**[W15-2C][V0.5]**

**Tasuke [助]**



|  |  |  |
| --- | --- | --- |
| Fazli SapuanTechnical LeadCode quality | Tan Li BoonIntegration leadTesting | Chng XinniDeadlines & DocumentationUI lead |



The User Manual

*of*

**Tasuke [助]**

Table of Contents

[An Introduction 1](#_Toc385269974)

[ System Requirements 1](#_Toc385269975)

[ Your first run 1](#_Toc385269976)

[ User Interface and Start Up View 2](#_Toc385269977)

[Quick Start Guide 3](#_Toc385269981)

[ Navigation 3](#_Toc385269982)

[ The Basic Commands 3](#_Toc385269988)

[Complete User Manual 5](#_Toc385269989)

[ Command formats explanation 5](#_Toc385269990)

[ Valid datetimes 5](#_Toc385269995)

[ Color coded commands 6](#_Toc385269996)

[ 1. Show or find tasks 6](#_Toc385269997)

[ 2. Add a task 7](#_Toc385269998)

[ 3. Edit a task 8](#_Toc385269999)

[ 4. Remove a task 10](#_Toc385270000)

[ 5. Mark as done 11](#_Toc385270001)

[ 6. Undo/Redo 12](#_Toc385270002)

[ 7. Exit program 13](#_Toc385270003)

[Summary table of commands 14](#_Toc385270004)

[Useful Features of Tasuke 16](#_Toc385270005)

[ Settings 16](#_Toc385270006)

[ Getting help 16](#_Toc385270009)

[Appendix 17](#_Toc385270010)

[ Appendix 1 - User Stories 17](#_Toc385270011)

[ Appendix 2 - Product Survey 19](#_Toc385270012)



# An Introduction

Your new personal assistant is here to save the day.

*Tasuke* is Japanese for “assistance”, which defines the objective of this program.

Tasuke combines the appeals of an elegant user interface with a powerful command interpretation engine to form a flexible task scheduler. Tasuke is designed to be as unobtrusive as possible - it folds itself away neatly into the system tray when not in use and can be quickly called to attention with a simple key combination. It is also easy to use - you will never feel lost with the help of our tooltips, color highlighting and tutorials. It is personalized - the features and look of Tasuke are all customizable!

##  System Requirements

Tasuke was designed to run on Windows 7 and above, with at least a Pentium III processor, at least 512MB of RAM and approximately 20MB of free space.

##  Your first run

Upon running Tasuke for the first time, the **slideshow tutorial** will be shown. From it, you can learn how to navigate Tasuke and familiarize yourself Tasuke’s command formats.



##  User Interface and Start Up View



### Sorting order of tasks

Tasks are displayed by urgency. The order of the tasks displayed will be overdue tasks, tasks due today, tasks with deadlines, and lastly, tasks without deadlines. These tasks are separated into different subsections.

### Getting full task details

To view full details of the task, simply hover over it. A tooltip displaying the full description, dates and tags will appear.

### Entering commands

When you enter commands in the command box, the tooltip guide will guide you through the format. Your keywords and spelling errors will also be highlighted. You will never enter the wrong command again!

# Quick Start Guide

##  Navigation

###  Show command box

 Open the command box by pressing Ctrl + Space.

###  Show task list window

 Open the task list window by pressing Alt + Space.

###  Show both

 Open both of the above by pressing Ctrl + Alt + Space.

 *All the above hotkeys work system-wide!*

###  Scrolling through your tasks

* You can use the cursor keys (up/down arrow keys) to scroll through **individual** entries in your task list.
* If you want to scroll through **whole pages** of tasks, use Ctrl + up/down arrow keys.
* To jump through **entire subsections**, use Shift + up/down arrow keys.

###  Exit Tasuke

 You can quit by entering ‘exit’ as a command or through the tray icon menu.

##  The Basic Commands

Here are Tasuke’s basic commands. For a full explanation of the commands, you can look at our complete user manual below.

|  |  |  |
| --- | --- | --- |
| **#** | **Command** | **Description** |
| **1** | **show** <search criteria> | This command is used to show specific items in the task list. You may request to show tasks containing a certain keyword, property, date or tag. **Example**: show done*(this will show all done tasks)* |
| **2** | **add** <my task> | This command is used to add tasks into Tasuke. You can specify dates, times and #tags. **Example**: add do homework*(this will add the task “do homework”)* |
| **3** | **edit** <task number> <something> | This command is used to edit existing tasks in Tasuke.**Example**: edit 1 do dishes*(this will change task 1 to “do dishes” instead)* |
| **4** | **remove** <task number/group> | This command is used to delete existing tasks in Tasuke**Example**: remove 1*(this will remove task 1)* |
| **5** | **clear** | Clears all your tasks |
| **6** | **undo** or **redo** | Undo or redo your last action. |
| **7** | **done** <task number/group> or**not done** <task number/group> | Mark unfinished and finished tasks. |
| **8** | **help** | Access Tasuke’s tutorial |
| **9** | **settings** | Access Tasuke’s settings |
| **10** | **exit** | Exit Tasuke |

