

AcroTeX.Net

The thorhammer package
Documentation for the system scripts

D. P. Story & Thorsten Grothe

Table of Contents

1	Introduction	3
I	Using the Thor system	3
2	Deploying thmclass.ps1 et al	3
3	Executing thmclass.ps1	3
3.1	On running thmclass.ps1	4
	• The results of running runps1.bat	4
	• Structure displayed	5
	• Structure explained	5
4	Procedures after running thmclass.ps1	9
5	Deploying the quizzes	10
6	Retrieving the quizzes	11
7	Redeploying the quizzes	12
8	How to execute a PS1 file without a BAT file	12
II	Using a standalone system	12
III	My retirement	14

1. Introduction

Part I describes how to execute `thmclass.ps1` and documents all the subordinate files and folders it generates. The `thmclass.ps1` file is a Powershell script file written, for the most part, *by the great Thor himself* to meet the needs of his classes. It can be used “as is” or modified to conform it to your requirements. The script is designed to be used with the `thorshammer` package, but this is not a requirement.

The file structure of Thor may not be for everyone, **Part II** describes a set of “standalone” scripts, which were derived from the scripts in **Part I**.

From the AcroTeX Legal Department. We offer no guarantees and accept no liabilities. Test the scripts extensively to be sure they are working for you. Modify the script as you will, but don’t bother us with your changes, unless they are improvements.

Platforms. The script has been tested on Windows OS and Linux. It is our understanding that Powershell can be installed and executed on MacOS, so this script *may* have use on that platform as well.

Part I: Using the Thor system

Thor has developed his own file structure for his class, this structure may be useful to your own classroom. Here, we describe the system that has been developed. Carefully read this part of the documentation to determine if Thor’s scheme can work for you.

2. Deploying `thmclass.ps1` *et al*

Locate the three files `thmclass.ps1`, `runps1.bat`, and `sample-vars.txt` from the `system-scripts` folder of the `thorshammer` distribution.

The first things first. (1) *Copy* `thmclass.ps1`, `runps1.bat`, and `sample-vars.txt` to a working folder. (2) Open the Powershell file `thmclass.ps1` in a text editor and edit the value of the variable appropriate to your platform `$pathThorshammerWin` or `$pathThorshammerLin`, being sure the default path points to your `thorshammer` folder, and if not, modify the path to point to the folder of the `thorshammer` distribution on your computer.¹

3. Executing `thmclass.ps1`

On Windows, the easiest way to execute `thmclass.ps1` is through the BAT file `runps1.bat`.² Running the script requires the input of eight (8) values for variables, these are listed below and described in the order they are expected to be input.

Eight basic variables

1. `$class`: The name of the class, could be name as listed in the college catalog (MAT101, STA221, HH21B, A21A, WF23, ...) or something more understandable (`calc1`, `alg1`, `euhist`, ...)³
2. `$number`: The quiz or exam number. The value of this variable is not required to be a number.³

¹The purpose of this is to grab the two support files `container.pdf` and `terminate-batch.pdf`. If your TeX installation has installed these two files elsewhere, find them, and modify the path to point to the folder containing these two files.

²The authors have little knowledge of how to do this on Mac OS. An interested user on Mac OS would need to advise us.

³The value of this variable is used in the file name of the TEX template file; consequently, this value should be kept short, yet meaningful.

3. `$subject`: Field of study, such as Math, German, History, and so on.³
4. `$theme`: The topic of this exam: factoring, grammar, 20thcentury, etc.³
5. `$instrName`: Instructor's name. The name should be a single word that identifies the instructor (Thor, Newton, Banach, and so on).³
6. `$date`: The date of the exam.
7. `$time`: The length of the exam; this is, the time the students have to work on the exam.
8. `$classPath`: The full path to the folder that contains the quizzes of the class.

3.1. On running thmclass.ps1

Now that you have read a brief description of the eight variables, rename the file `sample-vars.txt` in your working folder to something short yet appropriate to your work (perhaps rename it to just `class.txt`). In this document, we shall refer to the renamed TXT file as `<varlist>.txt`. Open `<varlist>.txt` in a text editor and modify the left column to values appropriate to your class. The contents of `<varlist>.txt` has the following form (shown with default values which need to be changed):

```
alg1          # class
1             # number
maths         # subject
factoring     # theme or topic
THOR         # instrName
01.01.2019    # date
60 Min.      # time
C:/temp      # classPath
```

Now execute `runps1.bat` with arguments `thmclass` and `<varlist>.txt` from the Command Prompt:⁴

```
runps1 thmclass <varlist>.txt
```

(1)

`thmclass.ps1` reads the data contained in `<varlist>.txt`. Using a batch file is the easiest way of executing the `thmclass.ps1` with argument `<varlist>.txt`.

