

Waseda CourseN@vi to Moodle Converter: nakaMoo

First: 2020-04-12 Ver.002 / Latest: 2020-05-03 Ver.010

This converter is for transferring quizzes made on Waseda CourseN@vi to Moodle. Once you save the quiz on your CourseN@vi to a csv file, this converter will convert it to an xml format file. Import the xml file into Moodle, and you will get them on Moodle.

This converter has been developed by me (Osamu NAKAMURA; SPSE, WASEDA University) for my own purposes. Since the university had no plans to provide such a converter, I decided to make my own. You are free to use it for non-profit purposes.

The purpose of this converter is to migrate from CourseN@vi to Moodle smoothly. However, the converter is also suitable for instructors who want to work with a spreadsheet software such as Excel, rather than typing on a Moodle screen. For example, if you want to create a large number of similar questions and randomize them to examinees, you can create them using the editing function in Excel, save them to csv, and convert to xml with this converter.

For inquiries, please use MyWASEDA to search for Osamu NAKAMURA's contact detail.

(Copyright) 2020 Osamu NAKAMURA

Contents

1	How to use.....	1
2	Notes	2

1 How to use

1. Prepare a Windows PC.

If you do not have Python 3 on your PC, install it. It is not hard.

Please check the option "Add Python 3.x to PATH" when you install Python 3.

See Python for more details.

2. Download the quiz on your CourseN@vi to a local file in CSV format.

See CourseN@vi for instructions on how to save to CSV.

A sample CSV file for testing is included in this package.

3. On Windows PC, place the following three files in the same folder:

The folder can be located anywhere.

- nakaMoo_bat.bat (or a file with a similar name)
- nakaMoo.py (or a file with a similar name)
- A CSV file containing CourseN@vi quiz

4. Drag-and-drop the CSV file onto the .bat file.

Then, a black window appears for a moment, disappears, and a file with the name _xml.xml added to the file name of the CSV you entered will be generated (A file named _log.txt will be generated together, but this is not necessary for ordinary use).

5. Open 'Question bank' in Moodle, import the .xml file, and the quizzes become available on Moodle. See Moodle for instructions on how to import.

Imported questions will be put in a category called "cNavi" on Moodle. If you select that category and check the "Also show questions from subcategories" checkbox, all imported questions will be shown. If you sort those questions, they will be in the order in which they entered on CourseN@vi. Only random questions are put into subcategories. Please see the following "Notes" section for more information about the conversion.

2 Notes

- See elsewhere for information on CourseN@vi, Moodle, and Python.

- Format of the CSV file

The CSV file must follow the format specified by CourseN@vi. Please see CourseN@vi for it. A CSV file saved from CourseN@vi quiz should be fine. You may edit your own CSV file as long as the format is correct.

- It seems that not everything on CourseN@vi can be achieved with Moodle. Unfortunately, there are some features that cannot be migrated if you look at the details. On the other hand, there are some new features you can do with Moodle (such as cloze questions with numbers). They are beyond the scope of this converter, so I hope you edit them manually after importing them into Moodle if you would like to incorporate them.
- The support status of question types and functions in CourseN@vi Quiz
The support status of this converter is as follows. Most question types in CourseN@vi are now supported. If an unsupported item is detected, it will be skipped.

➤ Question types and functions of CourseN@vi that are supported by this converter

- ✧ Essay question
- ✧ Multiple-choice question (single answer)
- ✧ Multiple-choice question (multiple answer)
(partial-point / all-or-nothing-point methods)
- ✧ Shuffle of choices
- ✧ Cloze question (string type; CourseN@vi only has string type cloze questions)
- ✧ Case sensitivity in cloze questions
- ✧ Distinction between full- and half-width characters in cloze questions
(partially supported)
- ✧ Chapter break (start position of chapter questions)
- ✧ Random question break (start position of random questions)
- ✧ Choice question break (start position of choice questions)
(Questions are put into different subcategories in Moodle according to these breaks)

