

TypeIt

TypeIt – A utility program that finishes your typing for you.

Release 1.0 Version 37.1

Copyright © 1994 Brian Moats

by Brian Moats

Introduction

Do you hate typing out often used words and phrases? Wish the computer knew what you were thinking? Not possible...yet, but **TypeIt** brings it a little closer.

Or maybe you have some words that you always misspell. Like "reciept" instead of "receipt", or "seperate" rather than "separate". Just put these in your dictionary and you'll never even have to *type* them again.

TypeIt uses a dictionary that contains words, phrases and even multi-line phrases. When you type one or more letters and then press a special "Expand Key" hotkey, **TypeIt** tries to match what you have typed with what is in its dictionary. It will then type the rest for you.

If you have more than one entry that matches what you have typed, repeated pressing of the hotkey will delete the characters sent and then send the next entry. The screen will flash when there are no more matching entries (If you have "Make Sound" checked in Sound Preferences, there will also be a beep).

TypeIt learns your typing habits. The more frequently used words move to the front of the list.

TypeIt resets its character memory when you press any non-printable key or the space bar. In other words, if you were typing a word and then pressed the space bar, return, ctrl-x, an arrow key or anything else that does not display a character, **TypeIt** will forget the previous characters.

The dictionary "TypeIt.dict" is just an arbitrary collection of words that is provided simply to demonstrate **TypeIt**. You will want to create your own custom dictionary.

A large dictionary such as a spell checker is not very useful because there are just too many words that begin with the same first few letters. For example, if you typed "calc", you would have all these possibilities:

calcification, calcified, calcifies, calcify, calcite, calcium, calculate, calculation, calculative, calculator, calculus

The dictionaries c.dict and m2.dict are included for programmers. You will probably have to make some modifications to make certain multi-line phrases work correctly with your particular editor.

Installation

Installation is simple. Just place **TypeIt** in your WBStartup drawer and your dictionary wherever you want. Enter its path and name in the Tool Types gadget of **TypeIt**'s icon. For example, `DICTIONARY=TypeIt.dict` will read the dictionary from the same directory that **TypeIt** is in.

TypeIt may be run from the shell or from its icon, but it is usually best to place it in your WBStartup drawer. Running **TypeIt** a second time will bring up its interface window.

From the shell or its icon, the arguments are the same.

CX_POPUP

If "CX_POPUP=yes", **TypeIt**'s interface will open when the program is first run. Normally you will set this to "CX_POPUP=no".

CX_POPKEY

The key combination to use to open the interface window. Use the normal key specification for commodities. Default is "CX_POPKEY=control alt t".

CX_PRIORITY

The priority that this program receives events. Default is "CX_PRIORITY=0".

DICTIONARY

The full path and file name for the dictionary. "DICTIONARY=TypeIt.dict" is the default, which means that the dictionary should be in the same directory as **TypeIt**.

EXPAND_KEY

This specifies the key combination to use to expand what you have typed into what matches in your dictionary. The default is "EXPAND_KEY=shift space".

NOTIFY

If "yes", **TypeIt** will watch your dictionary. If you make changes to it with an editor, **TypeIt** will re-read it so your changes are immediately available. You do not have to "Load" it again. **TypeIt** will not re-read it after saving through the "Save..." gadget; it already knows about your changes. The default is "NOTIFY=yes".

GET_SMART

When "yes", **TypeIt** will change the order that words are selected for expansion. Words used more frequently will be displayed first. The order will be forgotten when **TypeIt** is terminated unless you "Save..." from the interface window. The default is "GET_SMART=yes".

DONOTWAIT

Use this to allow Workbench to continue after starting **TypeIt**.

A shell example:

```
run >nil: <nil: TypeIt "expand_key=shift space" dictionary=s:typeit.dict
```

TypeIt uses the standard commodity key combinations.

Qualifiers

lshift	rshift	shift	lalt	ralt	alt	
lcommand	rcommand	control	numericpad	capslock	caps	repeat

Keys

space	backspace	tab	enter	return	esc
del	up	down	right	left	f1 - f10
help					

If you use "lcommand space", that means you press <left Amiga-space> as a hotkey. To use any normal key, just specify the character on the keycap. For example: "control e", "alt shift t", "rcommand t".

