ProBACS: Profile-Based Automated Compilation System User Manual

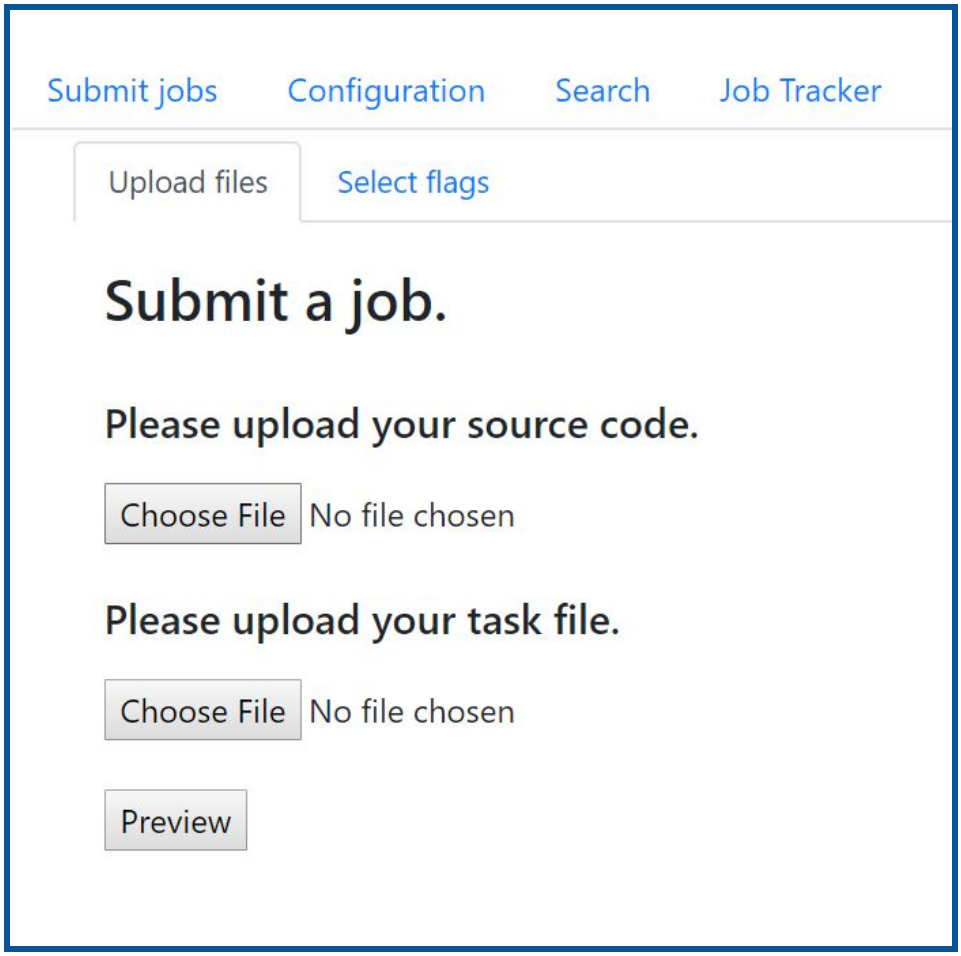
Myra Iltefat, Shipei Zhou, Qiyang He, Hanfu Zhang

CONTENTS

I] Submitting New Jobs

PROBACS allows users to submit compilation jobs using two different methods. If you have a pre-saved text file that you would like to use as a template for creating new jobs, click on the ‘Upload files’ tab. If you would like to select your own combination of OS, compiler, profiles, username and tags for creating new jobs, click on the ‘Select flags’ tab.

Submitting jobs - Using a task file



1) Click on “Choose File” and upload your source code and task file.
2) Click on “Preview”.

3) A simple task file example is like below.

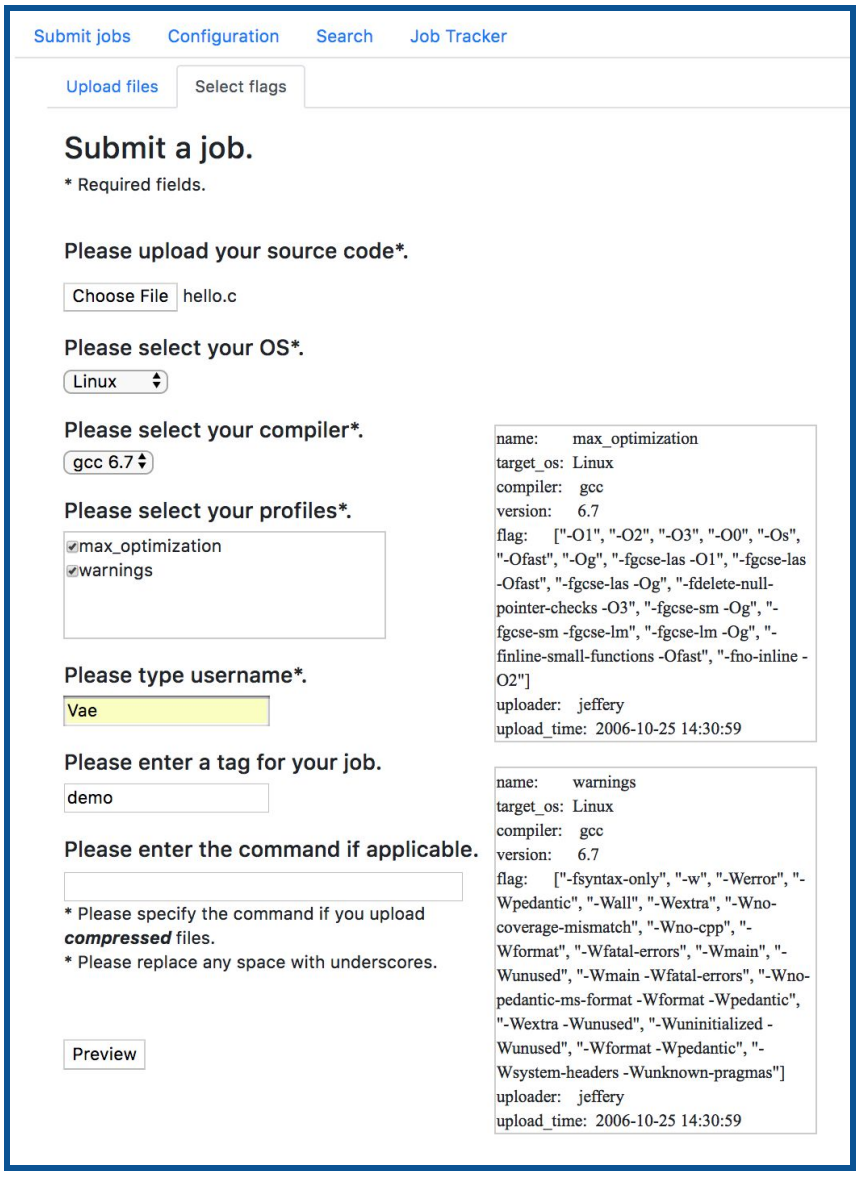
```
-----------------------------------
target_os: Linux, Linux
compiler: gcc, gcc
version: 4.8, 6.7
profile: warnings, max_optimization
username: jeff
tag: test
-----------------------------------
```