• The results of running runps1.bat

Executing `'runps1 thmclass <varlist>.txt'`, as just described (see display (1)), creates a series of folders and files within the same folder in which `runps1.bat`, `thmclass.ps1`, and `<varlist>.txt` reside.

Suppose `<varlist>.txt` has the following symbolic values:

```
<class>      # class
<number>     # number
<subject>    # subject
<theme>      # theme or topic
<instrName>  # instrName
<date>       # date
<time>       # time
<classPath> # classPath
```

⁴The Command Prompt needs to be located in the folder that contains `thmclass.ps1`, `runps1.bat`, and `sample-vars.txt`

• Structure displayed

Figure 1 represents the structure created by thmclass.ps1.

```

<working folder>
├─ runs1.bat
├─ sample-vars.txt ..... rename and edita
├─ thmclass.ps1
├─ <class>
│   └─ <instrName>
│       ├── backup
│       ├── anSUS (toStudents)
│       ├── vonSUS (fromStudents)
│       ├── runs1.bat
│       ├── csvTOcfg.ps1
│       ├── classFolders.ps1
│       ├── copyka.ps1
│       ├── delka.ps1
│       ├── moveka.ps1
│       ├── sample-list.csv ..... rename and editb
│       ├── container.pdf
│       └─ terminate-batch.pdf
├─ 00-<class>.cfg
├─ genquiz.ps1
├─ runs1.bat
├─ tex-template.tex
└─ web.cfg

```

^aGenerically referred to as `<varlist>.txt`

^bGenerically referred to as `<list>.csv`

Figure 1: Structure created by thmclass.ps1

• Structure explained

We now attempt the Herculean task of explaining each of these.

`<class>` (folder) The folder created by the `<class>` variable. This folder contains all the working files and folders of the class you are teaching.

`<instrName>` (folder) This is a short name for the instructor. This folder contains a number of other folders and files.

`backup` A folder that will contain the instructor's copies of the quizzes generated. When you open the file `00-<class>.cfg`, you will find: `\instrPath*{<instrName>/backup}`.

`anSUS (toStudents)` A folder to drop the graded quizzes into. This folder name is `anSUS` when the local language is German and `toStudents` otherwise.

`vonSUS (fromStudents)` The target destination of the script file `copyka.ps1`. This folder name is `vonSUS` when the local language is German and `fromStudents` otherwise.

csvTocfg.ps1 A Powershell script that reads the contents of the configuration file $\langle list \rangle.csv$ and modifies 00- $\langle class \rangle.cfg$ by appending appropriate `\classMember` statements.

classFolders.ps1 A Powershell script file that creates class folders based on the contents of the 00- $\langle class \rangle.cfg$ file.⁵

copyka.ps1 A Powershell script that copies the completed student quizzes from their student folders to the folder vonSUS (fromStudents) where the instructor can operate on them.

delka.ps1 A Powershell script that deletes the student quizzes from their personal student folders. (This is done after they are copied using copyka.ps1.)

moveka.ps1 The Powershell script that copies and deletes the student quizzes from their personal student folders. **Not recommended!** I prefer to copy them and verify that they are all successfully copied before deleting them from the personal student folders.

sample-list.csv A comma (or semi-colon) delimited file. Each line of the file consists of three entries and has two formulations:

```
 $\langle first \rangle, \langle last \rangle, \langle rel-path \rangle$ 
 $\langle first \rangle, \langle last \rangle, * \langle full-path \rangle$ 
```

In the first case, $\langle rel-path \rangle$ is the path, relative to the argument of the `\classPath`, to the student folder; in the second case, $\langle full-path \rangle$ is the full path to the student folder.

You should rename this file and modify its contents to reflect the class names and folder names of your class.

container.pdf A PDF file used by Thor's way action script. See the manual for the thorshammer package for details.

terminate-batch.pdf A PDF file used by Thor's way action script. See the manual for the thorshammer package for details.