If you use "-repeat", then when the hotkey is held down, each repeat will be considered another hotkey event. If "-repeat" is omitted, the first time will be a hotkey, repeats will then be treated as regular key repeats.

To further clarify: If your hotkey is "shift space", holding that combination down will generate one hotkey event (attempt to match the previous letters with a word in the dictionary) and then shift spaces which will not be hotkeys. If you use "shift -repeat space", then holding that will generate a hotkey event for each key repeat.

Creating Your Custom Dictionary

A custom dictionary is simply a list of words or phrases, one to a line. You may use any text editor or word processor that provides a way to save as normal text. Most, if not all, have some method of doing this. Use the included dictionaries as a guide.

1. Each entry must be less than 255 characters and must begin with a low-ascii character (! to ~), (hex 0x21 to 0x7e). In other words, all letters that are printable without needing the Alt key.
2. Other characters are allowed in the entry (such as <alt-e>), but they will not match when the "Expand Key" is pressed. That means that if given the entry "qw©rty" and the user types "qw", "©rty" will be displayed. But if the user types "qw©" that will not match.
3. "Dead keys" are handled as normal ascii characters. "Dead keys" are combinations that put accent characters on letters. For example, if you press "<Alt-f>e", you will get an accented e.
4. All entries with a common first character will be grouped into a single list with the most frequently used words first.

Given this group:

```
#include ".h"\3L
#include <.h>\3L
#define \
```

"#include ".h"" will be displayed when the user types a single "#", then "#include <.h>" and finally "#define ". (The \3L moves the cursor left 3 characters. The trailing \ for #define insures a space.)

The dictionary does not need to be in alphabetical order, although it might be easier for you to organize if it is. Within an alphabetical group, entries should be arranged by their desired order of appearance. After you "Save..." (from the interface window), word groups will be in alphabetical order with the more frequently used words listed first in each group. All words beginning with "a" will be before words beginning with "b". But, the words beginning with "a" will be in order of their usage, not in alphabetical order.

Given this group:

```
intuition
include:
international
```

"intuition" would be the first choice after typing "in", followed by "include:" and then "international".

It is also possible to use abbreviations like:

```
FSF\3BFree Software Foundation
```

This will delete user typed "FSF" and then send "Free Software Foundation". Once an abbreviation has been typed out, you cannot display more entries because the abbreviation will be deleted. Of course, if you use GNU Emacs, you already have an abbreviation mode.

About case sensitivity

When adding words, remember to only start an entry in upper case if you want it to match only upper case. To match either upper or lower case, start the entry in lower case.

When the user types "Bri" you want that to match "brilliant", but when the user types "bri", you do not want it to match "Brian Moats". This allows the matching of words normally lowercase but may be uppercase at the beginning of a sentence.

About punctuation

Dictionary entries may start with any punctuation character. '#include' is allowed. However, when TypeIt checks for matches, if it does not find a match and there is punctuation in what you have typed, **TypeIt** will attempt to match from the last punctuation character. For example:

Given these dictionary entries:

```
headers/
Conglomeration
```

and you type '/include/h', **TypeIt** will first attempt to match '/include/h'. Failing that, it will try 'h' and type out 'eaders/'.

If you type '"Con', **TypeIt** will first check for entries beginning with '"' and then it will skip the '"' and type out 'glomeration'.

Special Dictionary Features

Sometimes you will want the cursor placed at a specific location or have a Return sent. The arrow, Return, and the Backspace may be sent by prefacing a letter with a backslash (\).

\L ... Left arrow.
 \R ... Right arrow.
 \U ... Up arrow.
 \D ... Down arrow.
 \N ... Return (newline).
 \B ... Backspace.
 \ \ ... Backslash.
 \W ... Warn if multiple lines follow.

More than one keystroke may be sent by inserting a number between the backslash and the letter. "\10L" will send ten left arrows. The dictionaries c.dict and m2.dict provide many examples of this. Both of these dictionaries assume that your text editor has an auto-indent feature turned on.