4) if you want to compile with a tarball handin, we need you to specify compilation command with file names included. You can leave executable names and flags field because we will replace it with real one when triggering the real compilation command

```
-------------------------------------------
target_os: Linux
compiler: gcc
version: 6.7
profile: warnings, max_optimization
username: jeff
tag: test
command: gcc_flags_main.c_hello.c_-o_exename
-------------------------------------------
```

5) We also support multi-platform compilation. For more details please look into Probacs -> experiment-data, we provide several extra templates for you.

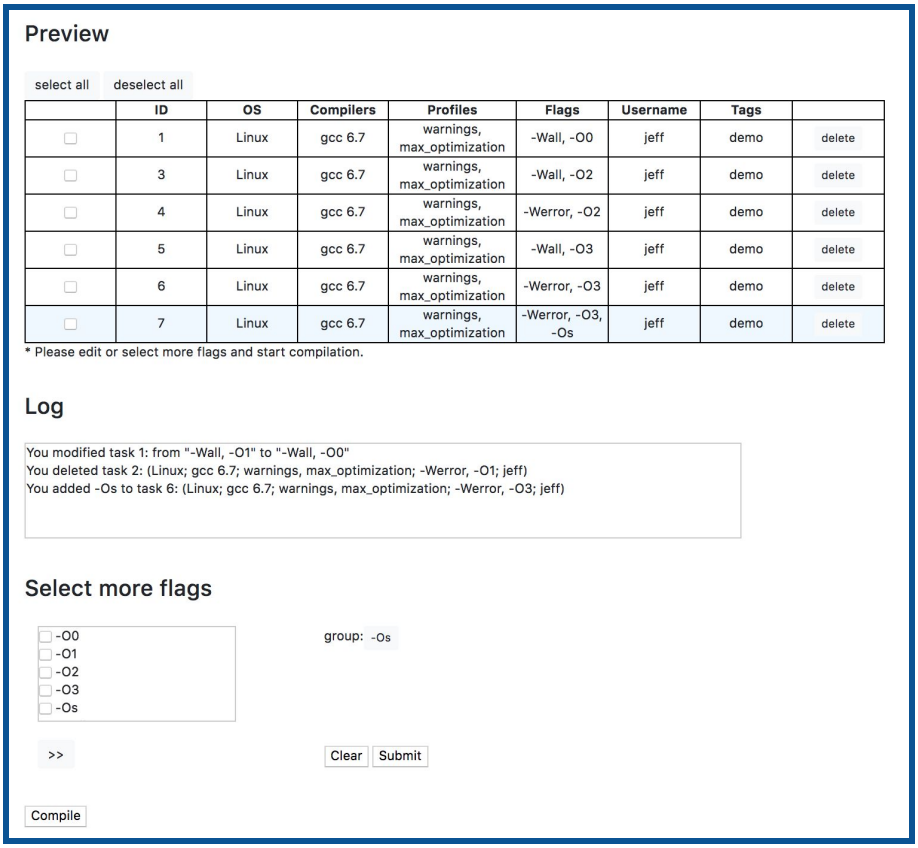
Submitting jobs - Starting from scratch



1) Click on “Choose File” and upload your source code.
2) Choose your desired Operating system (OS), compiler and profile(s). Notice that as you click on various profile names, their details pop-up in text boxes on the right so you can make your selection easily.
3) Type in your username in the textbox.
4) Steps 1,2 and 3 are necessary to successfully submit a job.
5) If you want to remember this task by associating it with a tag, please type it in the given textbox.
6) Click “Preview”.