At the top most level of this folder structure are four files.

genquiz.ps1 A Powershell script that works in concert with `tex-template.tex`. When executed, it reads the $\langle varlist \rangle.txt$ file (one level up), the variable list file used to create the folders and files, it then creates a TEX file named $\langle number \rangle-\langle class \rangle-\langle subject \rangle.tex$, which is a copy of `tex-template.tex`, with certain tagged variables in `tex-template.tex` replaced with their values based on $\langle varlist \rangle.txt$; for example, the tagged variable `#class` in `tex-template.tex` is replaced by the value of the class variable declared in $\langle varlist \rangle.txt$ (perhaps it is replaced by "alg1").

Steps to change the default name. You can change the default name of the generated TEX file by following these steps: (1) open `genquiz.ps1` in your favorite text editor; (2) find the line,

```
$templateName="$number-$class-$subject.tex"
```

and, finally, (3) change the assignment, perhaps to a function of one or more of the above variables.

⁵The sysadmin may now allow creation of folders on the system drive; this script may be used for testing on the instructor's own drive.

`tex-template.tex` There are two renditions (or versions) for the TEX template: (1) a German language version, and (2) an English language version. The Powershell script `thmclass.ps1` detects the language as set by your operating system and writes the proper TEX template. The TEX template, which is `tex-template.tex` by name, contains tagged variables `#class`, `#number`, `#subject`, `#theme`, `#instrName`, `#date`, and `#time`, which are replaced by their counterparts `$class`, `$number`, `$subject`, `$theme`, `$instrName`, `$date`, and `$time`, as declared in `(varlist).txt`.

German. This `tex-template.tex` file is Thor's visualization of what his quizzes should look like. When `genquiz.ps1` is executed, it reads this file and saves it again under a different name, as discussed above.

English. This form of the TEX template is more basic, without Thor's specialized features.

The contents of `tex-template.tex` can be modified to suite your own understanding of what your quiz should look like. Use the tagged variables as you wish in your personal template file.

`00-(class).cfg` The configuration file for building a quiz for the class. Initially, it has the following form:

```
% Reset the paths for \instrPath and \classPath for your system
% Use relative paths here for instructor according to dir, where tex file is located
\classPath{(classPath)}
\instrPath*{(instrName)/backup}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Syntax:
% \classMember{firstname}{lastname}{rel-path}
% \classMember{firstname}{lastname}*{full-path}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Running `csvTocfg.ps1` appends `\classMember` to the file. On running, it might now read,

```
% Reset the paths for \instrPath and \classPath for your system
% Use relative paths here for instructor according to dir, where tex file is located
\classPath{(classPath)}
\instrPath*{(instrName)/backup}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Syntax:
% \classMember{firstname}{lastname}{rel-path}
% \classMember{firstname}{lastname}*{full-path}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\classMember{Muehle}{Waeter}{MW634B/(instrName)}
\classMember{Anton}{Mue1ler}{AM256M/(instrName)}
\classMember{Laura}{Voegt}{LM356B/(instrName)}
```

`web.cfg` The TEX template file uses the `web` package, and as such, uses many `web`-defined commands. These commands can be redefined through the `web.cfg`. The file `thmclass.ps1` creates its own `web.cfg`, for local use. If you open this created file, its contents are,