#  Complete User Manual

##  Command formats explanation

The commands will be explained in the following manner:

### Format

* + This describes how a user command is formatted
	+ The following format syntax notation is used to describe the format:
		- **<>**: these are necessary options for it to be a valid command.
		- **[]**: these are optional and the command will still run without these options**.**
		- **…:** so on and so forth.
	+ This notation is **not** part of the command. i.e. you should not type the notation into the command.

### Description

* + This tells you what the user command does

### Synonyms

* + This tells you alternative words you can use in place of other words

### Examples

* + This is a list of examples and variations of the command

##  Valid datetimes

Tasuke accepts over 30,000 datetimes. For the purpose of this document, a **datetime** is a string that contains either a date, a time or a combination of both (in any order). If a date is not specified, then it will refer to today. If a time is not specified, then it will refer to 0:00 if it is a start date and 23:59 if it is an end date. Examples of valid datetimes recognized by Tasuke:

* [date] [time]
* 5pm
* today
* 10 Feb
* thursday 8am
* tomorrow 3pm
* 14 Feb 2014 5.30am
* 29/2/14 23:59

##  Color coded commands

**If any text is going to be treated as meaning as part of a command, it will be highlighted in a different color.** If you do not intend to use the text as part of a command you can escape the text by prepending a backslash before the sequence of characters that normally have meaning in a command. Extraneous whitespace will be trimmed from the command and will not affect how the command works.

##  1. Show or find tasks

Format:

Any of the following:

**show**

**show** <search criteria>

**show done**

**show undone**

**show overdue**

**show ongoing**

**show today**

**show tomorrow**

Description:

**The Main Part**

This command is used to show the task list if the task list window is closed. If the task list window is already open, this command will bring the task list to the front of the desktop and bring the task list into focus. A corresponding **task number** is displayed for every task in the list. This task number is used in other commands.

**Searching for stuff**

You can search for tasks by specifying a search criteria. The results will be displayed in the task list window.

Synonyms:

show = display, list, ls, find, search

Examples:

show

find dog

search my car keys

search #rekt

##  2. Add a task

Format:

**add** <task description> [**@**[<start> **-** ]<deadline>] [**#**<tag> [**#**<tag> [...]]]

Description:

**The Main Part**

This command is used to add tasks into the task list. The new task will be added at the bottom of the list. It will also be given a unique **task number**. This task numberis used in other commands. The **task description** must be given with the command. If there is no task description, an error will be displayed to you.

**Time Periods**

You can optionally specify a **time period** for the task. You do this by using the **@** character followed by a time period. To separate the two datetimes in your time period you must used the **-** character. If you do not specify the start datetime, then you do not need the **-** character and it is assumed that the datetime is only the **deadline**. If the datetime cannot be understood by Tasuke, an error will be displayed to you. If more than one **@** appears in the command, an error will be displayed to you (if you want to use the character **@** in your description, use a backslash like so: ‘**\@**’).

**Tags**

You can optionally specify one or more **tags** for the task. You do this by using the **#** character for each tag followed by a **tag name**. A tag name may not contain whitespace. If there is whitespace in the tag name, an error will be shown to you. If you give a duplicate tag to the task, it will only be tagged once per unique tag.

Synonyms:

add = a, create, do

@ = at, by

@ … - = from … to

Examples:

add do homework

create do dishes by today 7pm

add browse reddit from 9 Feb to 10 Feb

add buy chocolate @10 Feb - 13 Feb #valentines

##  3. Edit a task

Format:

**edit** <task number> [new task description] [**-@**] [**@**[<start> - ]<deadline>] [[**-**]**#**<tag> [[**-**]**#**<tag> [...]]]

Description:

**The Main Part**

This command is used to edit existing tasks in the list. A **task number** must be specified to identify the task you are editing. The task numbercan be retrieved from the **show** command.

**Adding/Removing Information**

All syntax used to specify description, date and tags in the add command is valid in this command. If you want to remove the time you can use **-@** and if you want to remove a tag simply prepend the tag with a **-** (e.g. **-#tag**).

