The *marm_2* and *edX* Scripts User Manual

Updated version of *marm_sql* script and new grades related *edX* script for CS136

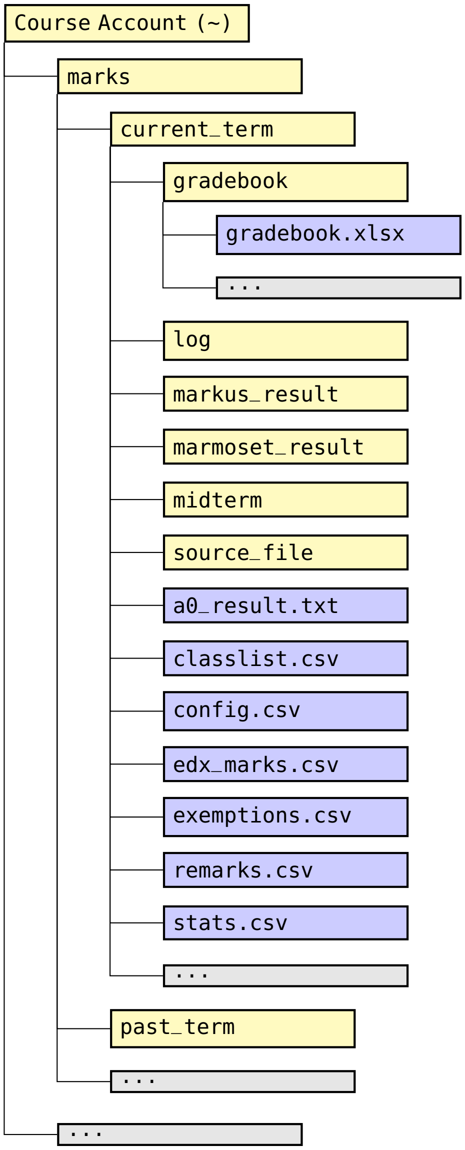
Name: Le Zhang

Update Date: Apr 1st, 2024

Contents

Chapter 1

File Structures



The new design of `marm2` and `edx` used file structure right by default. By running scripts setup in course account (course server environment). The left file structure will create in the course account. The `edx` could initial `current_term` folder like shown left. Many default path in subsequent scripts in this manual can be found based on the file structure on the left.

`current_term` is a linked symbol to the actual data folder in `past_term`. It will always linked with current term data folder in `past_term` by `edx -i`.

We could save data by linked symbol or actual data path. For convenient, we will use `current_term` bellow to introduce default path.

Important

The data is actually store in

`~/marks/past_terms/folder`

where folder name start with term code. e.g. WInter 2024

`~/marks/past_terms/1241_w24`

1.1. Default Marmoset Result Path

`~/marks/current_terms/marmoset_result`

1.2. Default Markus Result Path

`~/marks/current_terms/markus_result`

1.3. Default Source Code Path

`~/marks/current_terms/source_file`

1.4. Default Class List Path

`~/marks/current_terms/classlist.csv`

Key Insight

```
~/marks/current_terms/a0_result.txt
~/marks/current_terms/config.csv
~/marks/current_terms/edx_marks.csv
~/marks/current_terms/exemptions.csv
~/marks/current_terms/remarks.csv
~/marks/current_terms/stats.csv
```

Introduction to marm2

This manual provides detailed information on the usage of the `marm2` command line tool, which is designed for managing and downloading marks and submissions for assignments and projects. This tool supports various commands for specific operations, such as viewing usage information, downloading marks or submissions, specifying target files or directories, and more.

2.1. Usage

The basic usage of the tool can be invoked with the following options:

2.1.1. Help information

Flags: `-h`

```
marm2 -h
```

It will displays the help message with information on all available commands and their usage.

2.1.2. Marks Download

Flags: `-m`

```
marm2 -m proj
```

Downloads on-time marks for a given assignment or project. By following argument "proj".

The project identifier "proj" is case-insensitive and supports various formats, including:

Key Insight

- An integer `x`. This downloads marks for every project from Assignment `Ax`.
- A string of the form `Ax`, where `x` is an integer. This downloads marks for every project from Assignment `Ax`.
- The partial name of the project where a string of the form `AxQy`, `x` is integer and `y` is integer sometimes come with letter. e.g. `a1q1`, `a1q1a`. This downloads marks for projects which name start with `AxQy`.
- The full name o the project where string of the form `AxQy-NAME`, where`x` is integer and `y` is integer sometimes come with letter. e.g. `a7q3-stringq`, `a8q2a-sequence`. This usually downloads marks for specific project which name is `AxQy-NAME`.

This could also work with lab course where means `AxQy` could be `LABxQy`. All others are working same as cases shown above. The `Q` could be replaced with `P`.

Examples:

```
marm2 -m 3
marm2 -m a3
marm2 -m a7q3-stringq
```

2.1.3. Downloads Submissions Source Code

Flags: `-d`

```
marm2 -d proj
```

It downloads the best on-time submissions for a project, storing them in a directory named after the project. Where "proj" has same format with flag `-m` mentioned.