```
%
% AeB Web Configuration file
%
\ExecuteOptions{dvips}
\webCustomize
```

```

% Insert redefinitions between these two marks
\eWebCustomize

To illustrate the use of the web.cfg, modify the file to now read,

%
% AeB Web Configuration file
%
\ExecuteOptions{dvips}
\lWebCustomize
\author{Herr Dr. Thor}
\university{AcroTeX.edu}
\lWebCustomize

```

Above, we have declared `\author{Herr Dr. Thor}` and `\university{AcroTeX.edu}`. (The `\author` and `\university` commands are defined in the `web` package.) There are no restrictions, other redefinitions of commands defined by other packages are permitted. Freely use the special character '@' without having to type `\makeatletter`/`\makeatother` combination.

The `thorshammer` package defines a new command, that will be picked by `web` itself; the command is `\inputWebCfg`. This command has been embedded in the preamble of the TEX template.

Additional comments on the eight variables. We document how the eight variables are used throughout the files generated by `thmclass.ps1`.

<class> is used to construct a top-level folder containing all the files generated by `thmclass.ps1`. This variable is the base name of the CFG file (`00-<class>.cfg`). `00-<class>.cfg` is input by the template file using the `\InputClassData` macro. It also appears as the argument of the `\thQzHeaderL` and `\title` commands in the template file.

<number> This variable appears in the TEX template file as the argument of the `\DeclareQuiz` and of the `\version` commands.

<subject> This variable appears as an argument of `\thQzHeaderCQ` and `\subject` in the TEX template file.

<theme> This variable appears as the argument of `\thQzName`.

<instrName> This variable appears in several Powershell script files: `copyka.ps1`, `moveka.ps1`, `delka.ps1`, `csvTOcfg.ps1`, and `classFolders.ps1`. In the TEX template file, it appears as the argument of `\author`. Within `00-<class>.cfg`, *<instrName>* appears as an argument of `\instrPath`.

<date> This variable is an argument of `\copyrightyears` in the TEX template file.

<time> This variable appears in the argument of `\keywords`

<classPath> This variable is used in various places: `copyka.ps1`, `moveka.ps1`, `delka.ps1`, and `classFolders.ps1`. It also appears as the argument of `\classPath` within `00-<class>.cfg`.

4. Procedures after running `thmclass.ps1`

After the folders and files of [Figure 1](#) are created, go into the `<class>` folder, you will see `<instrName>` (a folder) and the two files `00-<class>.cfg` and `tex-template.tex`. Now, change directories by moving into the `<instrName>` folder. Now perform the following tasks.

1. **Edit `sample-list.csv`.** Change the name, if desired, of the file `sample-list.csv` and open the renamed file in your text editor. For the purpose of this documentation, we shall refer to the renamed file `sample-list.csv` as `<list>.csv`. The format of each line is exemplified by the three sample lines provided. More symbolically, each line consists of a comma (or semi-colon) delimited list:

```
<first>,<last>,<rel-path> or
<first>;<last>;<rel-path>
```

One or more lines in this file may have an alternate syntax:

```
<first>,<last>,*<full-path>
```

where the first character of the third entry is an `*`, the rest of the entry is a full-path to the student's folder.⁶

To continue, enter the first and last names of each student in the class, followed by the relative path (relative to `\classPath`) to that student's folder. The sample file reads,

```
Mühle;Wäter;MW634B
Anton,Müller,AM256M
Laura,Vögt,LM356B
```

(2)

These sample entries are to be deleted and replaced by the names and folders of the class members. If there are any student exception—one whose student folder is outside `\classPath`—, then a typical entry is has the following form,

```
Laura,Vögt,*C:/Users/dpstory/Desktop/TestFolder/myOtherClass/LM356B
```

For the last entry, the third element begins with an `*`, which signals that what follows is full path to this student's folder.

2. **Run `csvTocfg.ps1`:** Now, execute `csvTocfg.ps1` using the general purpose batch file `run1.bat` with a command line arguments of `csvTocfg` and `<list>` (a CSV file); for example, on the Command Prompt, type,

```
runps1 csvTocfg <list>
```

(3)

and execute by pressing the Enter key. *Do not include the extension in the command line argument.* This will append the student information to the end of the `00-<class>.cfg` one level up. Each student has an entry in `00-<class>.cfg` of `\classMember{<first>}{<last>}{<folder-name>}`. `csvTocfg` also replaces all German umlauts with non-problematic characters like so: `ü -> ue, ö -> oe`, etc.

⁶This is done when the student's does not lie in the path pointed to by the argument of `\classPath`.

3. **Run classFolders.ps1:** The next step, if needed, creates the personal student folders.⁷ This step may not be needed if the folders already exist. Run the `runps1.bat` with a command line arguments of `classFolders` and the CVS file; for example, on the Command Prompt, type,

```
runps1 classFolders <list> (4)
```

and execute by pressing the Enter key. *Do not include the extension in the command line argument.* If the folders already exist, the script does not destroy them and create them again. No harm done, I hope.

4. **Run genquiz.ps1:** (Optional) If you want to use the `tex-template.tex` file to create a TEX file for your quiz, run `genquiz.ps1` with the following Command prompt:

```
runps1 genquiz (5)
```

Executing this line produces a TEX template file named '`<number>-<class>-<subject>.tex`'. The file `genquiz.ps1`, when executed, reads the `<varlist>.txt`, located one directory up, to get the current values of `<number>`, `<class>`, and `<subject>`.

Preliminaries are done! Have fun! You are ready to compose your quiz/exam! Open the TEX template file and modify its contents to suite your needs, removing easy questions and adding difficult ones, or copy one of the demo files from the `examples` folder and modify it. For this part of the process you must be knowledgeable of `exerquiz` and the documentation of the `thorshammer` package. The sample files of `thorshammer` will help to compose the quiz using the approach you want.

Repeat. There is no need to run '`runps1 thmclass <varlist>.txt`' more than once a semester. As you work through the semester, you can copy an old quiz with a new file with a new name and modify for the new material studied or run `genquiz.ps1` to obtain a new clean template file:

```
runps1 genquiz
```

The file `genquiz.ps1` reads the `<varlist>.txt` one level up; you might want to first edit this file and change the value of `<number>`.

The contents of `00-<class>.cfg` remains unchanged; if students drop out of the course simply delete that student's line from `00-<class>.cfg` or if students add the course simply append a line to `00-<class>.cfg` using the template `\classMember{<first>}{<last>}{<folder-name>}`. Read the discussion on how to handle **Exceptions** to these general rules.

5. Deploying the quizzes

The quizzes may be deployed to the student folders in two ways:

1. When a quiz is compiled with the `usebatch` option, the `\sadQuizzes` command executes JavaScript lines when first opened in Acrobat, which builds and distributes the quizzes to the designated instructor folder and deploys individualized quizzes to the correct student folder.
2. When the `batchdistr` option is specified, `\sadQuizzes` does its thing, but does not send the quizzes to the individualized student folders. To deploy them, either use the action sequence **Thor protects** and distributes or just **Thor distributes**. The end result of these two is to deploy the quizzes to their designated student folders.

⁷The sysadmin may not allow creation of folders on the system system drive; this script may be used for testing on the instructor's own drive.

6. Retrieving the quizzes

After the exam period is over the quizzes need to be retrieved from the student folders and placed in a folder where the instructor can grade them. The mechanics of doing this are explained next:

- `copyka.ps1` copies the quizzes from their respective student folders in the class folder declared by the `\classPath` command to the folder designated by the `\instrPath`. These two declarations can be found in `00-<class>.cfg`.
- `delka.ps1` delete the quizzes from their respective student folders in the class folder declared by the `\classPath` command. **Important:** Do not run this script until you've first run `copyka.ps1` and you've *verified* all files have indeed been transferred.
- `moveka.ps1` copies and then deletes the quizzes. **Not recommended**, use a `copyka.ps1` → `verify` → `delka.ps1` workflow.

Danger Will Robinson! These scripts dig down deep into the folder structure (pointed to by `\classPath` or any exception paths (refer to [Exceptions](#)) looking for PDFs in the student's folder, in the subfolder named `<instr>`. If there are more than one PDF in this folder, the scripts will copy, delete, and move all PDFs found in the `<instr>` subfolder.

Running these scripts: In the Command Prompt and with the cursor in the same folder as the scripts you are to execute, type,

```
runps1 copyka
```

```
runps1 delka
```

```
runps1 moveka
```

Exceptions: In an ideal semester, each of the students takes the class for the first time and no additional information is needed by the above three scripts; however, for students who are taking the class again, their class folder may not be located on the `\classPath`. The three scripts make a provision for this. The above PS1 files look for a file named `altclasspaths.txt` in the current directory. The contents of this file is a list of *alternate class paths*. For example, suppose we declare

```
\classPath{C/Users/dpstory/Desktop/myClass}
```

but there are students whose class folder is located on the path

```
/C/Users/dpstory/Desktop/TestFolder/myOtherClass
```

To service these 'exceptional' students, create a file named `altclasspaths.txt` and add the following line to that TXT file

```
/C/Users/dpstory/Desktop/TestFolder/myOtherClass
```

BTW, on Windows, you can use standard Windows notation (with forward slashes) as well:

```
C:/Users/dpstory/Desktop/TestFolder/myOtherClass
```

When any of the above three scripts is run, it will first perform its task for the main `\classPath`, then it will perform the same task for the paths listed in `altclasspaths.txt`.

7. Redeploying the quizzes

After the quizzes are marked and saved in a folder of the instructor's choice, you can (optionally) return the graded quizzes to the students. To do this, use the Thor distributes action sequence.

8. How to execute a PS1 file without a BAT file

There are two methods.

Using the Powershell App: From File Explorer, open a window and navigate to the folder containing the PS1 file. Now select File > Open Windows Powershell (or perhaps the administrator version). In the Powershell command prompt, type

```
Powershell -ExecutionPolicy Bypass -Command .\basename.ps1
```

The `-ExecutionPolicy Bypass` allows the script to run without security warnings.

To run a PS1 file with argument,