**Warnings**

If at the end of the entire command the task is not edited, a warning will be displayed to you in a message bubble in the system tray notifying you that nothing was changed.

Synonyms:

edit = e, change, modify, update

@ = at, by

@ … - = from … to

-@ = no time

-# = no tag

Examples:

edit 1 do homework

change 1 by today

edit 1 #homework

modify 2 -@

edit 2 no time

edit 3 -#valentines

edit 1 do homework @9 Feb #homework

##  4. Remove a task

Format:

Any of the following:

**remove** <task number>

**remove** <from task>-<to task>

**remove** <first task>[,<second task>[...]]

**remove** <group of tasks\*> \**all/done/overdue/ongoing/today*

**clear**

Description:

**The Main Part**

This command is used to remove existing tasks in the task list. **Task number(s)** or a **group** must be specified to identify the task, a range of tasks or a list of tasks you are removing. The task number can be retrieved from the **show** command. If the task(s) does not exist, the command continues normally.

**Remove All and Done**

Alternatively, you can specify **all** to clear the entire task list or **done** to clear the tasks that are done.

**Clear**

**Clear** is a contextual command that removes all tasks in the current view.

**Warnings**

Once the task is deleted, all tasks will be given a running number, so some tasks might have a different number after deletion.

This is a destructive command, so please be aware that the **undo** command is useful when you accidentally delete tasks.

Synonyms:

remove = r, rm, delete

- = to

, = (whitespace)

Examples:

remove done

remove 1

rm 5-10

remove 2 to 4

remove 1,3,5,7

rm 1 2

##  5. Mark as done

Format:

Any of the following:

**done** <task number>

**done** <from task>-<to task>

**done** <first task>[,<second task>[...]]

**done** <a group of tasks\*> \**all/overdue/ongoing/today*

**undone** <task number>

**undone** <from task>-<to task>

**undone** <first task>[,<second task>[...]]

**undone** <a group of tasks\*> \**all/overdue/ongoing/today*

Description:

This command is used to mark existing tasks in the task list as done or undone. **Task number(s)** or a **group** must be specified to identify the task, a range of tasks or a list of tasks you are removing. The task number can be retrieved from the **show** command. If the task(s) does not exist, the command continues normally.

Synonyms:

done = d, finished, completed

undone = nd, not done

- = to

, = (whitespace)

Examples:

done 1

d 1,3

undone 4-6

done overdue

##  6. Undo/Redo

Format:

Any of the following:

**undo**

**redo**

**undo** <number of tasks\*> \**up to 10*

**redo** <number of tasks>

Description:

This command is used to undo and redo commands that change the task list. If there is nothing to undo/redo, nothing will happen. You can use the Ctrl+Z and Ctrl+Y shortcuts for this command. You can also specify a number of tasks to undo or redo, capped at a maximum of 10.

Examples:

redo

undo 5

undo max

##  7. Exit program

Format:

**Exit**

Description:

This command exits the program. The exit is immediate and you will not be prompted for confirmation.

Synonyms:

exit = quit

# Summary table of commands

|  |  |  |
| --- | --- | --- |
| Command | Synonyms | Features and Formats |
| show | lsdisplayfindsearchlist | showshow <keyword\*> \**part of description, date, or tag.*show doneshow undoneshow overdueshow ongoingshow todayshow tomorrow |
| add | acreatedo | **Simple**: add <description>**Timed**: add <description> from <start> to <end>**Deadline**: add <description> by <deadline>**Tagging**: add <description> #<tag1> #<tag2> ... |
| edit | eupdatemodifychange | **Update**: edit <task number> <desc/datetimes/tags>**Remove property**: edit <task number> <-datetimes/tags> |
| remove | rmdeleteclear | **Single**: remove <task number>**Range**: remove <from task> - <to task>**Multiple**: remove <task 1>, <task 2>, …**Group**: remove <done/overdue/ongoing/today>**All**: remove all or clear |
| undo, redo | u, r | **Once**: undo**Multiple**: undo <number of times\*> *\*up to 10*  |
| done,not done | d,nd | **Single**: done <task number>**Range**: done <from task> - <to task>**Multiple**: done <task 1>, <task 2>, …**Group**: done <all/overdue/ongoing/today> |
| help | tutorialguideinstructions |   |
| settings | options |   |
| exit | qquit |   |

# Useful Features of Tasuke

##  Settings



### Access Tasuke’s settings

Enter *‘settings’* into the command box.

### What can I do with settings?

* Choose to enable or disable features.
* Change your display font.
* Choose your favourite tooltip guide.
* Select the look of Tasuke from 6 different themes.