➤ Question types and functions of CourseN@vi that are not supported

- ✧ Maximum number of selections for multiple choices
(not supported by Moodle)
- ✧ Attachments
(you can do it with Moodle, but you cannot do it all at once from CourseN@vi)
- ✧ Time limit for individual questions
(per-question configuration is not supported by Moodle, but can be configured per-quiz)
- ✧ Limit to the number of characters for essay (not supported by Moodle)

- Operation confirmed on Windows

This converter has been confirmed to work on Windows, and the CSV files have also been saved/edited on Windows. I would expect it to work on the command line of any other OS if you have Python installed, but it may need some editing.

- Attachments are not supported.

Even if there is a file attached to a question on CourseN@vi, that information probably is not recorded in the CSV file. Therefore, attachments are treated as if they were not there. Therefore, if you have attachments on your CourseN@vi quiz, you should attach them manually on Moodle after converting the CSV file to xml and loading it into Moodle.

- Question bank categories in Moodle

If you use the converter to import a set of questions into Moodle's question bank, the questions will be placed in the "cNavi" category under the top in Moodle. In addition, questions that have been registered as random questions in CourseN@vi are further divided into subcategories such as "sec04_rnd" under that category. If there are two or more groups of random questions, they are placed in different subcategories, such as "sec04_rnd" or "sec05_rnd". The "_rnd" indicates that the subcategory is a set of random questions. Similarly, a set of choice questions are placed in subcategories such as "sec06_slc" and "sec07_slc"¹.

In order to see all imported questions at once, the Moodle's question bank should be set to display the followings after importing:

- Select a category cNavi
- Also show questions from subcategories Check the box.

The names and locations of these categories and subcategories can be changed on Moodle. For example, you can change the name of the category "cNavi" to "Lec_02_Quiz3" and so on for subsequent use².

The reason for giving random questions a special subcategory is that subcategories are

¹ However, Moodle does not seem to have a type of questions which examinees can choose which to answer to.

² The category name can also be changed in the Python program by 'lab_cat'.

required when questions are randomized on Moodle. However, if you would not like to use these subcategories, you can set `flag_sbct` to 0 near the beginning of the Python program. In that case, all the questions will be placed directly under "cNavi" and no subcategories will be created. On the other hand, if you would like to give a subcategory not only to random and choice questions but also to chapter questions and the others, set `flag_sbct` to 1. In this case, questions other than random or choice questions will also be placed into subcategories such as "sec01" and "sec02" based on the breaks in the CSV file. The default is `flag_sbct=2` (subcategories are given only to the group of random or choice questions).

- Sorting after importing into Moodle

When you import a set of questions into Moodle's question bank, they seem to be sorted by type (multiple-choice, essay, etc.), but by clicking on "question" you can sort them in the order they were set in CourseN@vi³. This is the same every time you open the question bank in Moodle afterwards.

To enable the sorting, the converter inserts a string such as "02-04" or "02-04-rnd03" at the beginning of the question name. The "02" is the chapter number (specified in the CSV, but for convenience if not specified; it is not necessarily a consecutive number), the "04" is the order in which it appears in CourseN@vi (specified in the CSV), and the "rnd03" shows that CourseN@vi gives three questions randomly from random questions to examinees.

- Question titles used in CourseN@vi

Moodle does not seem to indicate the title of the question to examinees. As an option, the converter allows the question title to be displayed at the beginning of the question in Moodle. If you would like to do this, set `flag_qttl` to 1 near the top of the Python program.

- Advantage of using question titles: For example, if the title is "How to write a bibliography," you can make it easier for examinees to understand the question.
- Disadvantage of using question titles: For example, a title like "Question 1" would be confusing when reordering later.

- Random question scores

This is out of the scope of the converter, but just for information: it seems that when you

³ What actually happens when you click on "Question" is that it seems to be sorted in alphabetical and syllabary order.

add questions from the question bank to a random-form quiz in Moodle, the score is automatically set to 1, regardless of how the score is set in the question bank. The score can be easily changed when the quiz is created, so it should not be a major problem. If the questions are not randomized and are combined into normal quizzes from the question bank, the score will be the same as the original setting.