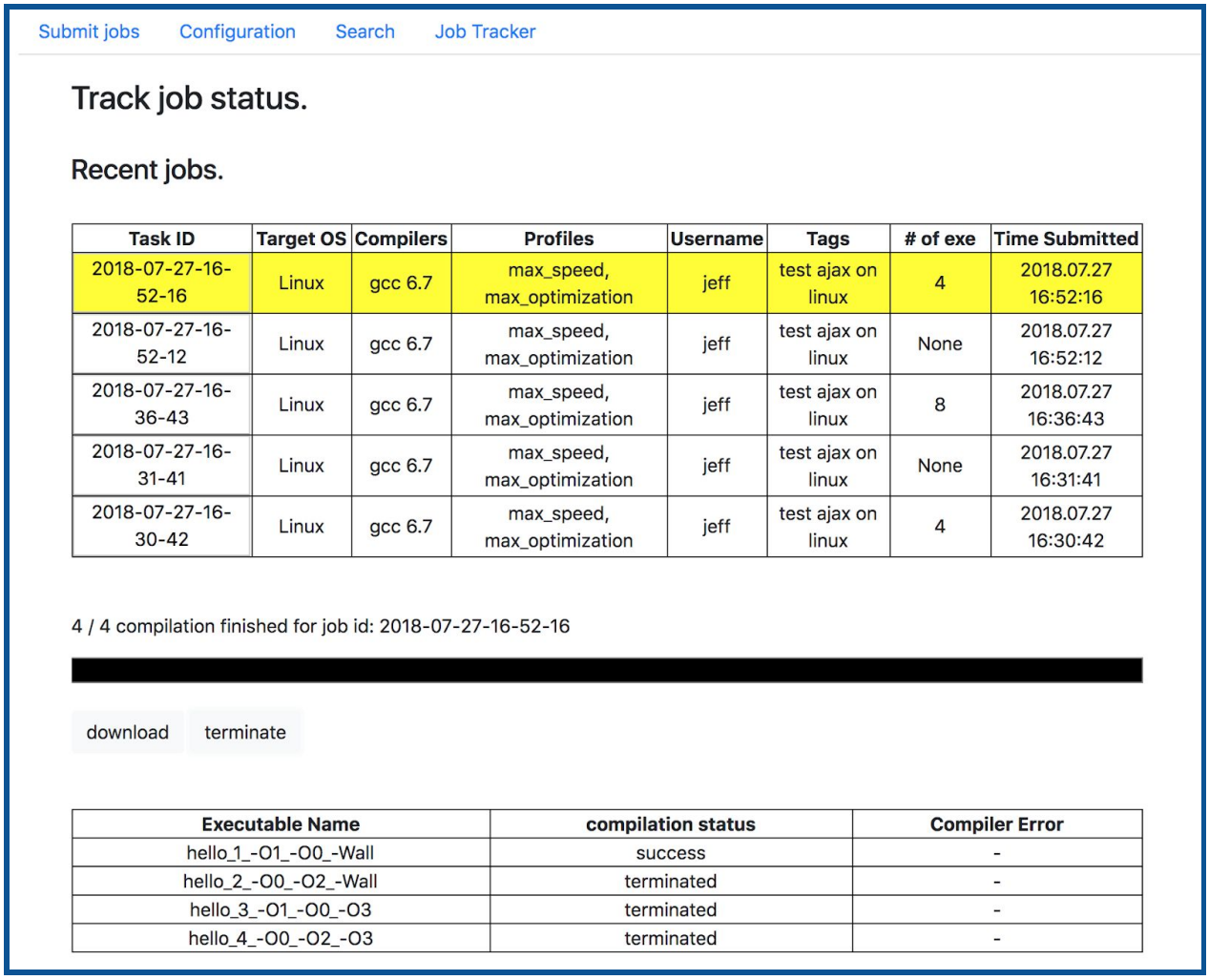
Submitting jobs using multiple source code files

Previewing, fine-tuning and submitting jobs



1) On the preview page, you can view all the compilation tasks lined up for the specified job.
2) You can delete any combination by clicking “Delete” on the right end of that particular row in the given table.
3) You can also modify the flags in any task you want by clicking on that particular cell in the “Flags” column and typing in your modifications.
4) The preview page also allows you to generate new combinations from new flags. Select the flags you want to append to any task given in the table. Once you have made your selection, click on “>>”. After that, select the tasks you want to append the new flag combination to by clicking on the checkboxes in the left of the appropriate rows. Click on “Submit”. The new tasks generated will appear highlighted in blue.
5) The log below the table will record any modifications you make while on the preview page.
6) Once you are satisfied with the job, click “Compile”.

II] Tracking and terminating jobs

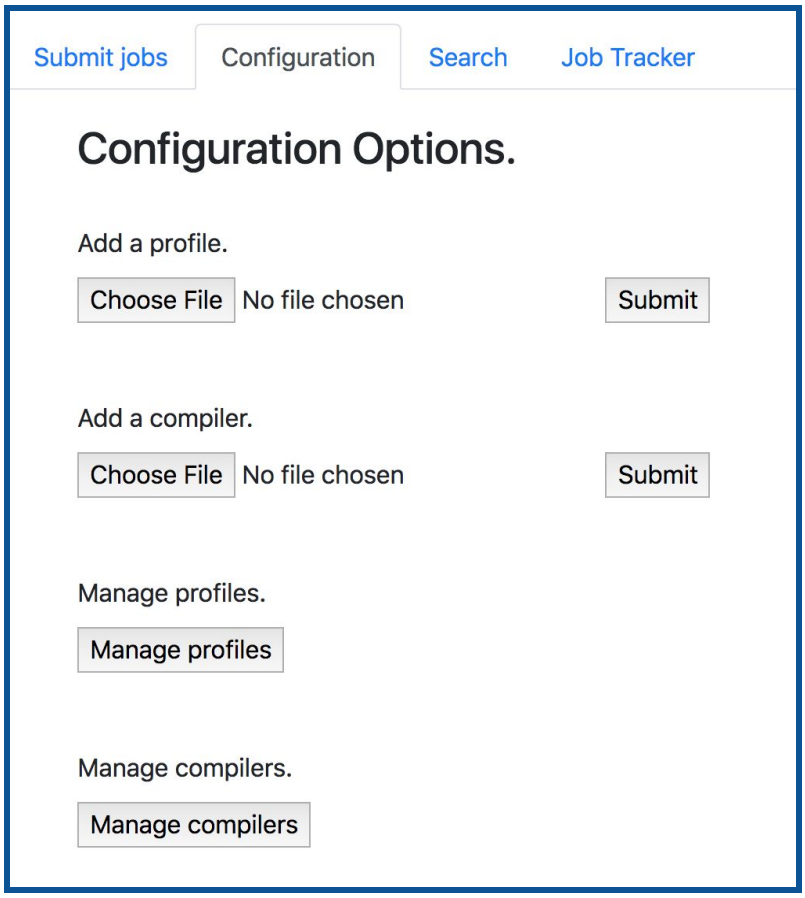


Probacs also allows users to track the status and progress recent jobs. Click on the “Job Tracker” tab.

