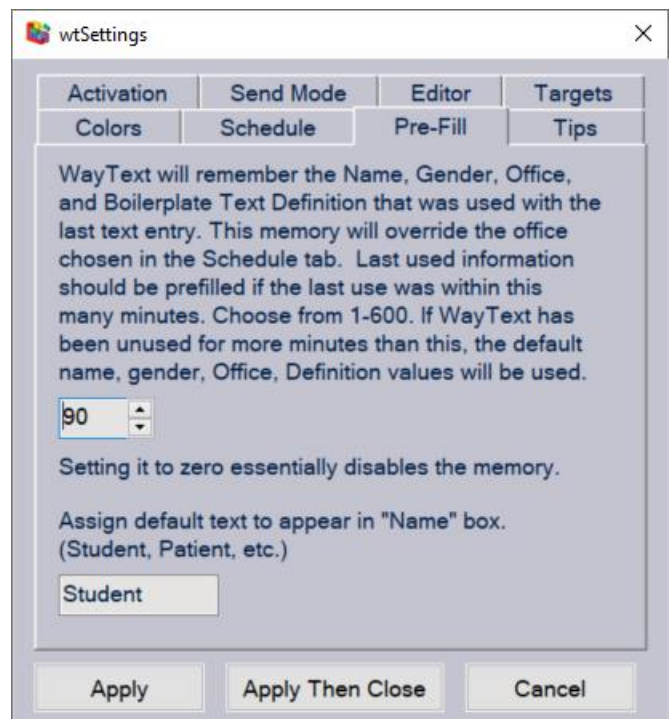
## Pre-Fill Student Data Tab

As indicated in the tab:
*WayText will remember the Name, Gender, Office, and Definition that was used with the last text replacement.*

*NOTE: This memory will override the Office chosen in the Schedule tab.*

*Last used information should be prefilled if the last use was within this many minutes. Choose from 1-600. If WayText has been unused for more minutes than this, the default name, gender, Office, Definition values will be used.*

It is noteworthy that the "last used" timestamp and other info is not saved in the Settings.ini file, so it will not be remember if the WayText.ahk file is restarted in RAM. (And restarting is indeed needed for most of the wtSettings to appear in the wt app. There is a "Reload" option in the systray menu for this very reason.)

<u>Tips Tab</u>

Just some tips.  No settings to change here.  As indicate in the "more tips" popup:

*These are the settings for the WayText application.*

*They save the settings to the Settings.ini file, then WayText reads from the ini file each time it is restated.  As such, WayText.exe gets restarted when setting changes are made here, using the following rules:*

*If these settings were launched from the WayText main form, then Pressing the [Apply] button at the bottom will save the settings, reload WayText, then open its form.*

*Pressing [Apply Then Close] saves and reloads, then closes the Settings form, but does not open the WayText form.*

*If Settings is not launched via WayText, then [Apply] only saves the settings to the ini file, and [Apply Then Close] saves to the ini file, reloads WayText without opening its form, and closes the settings form.*

**Credits and Sources**

Special thanks go to the creators of AutoHotkey and to the many wonderful people on the support forum.   WayText is by far my longest-running AHK project.  Without the help of the forum users, this project would not exist!   Extra thanks go to Descolada for his InputBuffer Class script.  Some of my boilerplate text entry items are pretty long… The InputBuffer ensures that accidentally-pressed keys don't get interspersed with the content.   Proper credit is also due for the SciTE editor that I have embedded.  I currently (mostly) use VSCode, but I used SciTE for many years before that. The WayText icon image is the old Windows Defrag Icon.  The one in the screenshot here was edited to add the wrench. The little mono-color icons in the SysTray image all came from icons8.com.

*~Thanks for reading.  Stephen (Steve) Kunkel (kunkel321)*