- Does not support the time limit for individual questions

Even if CourseN@vi sets a time limit for each of the questions, it is ignored. The reason is that Moodle does not seem to be able to set a time limit in the question bank. Note that if you create a quiz from the question bank in Moodle, you can specify a time limit for the entire quiz.

- Does not support the limit to the number of characters in essay questions

You can limit the number of characters in essay questions in CourseN@vi. However, it cannot be done with Moodle (Confirmed via the university; but they may enable it in the future). Therefore, even if the number of characters is limited in CourseN@vi, it is ignored.

- Number of lines in the answer column of essay questions

The number of lines in the answer column of essay questions is set to seven. This can be changed in the Python program (the '<responsefieldlines>' tag), or it can be changed in Moodle after importing.

- Partial points for multiple-choice questions

I am not sure how to give partial points in Moodle when a CourseN@vi multiple-choice question allows it. For example, if you allocate zero point to the wrong choice, examinees will get full points by choosing all the choices. To prevent this, CourseN@vi allows you to limit the number of choices that examinees can choose, but Moodle does not seem to be able to do that. Therefore, it is necessary to give some negative points to the wrong choices.

The converter is set to minus the same portion of points for the wrong choices as for the correct ones. However, if you would like to change it so that examinees get zero points for even one wrong choice, you can set flag_mgrd to 1 in the Python program. For your reference, in all-or-nothing multi-choice mode, which does not allow partial points, these worries are irrelevant.

- Answer numbering in multiple-choice questions
In Moodle's multiple-choice questions, answer numbering is set to 'abc'. If you would like to change it, you can edit the Python program. It can also be changed on Moodle after importing to Moodle.
- The number of correct answers in multiple-choice is undisclosed.
The number of correct choices in Moodle's multiple-choice is set to be undisclosed. If you would like examinees to be able to see the number after they complete answering, set `flag_nans` to 1 near the beginning part of the Python program. You can also change it on Moodle after importing to Moodle.
- Choices of multiple-choice questions are indicated with line breaks.
In `CourseN@vi`, it could be specified whether to line up the choices without a line break or one line break at a time. However, it seems that Moodle forces to put a line break per choice. Since there should be no obstacles, the converter ignores `CourseN@vi`'s specification of line breaks.
- Prohibited characters in cloze questions
In cloze questions, if the original correct answer contains `{}`, `#`, `~`, `/`, `"`, or `¥`, it does not work properly. This is because the following actions are required according to the Moodle manual, which the converter does not support:
"If the correct answer contains `}` `#` `~` `/` `"` or `¥` you will have to escape them by putting a `¥` in front of each such character."
- The number of correct answers in cloze questions
Alternative correct answers can be added.
In `CourseN@vi`, each blank field is matched with one correct answer, and the converter transfers it to Moodle. However, it is also possible to add multiple correct answers to a single blank field in Moodle (e.g. giving full points for both "Tokyo" and "Paris"). See the Moodle manual for more information.

It is also possible to edit the CSV file before inputting it into the converter and set two or more correct answers to Moodle. For example, if you have "Tokyo" as the correct answer in `CourseN@vi`, and you would like to add "Paris" and "London" as well, you can edit the relevant part of the "correct answer choice" in the CSV as the following, and the three

will be interpreted as correct answers after importing it into Moodle.

Before editing "Correct Answer Choice" in CSV: Tokyo

After editing: Tokyo~=Paris~=London

- Case sensitive in cloze questions

As for case sensitivity in cloze questions, this converter reflects CourseN@vi's setting.

- The-blank size of cloze questions

The CourseN@vi settings can now be reflected to some extent. You can further make adjustments of the size: When previewed on Moodle,

- if you think the blank size is large, set a smaller value to `fwth` in the Python program.
- if you think the blank size is small, set a larger value to `fwth` in the Python program.