- The tracking page displays recently compiled and running jobs.
- Selecting the desired job opens up a progress bar and gives a count of how many tasks have been compiled and how many are left. The table below gives detailed status and any associated errors with each task within the selected job.
- The “Compilation Status” field can display one of the following four values:
    - Success : The job has been compiled successfully.
    - Fail : The job could not compile because of some error.
    - Ongoing : The job is compiling or queued.
    - Terminated: The job was terminated by the user.

- The “Compiler Error” field displays errors as received from the compiler.
- In order to terminate a specific job, select the desired job and click “Terminate”. The Progress Bar will immediate fill up completely and the status table will display “Terminated” all the tasks that were ended before they could be compiled.
- Note that the “Download” button will only become available after the job is either complete or is terminated.

III] Configuration Options



Adding a new profile

1) Click on “Choose File” under “Add a profile”, and choose a profile configuration file. The file is a column separated file and an example is below:
   1.1) The `flag` field should be placed at the end of the profile configuration file. Each line after `flag` represents a combination of compiler flag that you wish to execute during the compilation.
   1.2) The profile must have a corresponding compiler (with the same `target_os, compiler, version`) in the compiler database or you will not be able to use this profile.
   1.3) A profile is dedicated to a specific compiler specified by `target_os, compiler, version`. If you wish to reuse a profile across different compilers, simply upload another profile configuration file.

```
-------------------------
target_os: Linux
name: test_profile
compiler: gcc
version: 8.0
uploader: probacs team
flag:
-O1
-O2
-Werror -O1
-Werror -O2
-------------------------
```

2) Click on “Submit”.

Adding a new compiler

1) Click on “Choose File” under “Add a compiler”, and choose a compiler configuration file. The file is a column separated file and an example is below:

```
-----------------------------------------
target_os:Linux
compiler:gcc
version:6.7
ip:127.0.0.1
port:8000
flag: -O1,-O2,-O3,-Wall, -Werror
invoke_format:gcc_flags_source_-o_exename
-----------------------------------------
```

1.1) Make sure you specify the ip as shown above **without** a “http://”

1.2) You can append the flags shown in `flag` field when submitting compilation jobs to this compiler, (on the preview page). You can add multiple flags in this field, using commas to separate them.

1.3) `invoke_format` field specifies how to run a compilation command as if you are compiling an executable directly in the platform server’s terminal (or command prompt). For example, if you can compile an executable in a platform server’s terminal by the following command:

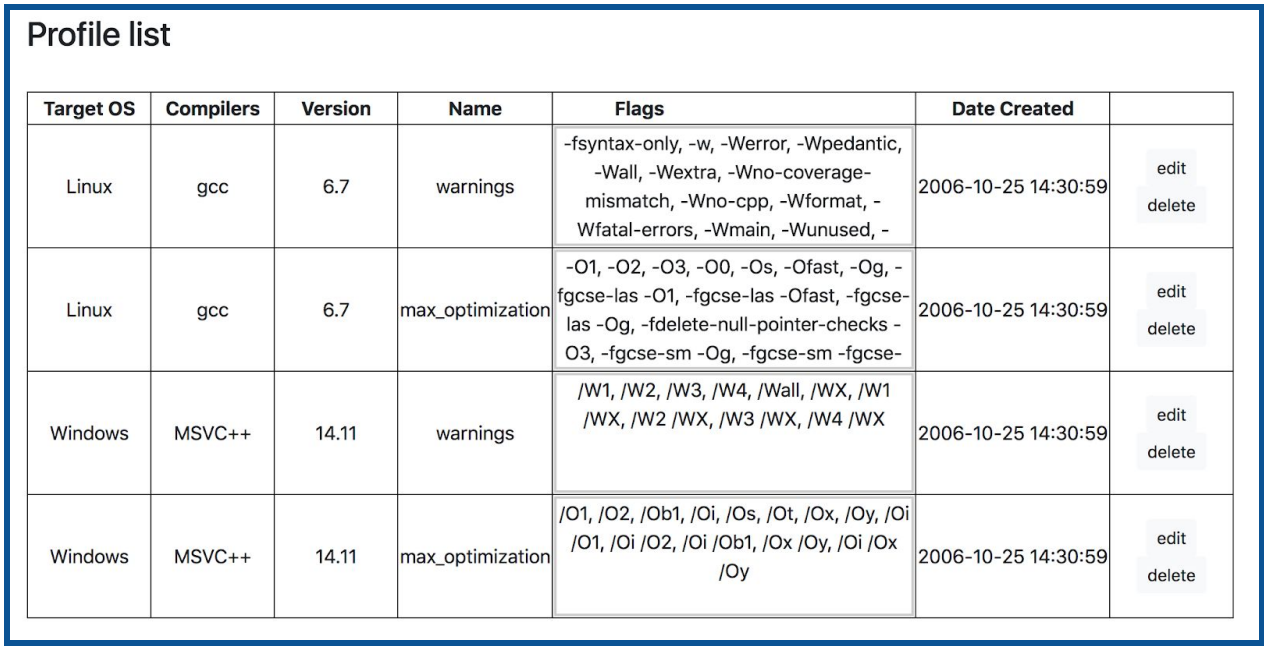
```
gcc -Wall -O3 hello.c -o hello
```

Then you can generate the `invoke_format` by replacing the white spaces in the command by underscore, all the flags by `flags`, the name of source code file by `source`, and the name of the output executable by `exename`. The resulting `invoke_format` will be `gcc_flags_source_-o_exename`

2) Click ‘Submit’.