If \U, \D or \N is used, the "backing up to delete sent characters" is disabled. This is required because **TypeIt** cannot possibly know how your editor functions. For this reason, the "\W" feature is provided. Use this to warn that the phrase is a multi-line one. For example:

```
Brian Moats\W\NPolyglot Software\N884 W. Melrose Ln.\NBoise, ID 83706
```

This displays "Brian Moats" and then a requester will appear allowing you to abort the execution.

Many text editors do not save trailing blanks. If you want blanks at the end of a word, you may use a backslash alone. E.g. "Telephone \".

The backslashed letters are not case sensitive.

The Interface Window

TypeIt's interface window may be opened by:

1. Pressing the key combination specified in the CX_POPKEY Tool Type.
2. Running the Exchange program, selecting **TypeIt**, and clicking on "Show Interface".
3. Running **TypeIt** a second time.

From the interface window, you may change the options:

Get Smart

With this selected, **TypeIt** will change the order that words are selected for expansion. Words used more frequently will be displayed first. The order will be forgotten when **TypeIt** is terminated unless you "Save..." from the interface window.

Notify

With this is selected, **TypeIt** will re-read the dictionary after you edit and save it. *Note!* Many text editors create backup files. If yours does, it probably will rename the older file which is the file that **TypeIt** is watching. **TypeIt** will not know that you made any changes. Use your editor's option to disable the creation of backups before saving.

After adding words to your dictionary from the interface window, you should click on "Save..." (which will open a file requester) and save your words. You should also save if you want to preserve what **TypeIt** has "learned" about your word order.

Clicking on "Load..." will open a file requester. Select the desired dictionary file.
Clicking on "Hide" or the close gadget will close the interface window.
Clicking on "Quit" will remove **TypeIt**.

Adding Words

New words may be entered into your dictionary by simply editing the dictionary file with an editor. Words may also be added on the fly through the interface window.

Suppose that you had just typed "configuration" for the millionth time and decided that you might as well add it in. Simply press the CX_POPKEY key combination (default: control alt t). The interface window will open with the word in the "New Word" string gadget. Modify it if you like and click on "Add". That word will be added to the head of the word list. It will be forgotten when you terminate **TypeIt** unless you also click on "Save..."

You do not have to have just typed the word that you want to add. You can open the interface window and enter a word in the gadget anytime.

If the window is already open when you press the CX_POPKEY, the letters that you had just typed will be placed in the string gadget. The same is true if you simply make the window active.

Shareware and Registration

TypeIt is being distributed as "buckware". This is the same as shareware, but with the low low price of just \$1. Obviously, buckware needs many users. How difficult can it be to slip a buck into an envelope? (Send more if you like.) Use it for "awhile" then register it if you want to continue using it. With enough users sending in just 1 dollar, more development will go into **TypeIt**. You may also make any suggestions and requests when you register.

Send your \$1.00 registration fee to:

Brian Moats
Polyglot Software
884 W. Melrose Ln.
Boise, ID 83706

Also available is MachV Release 5.0 Version 37.8. This program is a combination of many utility programs.

1. Keystrokes and mouse movements may be recorded and played back at the touch of a key or the click of a gadget.
2. Run programs or open a new shell at the touch of a key. Or move screens and windows.
3. Quickly view the contents of the clipboard. The contents may be pasted (even to windows that do not support the clipboard) or printed. Files may be loaded directly to the clipboard or saved from the clipboard.
4. Screen blanker/dimmer and mouse blanker.
5. Full ARexx interface. Send commands to other ARexx compatible programs and control MachV from ARexx programs.
6. Optional title bar clock. Gadgets may be added to the clock which may then execute macros.

This program is just \$15.00. A demonstration version is also available. The demonstration version is fully functional...it just has a reminder window when first run and a limit of 25 macros. It is available on many BBSs or send either a disk and SASE or \$2.00 and I'll send it to you.

Copyright and Disclaimer

TypeIt and this documentation are © 1994 by Brian Moats.

This product is sold "as is" without representation or warranty of any kind, either expressed or implied. The user assumes all risks and responsibilities related to its use. **TypeIt** was originally named Typhoon and released July 5, 1992 for JumpDisk.

This documentation was written in "Texinfo mode" using GNU Emacs. It was then run through "MakeInfo" to create ASCII, "AmigaGuide", and T_EX versions.