
AB-3 User Guide

AddressBook Level 3 (AB3) is a **desktop app for managing contacts, optimized for use via a Line Interface** (CLI) while still having the benefits of a Graphical User Interface (GUI). If you can type fast, AB3 can get your contact management tasks done faster than traditional GUI apps.

AB-3 User Guide

Quick start

Features

Viewing help : help

Adding a person: add

Listing all persons : list

Editing a person : edit

Locating persons by name: find

Deleting a person : delete

Clearing all entries : clear

Exiting the program : exit

Saving the data

Editing the data file

Archiving data files [coming in v2.0]

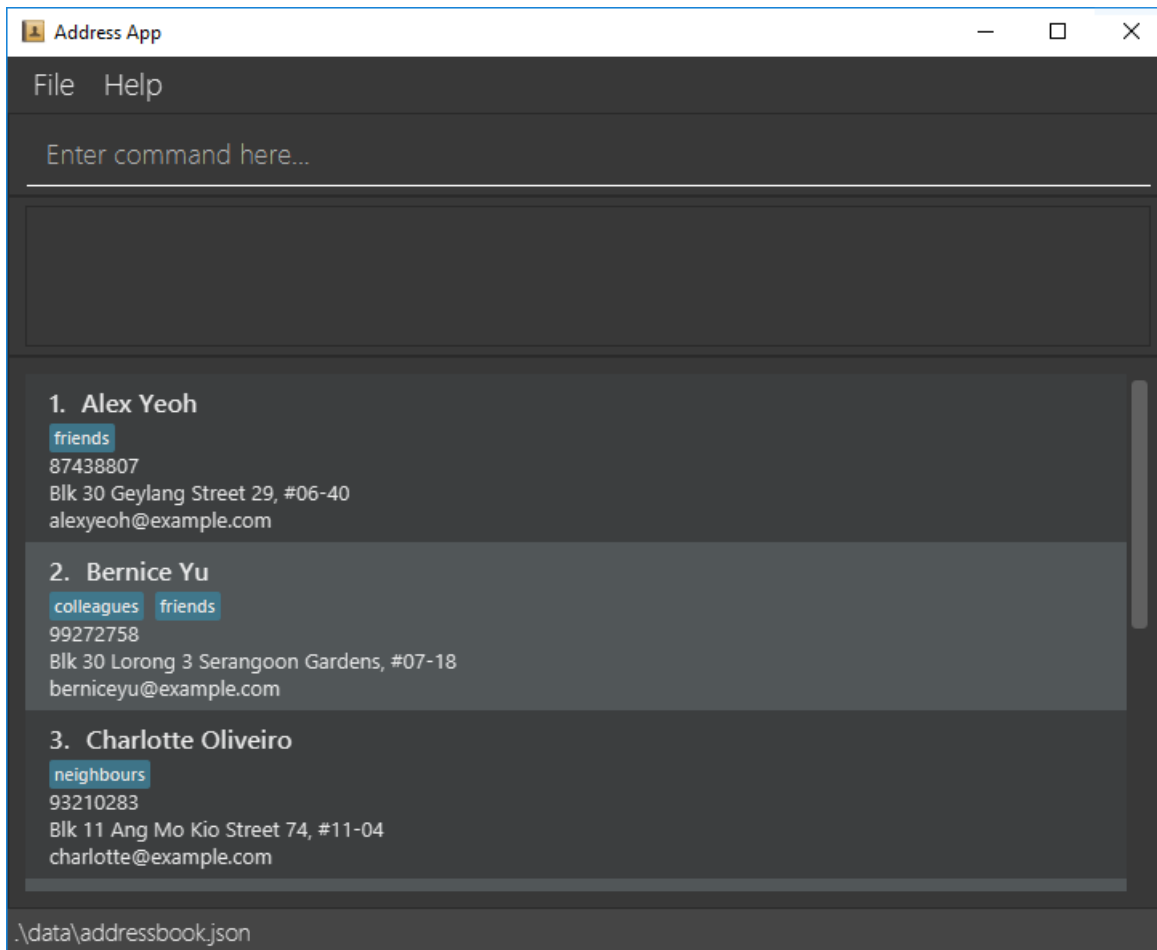
FAQ

Known issues

Command summary

Quick start

1. Ensure you have Java `11` or above installed in your Computer.
2. Download the latest `addressbook.jar` from [here](#).
3. Copy the file to the folder you want to use as the *home folder* for your AddressBook.
4. Open a command terminal, `cd` into the folder you put the jar file in, and use the `java -jar addressbook.jar` command to run the application.
A GUI similar to the below should appear in a few seconds. Note how the app contains some sample data.



5. Type the command in the command box and press Enter to execute it. e.g. typing `help` and pressing Enter will open the help window. Some example commands you can try:

- o `list` : Lists all contacts.
- o `add n/John Doe p/98765432 e/johnd@example.com a/John street, block 123, #01-01` : Adds a contact named `John Doe` to the Address Book.
- o `delete 3` : Deletes the 3rd contact shown in the current list.
- o `clear` : Deletes all contacts.
- o `exit` : Exits the app.

6. Refer to the [Features](#) below for details of each command.