Example:

```
marm2 -d 3
marm2 -d a3
marm2 -d a7q3-stringq
```

2.1.4. Set Classlist File

Flags: `-s`

```
marm2 -s path
```

If you use this flag follow with the path of a class list, a file include students information. It will specify that file containing a list of student IDs for processing with the `-m` or `-d` options. Otherwise it will use default settings.

Example:

```
marm2 -m 3 -s /some/path
marm2 -d a3 -s /some/path
marm2 -d a7q3-stringq -s /some/path
```

2.1.5. Set Destination Path

Flags: `-t`

```
marm2 -t path
```

If you use this flag follow with the path that you want to store the result. It will specify the target directory for storing results from the `-m` or `-d` options. Otherwise it will use default settings. The default directories are provided for both marks and source file downloads.

Example:

```
marm2 -m 3 -t /some/path
marm2 -d a3 -t /some/path
marm2 -d a7q3-stringq -t /some/path
```

2.1.6. Check The Project Tests Numbers

Flags: `-o`

```
marm2 -o proj
```

A quick way to get the full marks for a project, all projects, or the current project. Where "proj" has same format with flag `-m` mentioned. However if you just use `marm2 -o`. This will show latest assignment tests setup. If you use `marm2 -o a`. This will show all assignments tests setup.

2.1.7. Check Course Primary Key (PK)

Flags: `-c`

```
marm2 -c
```

Retrieves the current course's PK (unique identifier).

2.1.8. Verbose Mode

Flags: `-v`

```
marm2 -v
```

Enables verbose mode, providing extra information about the operations being performed, including a download progress indicator when used with `-d` or `-m`.

The comparison output with/without `-v` flag

```
cs136@ubuntu2204-010 $ marm2 -m 0
[Downloading A0] to: /u/cs136/marks/past_terms/1241_w24/marmoset_result
>> a0q1-piazza
>> a0q2-credentials
>> a0q3-marmoset
>> a0q4-testing
>> a0q5-markus
>> a0q6-integrity
>> a0q7-msteams
```

If we add flags `-v` will enable the verbose mode. This will show the progress when it downloading

```
cs136@ubuntu2204-010 $ marm2 -v -m 0
[Downloading A0] to: /u/cs136/marks/past_terms/1241_w24/marmoset_result
>> 749/749: a0q1-piazza
>> 749/749: a0q2-credentials
>> 749/749: a0q3-marmoset
>> 749/749: a0q4-testing
>> 749/749: a0q5-markus
>> 749/749: a0q6-integrity
>> 749/749: a0q7-msteams
```

2.2. Usage Examples

```
marm2 -m 0                          // download marks from all A0
marm2 -m 0 -s ~/classlist           // download marks from all A0 by ~/classlist
marm2 -m 0 -t ~/result              // download marks from all A0 to ~/result
marm2 -m 0 -s ~/classlist -t ~/result  // download marks from all A0 by ~/classlist to ~/result

marm2 -d 0                          // download source code from all A0
marm2 -d 0 -s ~/classlist           // download source code from all A0 by ~/classlist
marm2 -d 0 -t ~/result              // download source code from all A0 to ~/result
marm2 -d 0 -s ~/classlist -t ~/result  // download source code from all A0 by ~/classlist to ~/result
```

2.3. Usage Command Line Interface

```
Usage:
-h: Display this usage message.

-m proj or -m assn:
    Download on-time marks for a given assignment or project.
    "proj" is case-insensitive and may have any of the following forms:
    ∗ A string of the form AxPy, where x and y are integers.
    ∗ A string of the form AxBonus, where x is an integer.
    ∗ A string of the form Ax, where x is an integer. This will download
      marks for every project from Assignment Ax.
    ∗ An integer x. This will download marks for every project from
      Assignment Ax.
    If "proj" is the name of a single project, a .csv file called
    project-[proj]-grades.csv is created in the current directory containing
    the marks for that project (for example, project-A7P7-grades.csv). If
    "proj" is the name of an assignment, .csv files will be created for each
    project in the assignment.

    Examples:
    marm2 -m A7P4
    Downloads marks for Assignment 7, Problem 4 and saves them in a file
    called project-A7P4-grades.csv.
    marm2 -m a11bonus
    Downloads marks for Assignment 11, Bonus Problem and saves them in a file
    called project-a11bonus-grades.csv.
    marm2 -m A6
    Downloads marks for all Assignment 6 problems. For each problem, a .csv
    file containing the marks is created in the current directory.
    marm2 -m 3
    Downloads marks for all Assignment 3 problems. For each problem, a .csv
    file containing the marks is created in the current directory.

-d proj:
    Download the best on-time submissions for a project. The submissions
    are stored in a folder with the same name as the project in the current
    directory. Unlike the -m option, "proj" can only be a string representing
    a single project rather than a whole assignment.

-s file:
    Specify a file containing a list of student IDs to process with -m or -d.
    By default, these options run for every student in the classlist.

-t directory:
    Specify a directory where you want to store result to used with -m or -d.
    By default, these options run for
    -m (marks download): /u/cs136/marks/past_terms/1241_w24/marmoset_result
    -d (source files download): /u/cs136/marks/past_terms/1241_w24/source_file

-o proj OR a (all) OR c (current):
    Quick way to get the project full marks.

-c: Quick way to get the current course PK (unique number assigned to each
    offering of each course by Marmoset).

-v: Enables verbose mode. The script will print extra information about
    what it is doing. When used in conjunction with -d or -m, a download
    progress indicator is displayed.
```

Introduction to edx

This manual outlines the functionalities of the edx command line tool, specifically designed for managing and organizing marks within the context of academic courses. This utility offers a range of commands to streamline the process of editing course configurations, generating reports, and initializing repositories for the current term among others.

