

onScreenTyping v2.3

Simulating keyboard input; by Faguss (ofp-faguss.com)

1. Requirements:

Fwatch 1.16

<http://ofp-faguss.com/fwatch/download>

2. Installation

This script is a part of Fwatch Standard Library (since Fwatch 1.13) so you don't have to copy it to the mission directory. Simply execute it:

```
{ <options> } exec "..\fwatch\data\onScreenTyping.sqs"
```

3. Quickstart

Suppose you have a dialog control to which you want to add input functionality. Use the code below:

```
OST_ISRUNNING = false
{_isRunning="OST_ISRUNNING"; _idc=10} exec "..\fwatch\data\onScreenTyping.sqs"
@OST_ISRUNNING
@!OST_ISRUNNING
```

4. Usage

User can type all characters visible on a standard English keyboard. There's no hardcoded character limit. Hold SHIFT or enable CAPS LOCK to type uppercase.

User can scroll through text by using arrow keys, CTRL+arrow (to skip words), HOME (jump to start) and END (jump to the end).. Current position is indicated by blinking cursor.

Characters are deleted with BACKSPACE and DELETE. Words are deleted with CTRL+BACKSPACE or CTRL+DELETE. Text can be marked by holding SHIFT (unfortunately it won't be visible) and then modified or copied to a Windows clipboard with CTRL+C or CTRL+X. To paste text press CTRL+V or SHIFT+INSERT. CTRL+A will mark the entire text.

Editing can be finished by pressing ENTER or ESC. With the latter, the original text will be brought back.

5. Options

Script will execute passed string so it can contain code that changes internal variables in the *onScreenTyping* script. Here's a list of these variables:

`_input`

Initial text. Accepts string, number (dialog control `idc`) or array with strings (will be merged). Default is empty string.

```
_input="text"  
_input=["one", "two", "three"]  
_input=10
```

`_idc`

Dialog control number. Script will output current text there. This argument also changes other variables: `_input=ctrlText _idc` and `_quit="!dialog" and _supress="!ctrlVisible _idc"`. You can modify them after `_idc`. Default value is -1 (disabled).

```
_idc=10
```

`_outputSTR`

Global variable name. Script will copy currently visible text to this variable. Default is empty string.

```
_outputSTR="OST_DISPLAYTEXT"
```

`_workSTR`

Global variable name. Script will copy full text (`_outputSTR` can be limited by `_displayLimit`) and without a cursor to this variable. Changing content of this global variable programatically will modify the working text in the *onScreenTyping*. Default is empty string.

```
_outputSTR="OST_FULLTEXT"
```

_isRunning

Global variable name. Used to determine if the script is working. Script will not launch if this variable is set to `true` prior to executing it. If not then it will change it to `true` once the script is ready to take user input. It is changed to `false` when script stops running. Default is empty string.

```
_isRunning="OST_ISRUNNING"
```

_limit

Integer. Determines maximal text length. Default is -1 (unlimited).

```
_limit=100
```

_displayLimit

Integer. Determines how many characters will be shown. Text will be scrolled when cursor reaches the edge. Default is -1 (unlimited).

```
_displayLimit=50
```

_cursor

String. Used to indicate current position in the text. Default is underscore. If you change it to empty string then the user won't be able to scroll through text.

```
_cursor="|"
```

_cursorBlink

Float. How often cursor blinks in seconds. Default is 0.5. Use 0 for constant display.

```
_cursorBlink=1
```

`_allowMarking`

Bool. Allows user to mark text with SHIFT. Default is `true`.

```
_allowMarking=false
```

`_interval`

Float. Script sleep time. The higher value the slower the script will work. Default is 0.001.

```
_interval=0.02
```

`_notAllowed`

Array with characters. List of characters that user cannot type. Default is empty array.

```
_notAllowed=["<", ">", ":", "''", "/", "\\", "|", "?", "*"]
```

`_escapeKeys`

Array with strings. List of keys that end editing and revert to the original input. See [Fwatch documentation](#) for key names. Default is `["ESC"]`.

```
_escapeKeys=["TAB", "F1"]
```

`_acceptKeys`

Array with strings. List of keys that end editing and keep changes. See [Fwatch documentation](#) for key names. Default is `["ENTER"]`.

```
_acceptKeys=["F12"]
```

_suppress

Code string with condition. When evaluates to *true* then the script suspends typing functionality without quitting. Default is "false".

```
_suppress="OST_SUPPRESS"
```

_quit

Code string with condition. When evaluates to *true* then the script ends and reverts changes. Default is "false".

```
_quit="OST_STOP"
```

_quitAndSave

Code string with condition. When returns true then script ends and keeps changes. Default is "false".

```
_quitAndSave="OST_SAVE"
```

_InfoArr

Global variable name. Default is empty string. Script will copy array to this variable containing the following information:

- 0 – cursor position
- 1 – scroll amount
- 2 – display limit
- 3 – mark start position
- 4 – mark end position
- 5 – empty string (see `_InfoMarkText`)
- 6 – text length

```
_InfoArr="OST_INFO"
```