Editing a profile

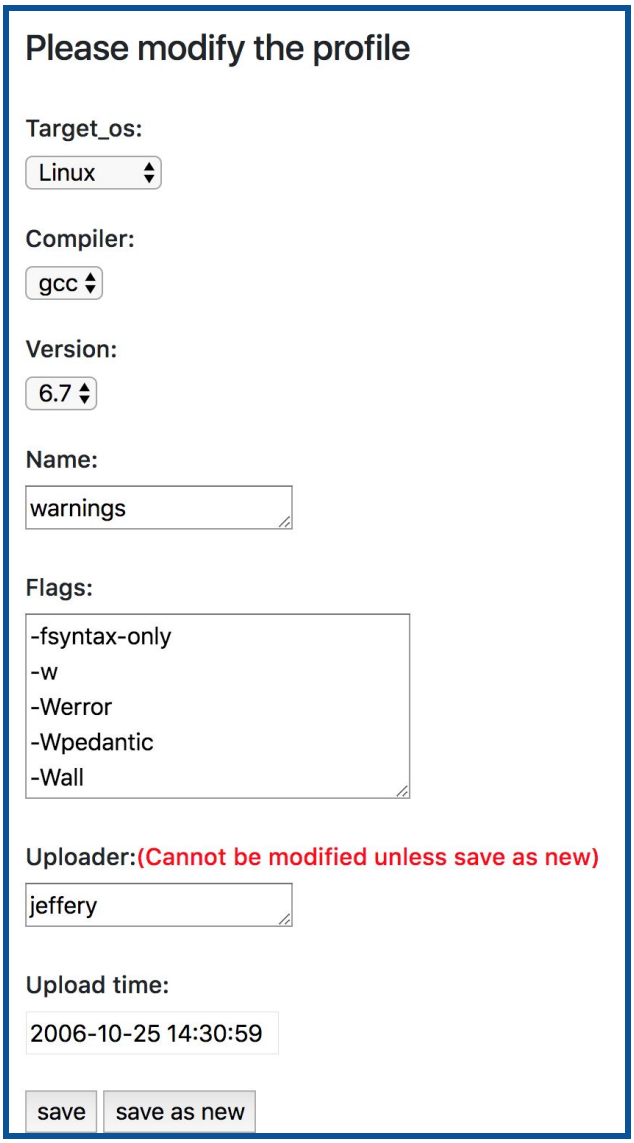
1) Click on “Manage profiles”, and redirect to the profile configuration page.
2) In this page, the profile list is provided from the database. For each profile, several fields are provided, such as `Target OS`, `Compilers`, `Version`, `Name`, `Flags` and `Date Created`.



Profile list

Target OS	Compilers	Version	Name	Flags	Date Created	
Linux	gcc	6.7	warnings	-fsyntax-only, -w, -Werror, -Wpedantic, -Wall, -Wextra, -Wno-coverage-mismatch, -Wno-cpp, -Wformat, -Wfatal-errors, -Wmain, -Wunused, -	2006-10-25 14:30:59	edit delete
Linux	gcc	6.7	max_optimization	-O1, -O2, -O3, -O0, -Os, -Ofast, -Og, -fgcse-las -O1, -fgcse-las -Ofast, -fgcse-las -Og, -fdelete-null-pointer-checks -O3, -fgcse-sm -Og, -fgcse-sm -fgcse-	2006-10-25 14:30:59	edit delete
Windows	MSVC++	14.11	warnings	/W1, /W2, /W3, /W4, /Wall, /WX, /W1 /WX, /W2 /WX, /W3 /WX, /W4 /WX	2006-10-25 14:30:59	edit delete
Windows	MSVC++	14.11	max_optimization	/O1, /O2, /Ob1, /Oi, /Os, /Ot, /Ox, /Oy, /Oi /O1, /Oi /O2, /Oi /Ob1, /Ox /Oy, /Oi /Ox /Oy	2006-10-25 14:30:59	edit delete

3) Users can click “edit” to modify specific profile, then more details will be displayed under the list.
4) In this page, more detailed information such as `Uploader` and `Upload time` are provided. The profile in the database is recognized by the quadruple key (`Target OS`, `Compilers`, `Version`, `Name`). Users can also check the existing `Target OS`, `Compilers`, `Version` in the dropdown lists.
5) Users can modify all information and then click “save” or “save as new”. If users click “save”, all the information will be saved in the original file. If users click “save”, all the information will be saved as a new profile, therefore the quadruple key should be different. In any case, if the quadruple key is repeated with other profiles in the database, the system will ignore it and not save it.
6) After saving the profile, it will redirect to the configuration page with corresponding message.