Features

i

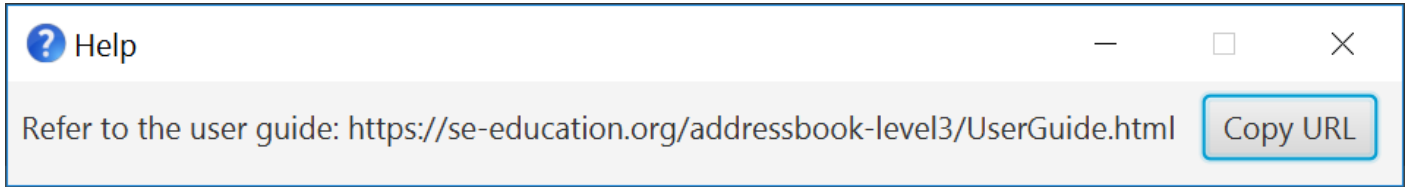
Notes about the command format:

- Words in `UPPER_CASE` are the parameters to be supplied by the user.
e.g. in `add n/NAME`, `NAME` is a parameter which can be used as `add n/John Doe`.
- Items in square brackets are optional.
e.g. `n/NAME [t/TAG]` can be used as `n/John Doe t/friend` or as `n/John Doe`.
- Items with `...` after them can be used multiple times including zero times.
e.g. `[t/TAG]...` can be used as `(i.e. 0 times)`, `t/friend`, `t/friend t/family` etc.
- Parameters can be in any order.
e.g. if the command specifies `n/NAME p/PHONE_NUMBER`, `p/PHONE_NUMBER n/NAME` is also acceptable.
- Extraneous parameters for commands that do not take in parameters (such as `help`, `list`, `exit` and `clear`) will be ignored.
e.g. if the command specifies `help 123`, it will be interpreted as `help`.

- If you are using a PDF version of this document, be careful when copying and pasting commands that span multiple lines as space characters surrounding line-breaks may be omitted when copied over to the application.

Viewing help : `help`

Shows a message explaining how to access the help page.




Format: `help`

Adding a person: `add`

Adds a person to the address book.

Format: `add n/NAME p/PHONE_NUMBER e/EMAIL a/ADDRESS [t/TAG]...`

 **Tip:** A person can have any number of tags (including 0)

Examples:

- `add n/John Doe p/98765432 e/johnd@example.com a/John street, block 123, #01-01`
- `add n/Betsy Crowe t/friend e/betsycrowe@example.com a/Newgate Prison p/1234567 t/criminal`

Listing all persons : `list`

Shows a list of all persons in the address book.

Format: `list`

Editing a person : `edit`

Edits an existing person in the address book.