_InfoMarkText

Bool. When set to `true` then the script outputs marked text to the info array. Default is `false`.

```
_InfoMarkText=true
```

_key

Arrays with strings. Converts selected key into given string. See [Fwatch documentation](#) for key names. First sub-array contains input keys and the other one – output. Default is empty array.

```
_key=[[ "TAB", "F1" ], [ " ", "ä" ]]
```

_keyCTRL

Similar to `_key` but works when CTRL is held. It's evaluated after a key name is converted to a letter and after `_key`.

```
_keyCTRL=[[ "F1" ], [ "<<" ]]
```

_keyALT

Similar to `_key` but works when ALT is held. It's evaluated after a key name is converted to a letter and after `_key`, `_keyCTRL`.

```
_keyCTRL=[[ "A" ], [ "ä" ]]
```

_keySHIFT

Similar to `_key` but works when SHIFT is held. It's evaluated after a key name is converted to a letter and after `_key`, `_keyCTRL`, `_keyALT`.

```
_keySHIFT=[[ "ä" ], [ "Ä" ]]
```

6. Version history

1.0 (21.04.2009)

First release.

1.1 (31.10.2009)

- implemented continuous writing

1.2 (28.06.2010)

- added global variables
- added user defined replacement
- blocked user control when script is active
- implemented save function
- prevented crashing by dividing message into buffers
- script's performance depends on benchmark

1.21 (05.07.2010)

- added global variables for character replacement with *SHIFT*
- fixed clearing message function
- fixed saving issue if filename contains space

1.3 (02.11.2009)

- compatibility with Fwatch v1.1
- removed global arrays for customization

2.0 (01.08.2014)

- now uses Fwatch v1.13 and is a part of it's library
- is based on array with characters instead of string buffers
- added many new options (custom arguments)
- implemented cursor
- can paste text from clipboard
- removed hotkeys for functions (freezing script, clearing text, saving text)
- doesn't use benchmark command anymore (just constant time interval)

2.1 (31.05.2015)

- uses Fwatch v1.14
- added arguments `outputArr`, `key`, `keySHIFT`, `keyCTRL`, `infoARR`
- removed arguments `inputIDC`, `outputIDC`, `acceptKeys`, `escapeKeys`

- argument `cursor` – display frequency can now be changed
- argument `globalString` was changed to `outputSTR`
- argument `globalArray` was changed to `workARR`
- argument `isRunning` wasn't working – fixed
- `isRunning` variable now must be initially `false` otherwise the script won't work
- argument `limit` can now limit number of displayed characters (implemented scrolling)
- argument `input` recognises variety of values (string, number, array with strings)
- argument `quitCond` was changed to `quit`; now it can give condition to quit and keep changes
- argument `quitKeys` has now functionality of former `acceptKeys` and `escapeKeys`
- quit conditions are now checked before suppression condition
- certain variable can be modified from outside the script using `info` array
- characters can now be marked with Shift key
- improved word skipping (Ctrl + Arrow)
- pasting from clipboard will add all text (instead of just one line)
- marked text can be copied to the clipboard (via marking)
- Shift + Insert can be used to paste text from clipboard

2.2 (01.06.2016)

- uses Fwatch 1.15
- now operates on a string instead of array with characters (faster)
- arguments are now passed as a code string instead of using nested arrays
- removed arguments `outputARR`, `outputIDC`, `workARR`
- added `workSTR` argument
- limit of visible characters is now set by `_displayLimit` instead of `limit` array
- cursor duration is set by `_cursorBlink` instead of `cursor` array
- marking can be turned off by `_AllowMarking` instead of `cursor` array
- keys that quit&save are now set by `_acceptKeys` instead of `quitKeys` array
- quit&save condition is now set by `_quitAndSave` instead of `quit` array
- quit conditions can now use vars `_keys` (asynchronous), `_keys2` (synchronous), `_shift`, `_ctr`, `_caps`
- mark text is empty in the `info` array by default; use new argument `_InfoMarkText` to add it
- arguments `_escapeKeys` and `_acceptKeys` are now checked by `FLIB_INKEYS` so a combination can be used
- argument `supress` was renamed to `suppress`

2.21 (18.09.2025)

- uses Fwatch 1.16

2.3 (29.03.2026)

- CTRL+DELETE and CTRL+BACKSPACE now delete words
- CTRL+A will now mark the entire text
- added `_keyAlt` input variable

- updated demo mission
- changing length in `_InfoArr` now does nothing
- changing `_workSTR` from outside the script now changes the working text
- fixed crashing with strings above 2040 characters
- fixed syntax error when using `_InfoArr`
- fixed issue where text would be scrolled when reached string limit and user keeps pressing keys
- fixed issue where text would be truncated when reached string limit and user keeps typing in the middle (now typing is blocked)
- fixed issue where script would immediately finish working when ENTER (`_acceptKeys`) was held during launch
- fixed issue with `_quit` and `_suppress` not working after setting `_idc`
- fixed issue with `_keysCTRL`, `_keysSHIFT` not working when using non-letter keys
- script will now scroll back properly when removing last visible character from the end