> **Important**
>
> If there is hand marking associated with the assignment, we need to downloads result and upload to server (course account) manually. Upload the markus result to `~/current_term/markus_result/`

3.1. Usage

To utilize the edx tool, you can employ the following options to execute distinct operations:

3.1.1. Initialize Term Repository

Flags: `-i`

```
edx -i
```

This command initializes the repository for the current term, setting up the necessary structure and files in the current term folder. If there is current term folder linked with, this process will unlink the old one. Then make a symbol link to the actual data folder in `~/marks/past_term`.

3.1.2. Modify Assignment Configuration

Flags: `-c`

```
edx -c
```

Utilizes vim to edit the assignment configuration file, allowing for adjustments to the assignment parameters directly from the command line.

Structure of config.csv

> **Key Insight**
>
> 1. `project` is the name of the project. This do not need full name only the code is good.
> 2. `fullMarks` is the total points the question have in the marmoset test setup this could be get by flag `-o`.
> 3. `weight` is the weight of this question.
> 4. `isHandMarking` is the marking type of this project. `1` means this is a memory snapshot question. `2` means this question is counted with stylemark. Otherwisw means this question do not have hand marking.
> 5. `styleWeight` is the weight of this hand marking question. Only fill this when the `isHandMarking` is `1` or `2`.

```
project,fullMarks,weight,isHandMarking,styleWeight
# Assignment 6
a6q1,13,25,2,10
a6q2a,11,10,2,15
a6q2b,8,30,2,60
a6q3,6,35,2,15
```

3.1.3. Modify Exemptions File

Flags: `-e`

```
edx -e
```

Opens the exemptions file in vim, enabling the user to update or modify exemptions for assignments or tests as necessary.

Structure of exemptions.csv

Key Insight

1. `student` is the name of the student.
2. `assessment` is the assessment the student requested.

```
student,assessment
login_id1,A2
login_id2,A2
login_id3,A6
```

3.1.4. Modify Remarks File

Flags: `-r`

```
edx -r
```

Opens the remarks file in vim, facilitating the entry or modification of remarks pertaining to assignments or students.

Structure of remarks.csv

Key Insight

1. `studentID` is the login id of the student.
2. `question` is the question/project the student regrade.
3. `newTotal` is the new grades of marmoset/midterm.
4. `markusMakrs` is the anew grades of markus. This could be None or `X`.

```
studentID,question,newTotal,markusMakrs
login_id1,a4q2,3,X
login_id2,a4q2,3,X
login_id3,a5q2b,9,X
```

3.1.5. Update Grades Result of Marmoset

Flags: `-u`

```
edx -u n
```

Updates the grades report with valid grades from Marmoset, post-deadline, ensuring that the latest assessment results are accurately reflected. The argument `n` could be an integer or flags `-a` where will update all projects by downloading from marmoset database. If there is no arguments `n`, this will download assignment result and download A0 as well.

3.1.6. Generate edx.csv File

Flags: `-g`

```
edx -g
```

Generates an `edx_marks.csv` file within the current term folder, which contains acceptable grades data in comma-separated values format for edX website.

3.1.7. Generate Stats File

Flags: `-s`

```
edx -s
```

Creates a `stats.txt` file within the current term folder, aggregating statistical data regarding course assessments and outcomes.

3.1.8. Check Total Number of Tests in Project

Flags: `-o`

```
edx -o n
```

Displays the total number of tests in a specified project, aiding in the evaluation of the project's testing coverage. The argument `n` could be an integer or flags `-a` where will show all projects tests setup. If there is no arguments `n`, this will show marmoset tests setup of latest assignments.

Example of Output

Key Insight

1. `project` is the name of the project. This do not need full name only the code is good.
2. `fullMarks` is the total points the question have in the marmoset test setup.

```
project,fullMarks
# Assignment 6
a6q1,13
a6q2a,11
a6q2b,8
a6q3,6
```

3.2. Usage Examples

```
edx -i    // intialize data folder
edx -u    // update marks for A0 and latest Assignment
edx -u 1  // update marks for A1
edx -g    // generate edx_marks.csv
```

3.3. Usage Command Line Interface

```
Usage: edx [-i]
Options:
  -c      Use vim to modify assignment config
  -e      Use vim to modify exemptions file
  -g      Generate edx_marks.csv file in current term folder
  -i      Initialize term repo current term folder
  -o      Check the total number of tests in perojet
  -r      Use vim to modify remark file
  -s      Generate stats.txt in current term folder
  -u      Update all valid grades report from marmoset (after deadline)
```