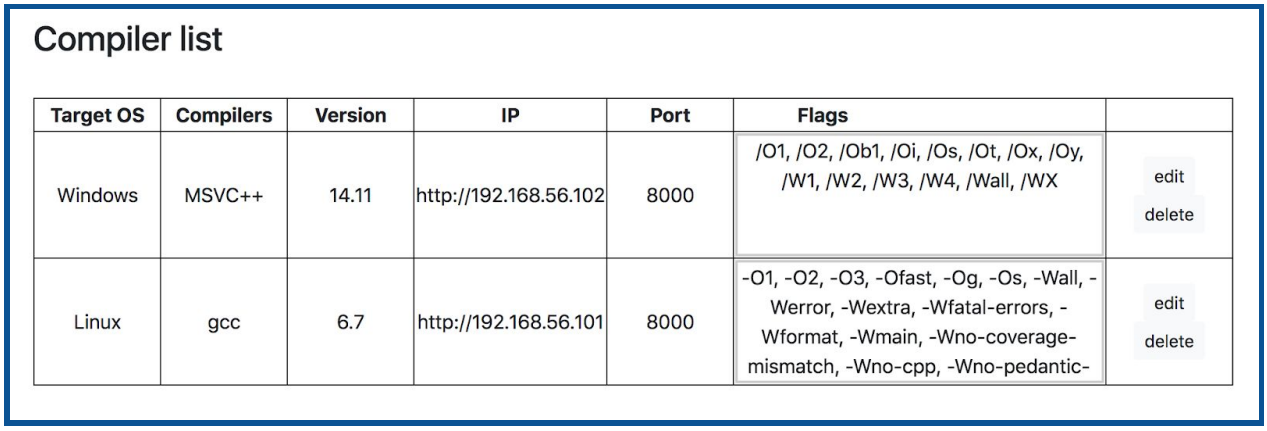


Deleting a profile

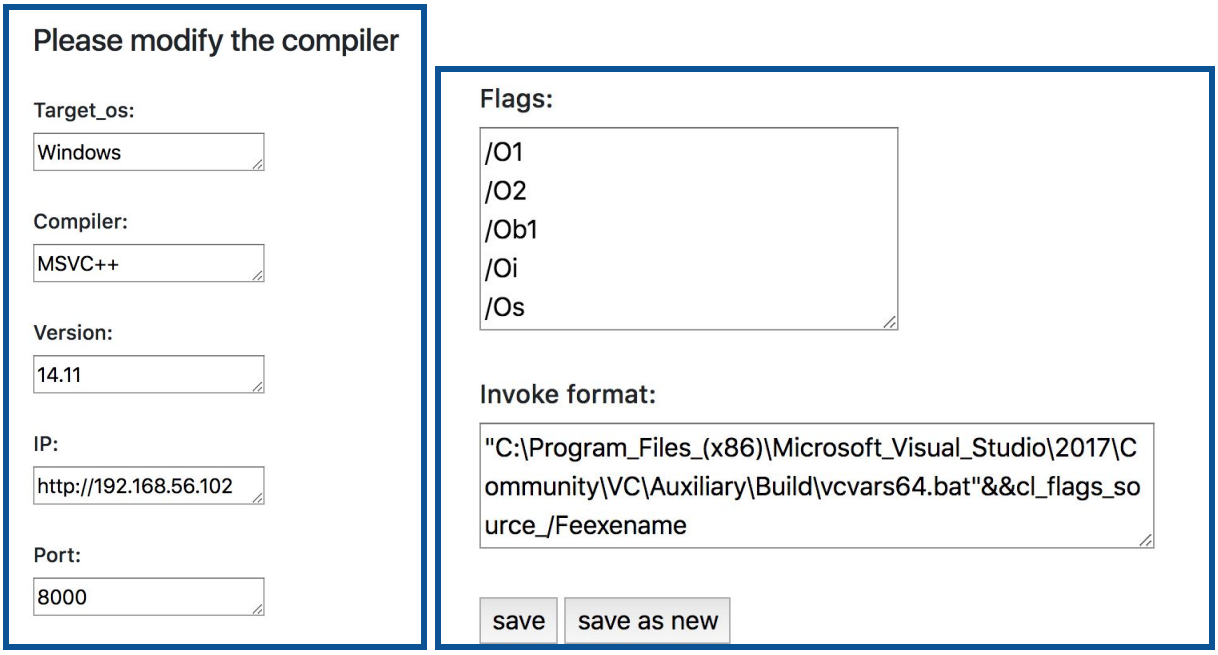
- On the profile configuration page, click ‘delete’ on the right of the profile that you wish to delete, then it will redirect to the configuration page with corresponding message.

Editing a compiler

1) Click on “Manage compilers”, and redirect to the compiler configuration page.
2) In this page, the compiler list is provided from the database. For each compiler, several fields are provided, such as `Target OS`, `Compilers`, `Version`, `IP`, `Port` and `Flags`.



3) Users can click “edit” to modify specific compiler, then more details will be displayed under the list.



4) In this page, more detailed information such as `HTTP path` and `Invoke format` are provided. The compiler in the database is recognized by the triple key (`Target OS`,

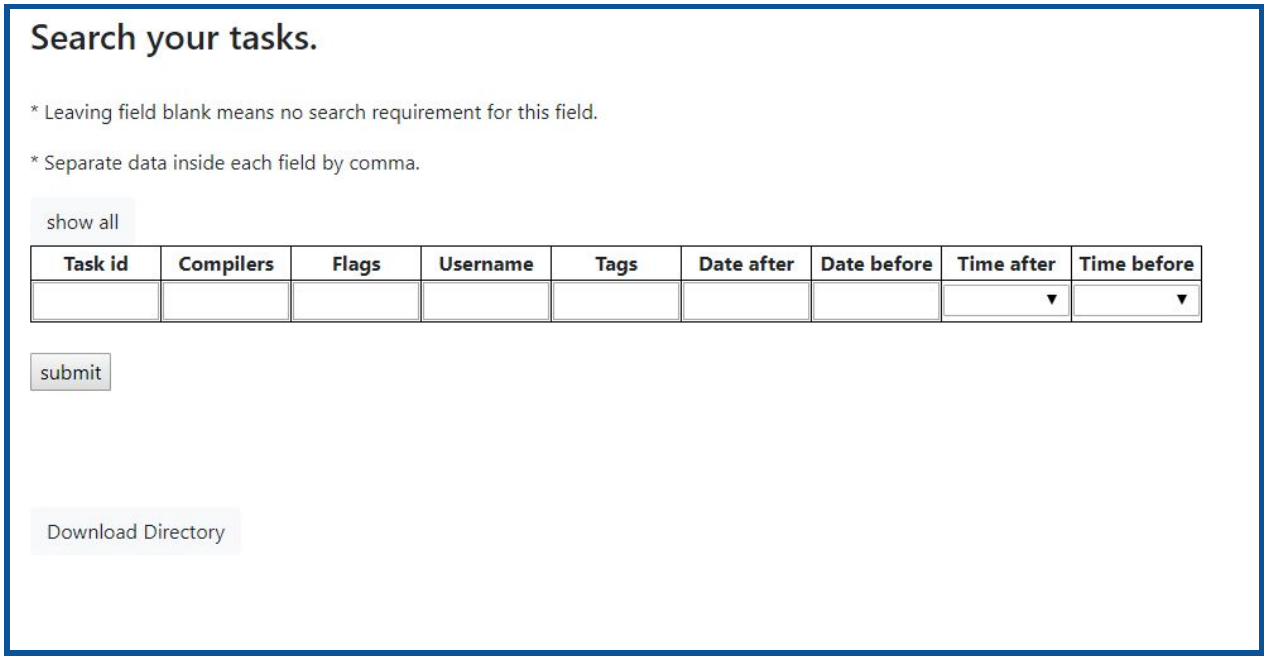
`Compilers`, `Version`). Users can also check the existing `Target OS`, `Compilers`, `Version` in the dropdown lists.