```
Powershell -ExecutionPolicy Bypass -Command .\thmclass.ps1 myVars.txt
```

Using the Powershell ISE App: This application has a built-in editor. Start Powershell ISE. Here you can load in your PS1 file by simply dragging and dropping it into the window. Be sure the command prompt is showing the folder where the script file is located. Change directories using the `cd` command, as needed. For a PS1 file that does not have an argument, simply press on the Run Script icon on the toolbar, or choose the Run menu item from the File menu.

To run a PS1 file with argument, type, for example,

```
Powershell -ExecutionPolicy Bypass -Command .\thmclass.ps1 myVars.txt
```

into the command prompt window and press the Enter key to execute that line.

Part II: Using a standalone system

Using the Thor system of class structure (see [Figure 1](#)) may not be for everyone. For this reason, some standalone scripts are provided in the `standalone` folder of the `system-scripts` folder.

Important: The contents of the `standalone` folder should be copied to your working folder, the one that contains the source files for your quizzes you are developing.

Within the `system-scripts/standalone` folder, the following files are provided:

`cpquizzes.ps1` A renamed and modified version of `copyka.ps1`. To use this script, first open it in your editor and modify the following lines:

```
$baseName="baseName"
$classPath="classPath"
$destPath="destPath"
```

The value of `$baseName` is usually one word that represents the instructor (eg, THOR), but it does not have to be the instructor's name. `$classPath` is the path to the class folders. The script searches `$classPath` and copies all PDFs within *any* subfolder with `$baseName` as its name.

It copies the PDFs to `$destPath`. The quizzes all must be uniquely named because they are all copied to the same destination folder (`$destPath`). Any older PDFs (perhaps from earlier quizzes, or actually any latent PDFs) are also copied, so you must set up your protocols appropriately. Execute this script from the Command Prompt:

```
runps1 cpquizzes
```

`rmquizzes.ps1` After you've run `cpquizzes.ps1` and verified that all required files have been copied, you can safely delete (or remove) the quizzes from the student quiz folders. Open `rmquizzes.ps1` in your editor, and modify the values of this variables:

```
$baseName="<baseName>"
$classPath="<classPath>"
```

The meanings of the variables `$baseName` and `$classPath` are the same as above in the description of `cpquizzes.ps1`. Run the script as follows:

```
runps1 rmquizzes
```

These first two scripts are basic and the class can be conducted using only `cpquizzes.ps1` and `rmquizzes.ps1`. The next two scripts can be considered optional.

`mkcfg.ps1` This is a modified and renamed version of `csvT0cfg.ps1`, refer to the description of `csvT0cfg.ps1`. Prior to using this script, delete the file `00-web.cfg` in the current folder (this file is created by `mkcfg.ps1`). Open your editor and modify the values of the following variables:

```
$classPath="<classPath>"
$instrPath="<instrPath>"
```

The values of these variables correspond to `\classPath` and `\instrPath`, respectively. The script is executed from the Command Prompt as follows:

```
runps1 mkcfg <list>
```

where the argument `<list>` is a list of class members and folder locations data, as described in 'Edit `sample-list.csv`' on page 9.

The purpose of this script is to create the `00-web.cfg` file. Once created, move this file to the same file as your quizzes; input this file using `\InputClassData{00-web}`. Examples of the use of `\InputClassData` are found in the `examples/cfgs` folder. As described earlier, the format of this file is,

```
\classPath{<classPath>}
\instrPath{<instrPath>}
\classMember{Muehle}{Waeter}{MW634B/_Thor}
\classMember{Anton}{Mue1ler}{AM256M/_Thor}
\classMember{Laura}{Voegt}{LM356B/_Thor}
```

Here, the sample class members of `sample-list.csv` are shown. For the `\classPath` and `\instrPath`, *use only full paths*. Refer to 'Edit `sample-list.csv`' on page 9 for more information.

`mkfolders.ps1` This is a modified and renamed version of `classFolders.ps1`. This script is used for testing purposes; it can also be used to set up class folders provided you have the necessary permissions. Run this script from the Command prompt:

Open the file and edit the variable

```
$classPath="\classPath"
```

where `\classPath` is the full path to the root of the class folders. Run this script from the Command prompt:

```
runps1 mkfolders <list>
```

where the argument `<list>` is a list of class members and folder locations data, as described in 'Edit `sample-list.csv`' on page 9.

Part III: My retirement

Now, I simply must get back to it. ~~DS~~