##  Getting help



New users need not worry! A **slideshow tutorial** on how to use Tasuke will be shown on first run. Simply enter the command “*help*” to go through the tutorial again.

# Appendix

##  Appendix 1 - User Stories

**User stories about the user interface**

* As the user, I would like to be able to change the color scheme of the task scheduler, so that it looks better on my screen.
* As the user, I would like to be able to access a help or tutorial screen so that I can further familiarize myself with the program.
* As the user, I would like the user interface to guide me on the command format as I type, so I do not need to always remember the exact command syntax.
* As the user, I would like to keep track of tasks that are done or otherwise, so that I know what I have been doing, and what I must do next.
* As the user, I want an indicator of urgency on overdue tasks, so that I will get to them immediately.
* As the user, I want some animations in the window, as it improves feedback on actions that I have momentarily taken.

**User stories about features**

* As the user, I want my tasks to be sorted automatically by date-time, tags, descriptions, or some other identifying means, so that I can order it as I wish.
* As the user, I would like to set my own hotkeys for Tasuke, as the existing hotkeys interfere with other functions in my other programs.
* As the user, I would like to change the details of a task, because I have tasks and events that are possibly mutable.
* As the user, I would like to find the nearest free time that I have, so that I can schedule a next task for that time..
* As the user, I would like to give tags to tasks to further classify them, or so that I can search for groups of tasks that are related to each other.
* As the user, I would like to be able to import or export tasks as I move between computers.
* As the user, I want to scroll through the list of tasks easily without having to part my hands from the keyboard.
* As the user, I want the input window to indicate to me that I have made a spelling error, because I am a spelling-conscious person.
* As the user, I want the task scheduler to run on startup, so that I do not have to manually run it every time I turn on my computer.
* As the user, I want some form of indication of ongoing tasks, so that I know what I should be doing at that moment.

**User stories about commands and input**

* As the user, I want to quickly call the task scheduler with a simple keystroke, and minimize it similarly.
* As the user, I want to see a summary of all upcoming tasks with a simple command.
* As the user, I would like to undo a previous action, as I have made a mistake with that action.
* As the user, I would like to redo a previous action, as I realize that the previous action was not a mistake.
* As the user, I want to see some form of indication of command keywords as I type, so that I do not accidentally enter a keyword as part of the description. This can be achieved through syntax highlighting.
* As the user, I want to add an item from the clipboard into the task scheduler without having to spell out a command for it, so that I can simply copy and paste.
* As the user, I want to use commonly used spelling shortcuts as I use them frequently.
* As the user, I would like to see tasks and events happening today and tomorrow with a simple command as opposed to searching for it by setting a start and end date-time.

## Appendix 2 - Product Survey

We have selected the following products as part of our market research. The following are our experiences with them.

### Microsoft Outlook

Outlook is a popular personal information manager from Microsoft, and it is available as part of the Microsoft Office productivity suite. It is mainly used as e-mail application, it can also natively serve as a task manager, contact manager and calendar.



Although it is a powerful tool with many features, it does not really fit into the needs that Jim needs. It takes too long to create an event; there are too many options to fill in; there is no way to set the frequency of reminders, which Jim thinks is arbitrary and therefore annoying. Furthermore, there is no autocomplete. There is also no quick way for Jim to summon Outlook and then dismiss it when he has completed input.

The difference between Outlook and our product is that our product aims to solve the abovementioned deficiencies that Outlook has, while trimming it down to lightweight. As the calendar is part of Outlook, it also must bear all of the bulk of Outlook.

###

###

### Google Calendar

Google Calendar is a task manager and calendar from the search giant. As part of Google’s online productivity suite, Calendar follows Google’s philosophy of minimalism in their services, and it is evident from the interface of Calendar.



Though Google Calendar is somewhat close to our ideal, it is still not quite there. Being a web-based service, Google Calendar becomes unavailable the moment the user goes offline. This can happen often for road warriors on their laptops when travelling by flight or sea. This is, in our opinion, a glaring flaw. Google’s services assume that its users has an always-on Internet connection, and while this may be true for many users, we want to pre-empt the possibility of the user going offline.

###

###

### Fantastical for Mac

Fantastical is a personal task and event manager only available for Macintosh OSX. It features natural language processing in a neat user interface and events can be created and updated in an all-in-one bar. It also docks neatly into the system tray and can summoned with a key combination or mouse click.



While Fantastical appears to fit the user case very well, the downside is that it is only available for the Apple OSX platform. As our target demographic largely includes office users running Windows, Fantastical is not a viable solution. We aim to build a program that provides many of the features that Fantastical does, with the difference that it runs on Windows.