5) Users can modify all information and then click “save” or “save as new”. If users click “save”, all the information will be saved in the original file. If users click “save”, all the information will be saved as a new compiler, therefore the triple key should be different. In any case, if the triple key is repeated with other profiles in the database, the system will ignore it and not save it.
6) After saving the compiler, it will redirect to the configuration page with corresponding message.

Deleting a compiler

- On the compiler configuration page, click ‘delete’ on the right of the compiler that you wish to delete, then it will redirect to the configuration page with corresponding message.

IV] Searching for submitted tasks



The search functionality lets users search for previously compiled tasks. Click on the “Search” tab.

- Tasks can be searched by task ID, compilers, username, tags and the date and time of the task submission.
- Entering multiple search criteria in the same field, e.g entering #helloworld and #test will imply an **OR** between the two keywords. That means it will return tasks having either #helloworld or #test or both tags.
- Entering multiple arguments in different search fields e.g “Date after : 7/1/2018 Username: Jeffery” will imply an **AND** between the keywords. That means it will return tasks compiled by Jeffery after 7/1/2018.
- For searching compilers, we enable users to specify multiple compiler names and versions by using ‘*’, and split different compilers by comma, for example:
    - If you want to search executables compiled with gcc 3.0 and gcc 3.1:
        - gcc 3.0,gcc 3.1
    - If you want to search executables compiled with all versions of gcc:
        - gcc *
    - If you want to search all compilers with version 3.0:
        - * 3.0

V] Using the command line interface

Probacs also provides a command line interface (CLI). The CLI requires Python 3 and allows the users to perform the following features:

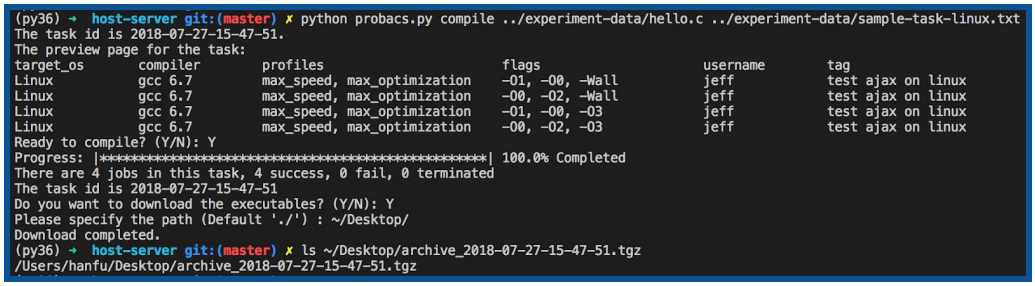
- Submitting new jobs using source code and task files
- Terminating running jobs
- Searching for submitted tasks
- Tracking progress of jobs

User can invoke CLI with command “probacs” by setting up the alias. For how to set an alias, please refer to installation guide section C.

Submitting new jobs using the CLI

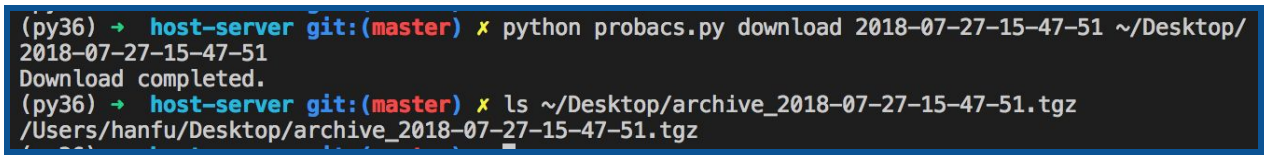
- Use the following syntax to submit a source code and task file for compilation:
    - **“ python probacs.py compile** *sourcefile taskfile* **”**
- Where:
        - *sourcefile* is the path of the source code file
        - *taskfile* is the path of the task file

- After submitting, the system will provide a task id and display the preview information with the task details like in the web interface.
- The system will ask for a confirmation for whether you want to complete the compilation. Type ‘Y’ or ‘y’ to continue. It is case-insensitive.
- The system will begin the compilation and display a live progress bar.
- After the task is completed, it will give a total count of submitted, completed and failed tasks.
- The system will ask for a confirmation for whether you want to download the executables. Type ‘Y’ or ‘y’. Specify the path of the download folder.
- You can use the command Ctrl-c to terminate the running jobs during compilation, then the system will stop the jobs and print corresponding message.