Note that Moodle does not allow you to specify the size of the blank directly; it seems that Moodle automatically adjusts it based on the length of the answer string. So, the converter generates a random string of a certain length, and then adds it to the list of incorrect answers, so that Moodle can create a blank size that matches the length of the string. So, if you edit the question on Moodle after importing, please be aware that a random string like "rqd32eDyewWy5kQ" has been added as an incorrect answer.

If you would not like the blank size of CourseN@vi specification to be reflected by the converter, set `flag_wdth` to 0 in the Python program. In this case, no random strings will be added, and the blank size will follow Moodle's default.

Note that in CourseN@vi, the blank size could only be selected from 15, 50, or 100 (each corresponding to small, medium, and large) in the CSV file. However, when this converter is used, there is no problem even if other values are used.

- Partial points of cloze questions

Moodle does not allow you to set 'zero points unless all the blanks are filled in correctly', so a partial point system must be used. Therefore, if a question in CourseN@vi does not allow partial points, the converter will calculate the score per blank as "full point / number of blanks" and give it as partial points. However, the score must be an integer, so it is rounded off to the nearest integer.

For example, if a CourseN@vi question gives 5 points only for complete answers to 3 cloze blanks, this converter will give 2 points (rounded to the nearest $5 \div 3$) as partial points for each blank. In this case, the full score for the three blanks will be 6 points when the questions are placed in Moodle's question bank, which is different from the original setting. The full points can easily be changed to 5 when creating a quiz in Moodle, so please adjust it that way.

- Handling Spaces in cloze questions

If a correct answer contains spaces such as "research question", it did not work properly on CourseN@vi. However, there is no problem for this converter to process such answers. For example, if you edit a CSV file to include spaces in the correct answer for a cloze question, convert it to xml by the converter, import it into Moodle, and it will work.

- Distinguish between full-width and half-width characters in cloze questions

CourseN@vi has an option to distinguish (or not distinguish) full-width and half-width characters, but Moodle does not have that option. Therefore, the converter ignores the specification in CourseN@vi by default. In this case, Moodle makes a distinction between full-width and half-width characters.

However, if you set flag_znhn to 1 in the Python program⁴, you can partially migrate the CourseN@vi settings to Moodle. This is accomplished by converting the correct answer into a half-width character string and a full-width character string and adding both to the correct answer. Please note the followings if you enable this:

- Considering the followings, it is recommended that examinees be instructed to write in half-width (or in full-width) characters even when using this mode.
- The converter converts the correct character string into half-width and full-width characters and adds them to the correct answer. In this case, numbers, symbols, alphabets, and spaces are converted uniformly, but Japanese characters are not. The examinees will be judged to be a correct answer if their input matches either (1) the original string, (2) the half-width conversion or (3) the full-width conversion.
- If you would like to be case-sensitive and not full-width and half-width in the alphabet, this probably will not work as expected for full-width characters.
- If you are editing your own CSV file and adding multiple correct answers, this option will not work as expected.

⁴ The default is 0 because there are many usage caveats.

- If there is an Error/Warning
Error and Warning are recorded in the _log.txt file. You do not need to refer to it if you feel nothing wrong in conversion.
- Moodle xml format and tags
The format and tags of the xml output from the converter follow the xml obtained by exporting Moodle's sample questions in 2020-Apr. If the version of xml used by Moodle changes in the future, or if the required tags change, the converter may need to make new changes.
- The batch file used on Windows has a timeout setting for debugging. It causes no problem to delete this line of timeout.
- About putting all three files in the same folder
The .bat and .py files must be in the same folder. On the other hand, you can edit the batch file so that the .csv file will work even when they are placed in a different folder (changing `python nakaMoo_vXXX.py %1` to `python "%~dp0/nakaMoo_vXXX.py" %1`). However, if the csv file path contains half-width or full-width spaces or special characters, the Python program would fail (details are not yet confirmed). For this reason, it is instructed that the three files be placed in the same folder.