Format: `edit INDEX [n/NAME] [p/PHONE] [e/EMAIL] [a/ADDRESS] [t/TAG]...`

- Edits the person at the specified `INDEX`. The index refers to the index number shown in the displayed person list. The index **must be a positive integer** 1, 2, 3, ...
- At least one of the optional fields must be provided.
- Existing values will be updated to the input values.
- When editing tags, the existing tags of the person will be removed i.e adding of tags is not cumulative.
- You can remove all the person's tags by typing `t/` without specifying any tags after it.

Examples:

- `edit 1 p/91234567 e/johndoe@example.com` Edits the phone number and email address of the 1st person to be `91234567` and `johndoe@example.com` respectively.
- `edit 2 n/Betsy Crower t/` Edits the name of the 2nd person to be `Betsy Crower` and clears all existing tags.

Locating persons by name: `find`

Finds persons whose names contain any of the given keywords.

Format: `find KEYWORD [MORE_KEYWORDS]`

- The search is case-insensitive. e.g `hans` will match `Hans`

- The order of the keywords does not matter. e.g. `Hans Bo` will match `Bo Hans`
- Only the name is searched.
- Only full words will be matched e.g. `Han` will not match `Hans`
- Persons matching at least one keyword will be returned (i.e. `OR` search). e.g. `Hans Bo` will return `Hans Gruber`, `Bo Yang`

Examples:

- `find John` returns `John` and `John Doe`
- `find alex david` returns `Alex Yeoh`, `David Li`

2 persons listed!

1. Alex Yeoh

`friends`
 87438807
 Blk 30 Geylang Street 29, #06-40
 alexyeoh@example.com

2. David Li

`family`
 91031282
 Blk 436 Serangoon Gardens Street 26, #16-43
 lidavid@example.com

Deleting a person : `delete`

Deletes the specified person from the address book.

Format: `delete INDEX`

- Deletes the person at the specified `INDEX`.
- The index refers to the index number shown in the displayed person list.
- The index **must be a positive integer** 1, 2, 3, ...

Examples:

- `list` followed by `delete 2` deletes the 2nd person in the address book.
- `find Betsy` followed by `delete 1` deletes the 1st person in the results of the `find` command.

Clearing all entries : `clear`

Clears all entries from the address book.

Format: `clear`

Exiting the program : `exit`

Exits the program.

Format: `exit`

Saving the data

AddressBook data are saved in the hard disk automatically after any command that changes the data. There is no need to save manually.

Editing the data file

AddressBook data are saved automatically as a JSON file `[JAR file location]/data/addressbook.json`. Advanced users are welcome to update data directly by editing that data file.

!

Caution: If your changes to the data file makes its format invalid, AddressBook will discard all data and start with an empty data file at the next run. Hence, it is recommended to take a backup of the file before editing it.

Furthermore, certain edits can cause the AddressBook to behave in unexpected ways (e.g., if a value entered is outside the acceptable range). Therefore, edit the data file only if you are confident that you can update it correctly.

Archiving data files `[coming in v2.0]`

Details coming soon ...

FAQ

Q: How do I transfer my data to another Computer?
A: Install the app in the other computer and overwrite the empty data file it creates with the file that contains the data of your previous AddressBook home folder.

Known issues

1. **When using multiple screens**, if you move the application to a secondary screen, and later switch to using only the primary screen, the GUI will open off-screen. The remedy is to delete the `preferences.json` file created by the application before running the application again.

Command summary

Action	Format, Examples
Add	<code>add n/NAME p/PHONE_NUMBER e/EMAIL a/ADDRESS [t/TAG]...</code> e.g., <code>add n/James Ho p/22224444 e/jamesho@example.com a/123, Clementi Rd, 1234665 t/friend t/colleague</code>
Clear	<code>clear</code>
Delete	<code>delete INDEX</code> e.g., <code>delete 3</code>
Edit	<code>edit INDEX [n/NAME] [p/PHONE_NUMBER] [e/EMAIL] [a/ADDRESS] [t/TAG]...</code> e.g., <code>edit 2 n/James Lee e/jameslee@example.com</code>
Find	<code>find KEYWORD [MORE_KEYWORDS]</code> e.g., <code>find James Jake</code>
List	<code>list</code>
Help	<code>help</code>