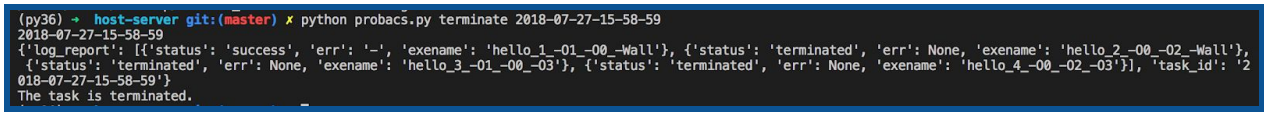
Downloading jobs using the CLI

- Use the following syntax to download a specific file from the server, given the task id.
  “ python probacs.py download *task_id destination_folder* **”**
- Where:
  - *task_id* is the id of the task you want to download
  - *Destination_folder* is the path of the folder where you want to download



Terminating running jobs using the CLI

- Use the following syntax to terminate a running job:
  “ python probacs.py terminate *task_id***”**
- Where:
  - *task_id* is the id of the task you want to terminate

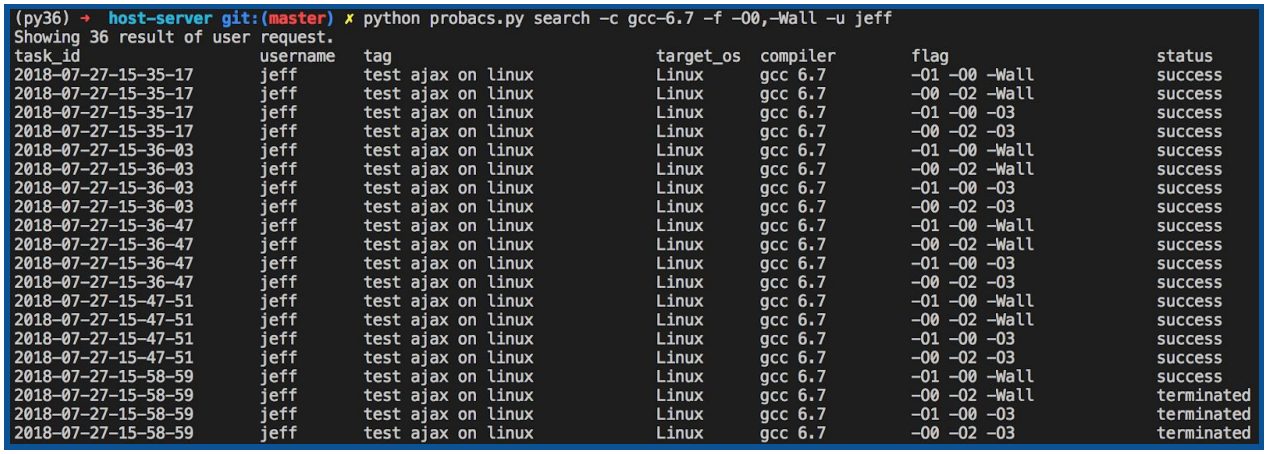


Searching for submitted tasks using the CLI

- Use the following syntax to search for tasks using keywords:
  “python probacs.py search *keywords* **”**
- You can specify the keywords in any order. However, the name of the search field needs to be specified in the form of flags. Use the table below for reference.

Search field	Flag to use
Task id	-tid
Compiler	-c
Flags	-f
Username	-u
Tag	-t
Search all	-all

- Note that if there are multiple keywords within one search field, they need to be separated by commas.



```
(py36) ➜  host-server git:(master) ✗ python probacs.py search -c gcc-6.7 -f -O0,-Wall -u jeff
Showing 36 result of user request.
task_id              username   tag                        target_os  compiler       flag                 status
2018-07-27-15-35-17  jeff       test ajax on linux         Linux      gcc 6.7        -O1 -O0 -Wall        success
2018-07-27-15-35-17  jeff       test ajax on linux         Linux      gcc 6.7        -O0 -O2 -Wall        success
2018-07-27-15-35-17  jeff       test ajax on linux         Linux      gcc 6.7        -O1 -O0 -O3          success
2018-07-27-15-35-17  jeff       test ajax on linux         Linux      gcc 6.7        -O0 -O2 -O3          success
2018-07-27-15-36-03  jeff       test ajax on linux         Linux      gcc 6.7        -O1 -O0 -Wall        success
2018-07-27-15-36-03  jeff       test ajax on linux         Linux      gcc 6.7        -O0 -O2 -Wall        success
2018-07-27-15-36-03  jeff       test ajax on linux         Linux      gcc 6.7        -O1 -O0 -O3          success
2018-07-27-15-36-03  jeff       test ajax on linux         Linux      gcc 6.7        -O0 -O2 -O3          success
2018-07-27-15-36-47  jeff       test ajax on linux         Linux      gcc 6.7        -O1 -O0 -Wall        success
2018-07-27-15-36-47  jeff       test ajax on linux         Linux      gcc 6.7        -O0 -O2 -Wall        success
2018-07-27-15-36-47  jeff       test ajax on linux         Linux      gcc 6.7        -O1 -O0 -O3          success
2018-07-27-15-36-47  jeff       test ajax on linux         Linux      gcc 6.7        -O0 -O2 -O3          success
2018-07-27-15-47-51  jeff       test ajax on linux         Linux      gcc 6.7        -O1 -O0 -Wall        success
2018-07-27-15-47-51  jeff       test ajax on linux         Linux      gcc 6.7        -O0 -O2 -Wall        success
2018-07-27-15-47-51  jeff       test ajax on linux         Linux      gcc 6.7        -O1 -O0 -O3          success
2018-07-27-15-47-51  jeff       test ajax on linux         Linux      gcc 6.7        -O0 -O2 -O3          success
2018-07-27-15-58-59  jeff       test ajax on linux         Linux      gcc 6.7        -O1 -O0 -Wall        success
2018-07-27-15-58-59  jeff       test ajax on linux         Linux      gcc 6.7        -O0 -O2 -Wall        terminated
2018-07-27-15-58-59  jeff       test ajax on linux         Linux      gcc 6.7        -O1 -O0 -O3          terminated
2018-07-27-15-58-59  jeff       test ajax on linux         Linux      gcc 6.7        -O0 -O2 -O3          terminated
```