

# Quick Start Manual for Mechanical TA

Chris Thornton  
cwthornt@cs.ubc.ca

August 18, 2013

## Contents

<b>1 Quick Install</b>	<b>1</b>
<b>2 Creating Courses</b>	<b>2</b>
<b>3 User Management</b>	<b>2</b>
<b>4 Assignment Management</b>	<b>3</b>
4.1 Peer Review Assignment Life-Cycle . . . . .	3
<b>5 Other Useful Tips</b>	<b>4</b>

This is the super brief manual for Mechanical TA. I assume that you're familiar with setting up databases/websites, so if anything is unclear send me an e-mail and I'll try to help you out. Parts of MTA are fairly well documents in the code, but other sections were completed just a few hours before students needed to start using it, so some things may be a bit tricky to decipher. If anything is unclear send me an e-mail and I'll try and help you out.

## 1 Quick Install

Mechanical TA requires PHP > 5.3 (with PDO), and MySQL. To set up the database, you first need to load the schema provided in `mta.sql`, and give a user access to the newly created database. On the website side of things, you need to modify a few files (all paths relative from the top level MTA directory):

- **config.php** This file contains the fundamental config options for MTA, including the database connection settings (as well as the LDAP authentication config for the time being). Make sure to update `$SITEURL` to contain the web accessible path to your install, as well as the location of the database (see the PDO documentation for setting the dsn)
- **.htaccess** This file contains the rewrite rules that will give your courses pretty names as opposed to just their IDs in the GET string. The only line that needs to be changed here is the `RewriteBase` to point to the install of MTA on the filesystem.

- **admin/.htaccess** and **admin/.htpasswd** Some functions can be done with a special admin account, these files are located under the admin folder. Rather than relying on some other authentication method, MTA just exploits the webserver to provide access to these files. To use the provided **.htpasswd** file (with username **root** password **mta**, you'll have to update **.htaccess** to point to the full path of the **.htaccess** file. This folder should probably be removed from the MTA install once you've created all your courses, as once you have created your instructor logins there is no use to it any more.

## 2 Creating Courses

Once your install is working (by navigating to your MTA install in a web browser, you should see just a blank screen with the words Mechanical TA up at the top. To make this more exciting, you need to create a course. Navigate to the **admin/coursemanager.php** page. Here, you can either edit any of the created courses or make a new one. When you're editing a course, there are the following fields to enter

**Short Name** The name of the course that will be used to make the URLs more friendly. No spaces or crazy characters would be a good plan.

**Display Name** The full name of the course, no restrictions on what characters are allowed.

**Authentication Type** There are a few options, PDO, LDAP and LDAP with PDO fallback. The PDO option will create usernames/passwords that are stored in the database, while using LDAP will use the specified LDAP server in the **config.php** for authenticatino. The LDAP with PDO fallback will first try to authenticate users using the LDAP method, and if that fails, will resort to using a PDO user. The fallback method is not supported using the UI, you'll have to manually insert users into the database, so only use this one if you know what you're doing.

**Registration Type** An **open** registration allows students to enrol in the course by logging into MTA, while the **closed** only allows users to be added using the usermanager script (see Section 3).

## 3 User Management

On a newly created course, you'll first have to add a user as an instructor before you can do anything useful. Using the **admin/usermanager.php**, select the course that you want to manage, then click the 'New User' link. (On courses that have registered users, you will see them show up here). A page shows up with the following fields:

**Username** The login name for the user

**Password** If the authentication system supports setting a password, you can enter the user's password here, otherwise this field is omitted. If you leave the password blank, no changes to the password database are done.

**First/Last Name** The full names of the user

**Student ID** The user's student ID - useful for performing joins when you're exporting the grades

**Type** The role given to this user. **Instructors** have the ability to create/edit assignments users, while **markers** can only perform marking operations on assignments. Make sure that you set up at least one instructor before you remove the **admin** folder.

Note, once a user is created, they are given an internal ID in the MTA database. This means that you should be very careful when editing the username of a user, since you may have them being credited with all the work of another student.

You can also upload a CSV with the following columns to load in a number of users (there is no header row): Last Name, First Name, Student ID Number, Account Name, User Type (one of student ,instructor or marker).

## 4 Assignment Management

Once you've created your instructor user, it's time to log into your course. Navigate back to the main directory of your MTA install, click your course, and log in. You should be greeted with three buttons on the top left. The first one allows you to create assignments, the second lets you run scripts on all assignments, and the final button lets you manage users.

When you create an assignment, you're given an option of what kind of assignment to make. Most often, you're going to want to create a peer review assignment. Here, you can set the assignment's name and other common properties that are self evident from their label. Once you've saved it, you'll be taken back to the home screen where you can see your newly created assignment. The buttons in the first row of icons let you:

- Move the assignment up in the list
- Move the assignment down in the list
- Delete the assignment
- Edit the assignment settings
- Run scripts on the assignment
- Duplicate the assignment (just the settings, not the student submissions)

The second row of icons is specific to a peer review assignment, in this case they let you:

- Specify the peer review questions
- Specify who is allowed to complete the assignment/ which users are independent/trusted and which are not
- Specify the reviews that are assigned to each student.

## 4.1 Peer Review Assignment Life-Cycle

The peer review process consists of 4 stages. In the first stage, students have to submit what the work that they are going to have reviewed by their peers. Once this has completed, an instructor needs to run the a few scripts. The first is the **Compute Independents**. This allows the instructor to specify the number of previous weeks that are judged for determining the reviewer quality, and what threshold the need to have in order to be independent. (This stage can be done manually as well using the middle button in the second row of icons). Next, the instructor runs the **Assign Reviews** script. This is a stochastic process, so you may have to increase the number of attempts or noise in order to find a good assignment (There is some balancing done on the assignment so that a student doesn't get assigned all authors that have typically performed poorly in the past). If someone has already completed a review, there will be a large warning message that informs you that you're about to delete some student work. **Note** The instructor should always remember to do this between the submission due date and the start of the reviewing period.

The second stage then occurs, when students can perform their peer reviews of other student's work. Once this is done, the instructor needs to run the **Auto-Grade + Assign Markers** script. This allows you to load balance between your markers, and set the threshold for spot checking.

Once this has been completed, the third stage happens when the TAs need to perform all of their assigned marking (found under the "Go to marking page"). Here, markers will see what spot checks they have to perform, or what reviews they need to do because of a student who didn't complete their assigned work. On this page, the markers can mark both the submission and the reviews by clicking on the "--" that is after a student's name. If an automatic grade has been entered, a number will appear with an A after it. If a TA leaves comments with the mark, there will be a + beside the number. If a marker adds a review, then the student will be able to see that the review was given by a marker. If you want to hide this fact, click the "Add Anonymous Review" button, and the student will see your added review as if another student in the course wrote it.

After the marking is all complete, the marks are posted and the students can see what was allowed under the assignment settings. The appeal process will be open until the specified end date. If a student makes an appeal, a message will show up in the marking page beside the offending review. At this point, the TA can engage in a discussion with the student through MTA until the matter is resolved.

## 5 Other Useful Tips

To see what a student actually sees, you can click the "Become Student" link up in the header, then click on their name. As far as MTA is concerned, you are actually them at this point, so if you happen to read an updated appeal, the student will no longer see a notification of an updated response.

You can download all the submissions of a peer review assignment using the **Download Submissions** script. This will give you a zip containing all the data that has been uploaded at this time, which can then be fed into any type of plagiarism checker.

If you don't want appeals, make sure to set the appeal stop date before the mark post date.