Network Operations Documentation

Author : Callum McCormick ICT Technician

GENERAL TIPS CONCERNING THE POWERSHELL SCRIPT TO CREATE THE XML FILE FOR THE STUDENT TIMETABLE

Doc Ref : QEHS-DOC-000

Overview of the student timetable script.

The script can be divided into roughly four parts.

* The preamble contains constant variables such as the directory name, the formulae used within the main body of the script, and ensuring that there is a blank log file ready to be used for monitoring progress.
* The first part (labelled Part A within the script) of the process which imports the data from the provided CSV file and uses that to create a number of arrays.
* The second part (labelled Part B within the script) which takes one of those arrays and filters it for unique values so that we have a student list.
* Finally, the third part (labelled Part C within the script) which goes through that list, and for each student it finds their timetable data in the previously created arrays. It dovetails that with the coding required to turn it into XML data and writes it to the output file.

The output file will be a text file, but it contains all the correct coding so it just requires that the file extension be changed after the script is complete.

Instructions

If required, import the SIMS report into your instance of SIMS. When run, this discovers all students’ full names, registration form name, subject code, staff initials, room number and the period of the day. Run the report and save it into the same folder as the powershell script. Alternatively, save it in a different subfolder and change the directory location within the script.

Once the spreadsheet has been created, it may be useful to replace any other blank cells with a placeholder to indicate that the data is lacking. The process will work fine without it, but it can be useful to indicate within the final spreadsheet that the data is missing rather than it has failed to display. Something simple like a question mark should suffice. To do this:

1. Select the top left cell of the spreadsheet and then hold shift while selecting the bottom right cell. All of the cells in between should be highlighted.
2. Click on the 'Find & Select' button and select 'Go To Special...' from the drop-down options.
3. In the pop-up window that appears, tick the box next to the word 'Blanks' (make sure it is the only option ticked) and click the 'OK' button.
4. Type in your placeholder then press CTRL + Enter to fill all the selected cells with the same thing.

Ensure that the spreadsheet has been saved as a csv file. The default name in the script is 'student\_timetables.csv'. If a different name for the csv is preferred, the default name must be changed within the script.

Open the script 'Create\_student\_timetable\_XML.ps1’. If the directory or input file name require changing, make the alterations to the constant variables at the top of the script now.  
  
Run the script. ***This can take ~50 minutes to run.*** Any errors that occur will be noted within 'log\_file.txt', created within the same directory location.  
  
Once this has finished running, open the file 'script\_out.txt' within notepad. This will contain the updated xml code. Re-save this file as 'stutimteable.xml' (make sure it does not add the txt extension on the end when saving) within the location HAP is expecting to find this data. If a previous version is already located there, it will have to be overwritten.  
  
The timetable on HAP will now immediately begin using the new version of the student timetable.

Constants

These are setup at the beginning. They will rarely change once initially setup.

* “$Directory” refers to the directory in which the input, output and log files are all contained.
* “$OutputFileName”, “InputFileName” and “$LogFileName” are all fairly self-explanatory.
* “$initial\_length” refers to the number of initials that the staff code is expected to be.
* “$teacher\_prefix” refers to the text that appears before the staff’s initials in their email address.
* “$teacher\_suffix” refers to the remainder of the staff member’s email address.
* “$SnippetA” through “$SnippetE” are sections of XML code pre-written to match the ASPX file. Changes to the ASPX file may require changes made to this.

Built-in Presumptions within the code

* The school operates on a two-week timetable.
* The school has six teaching period per day, and an AM and PM registration period.
* The source SIMS report will output period data in the form week number, first three letters of the day of the week, a colon, and a number indicating the day of the working week (1-5). For example 1Mon:1 refers to week 1, Monday, period 1.
* Two students with the exact same name will not be put into the same registration group.
* Staff have a unique identifier that always has the same number of characters (the number specified by the variable “$initial\_length”).
* Staff members’ email addresses contain their unique identifier. The rest of the email address is identical for them all.
* The XML file being created will not be changed in a significant manner (e.g. changing the order of lines).

Suggested actions to be taken if presumptions are false

* If the school does not have a two-week timetable or uses a different number of days per week, the ASPX file will need altering. The function FindPeriodData will need altering to match your layout.
* If the number of periods within a day is different, new blank arrays will need to be created at the beginning of the main body of the script, an ‘if’ statement within part A, and a section of part C which calls the FindPeriodData function will all need altering.
* Students with identical names will either need to be uniquely identified somehow, such as with different middle initials, or the $namesArray will need an additional data point adding to separate them.
* If staff do not have unique identifiers of a common length then the function FindPeriodData will need changing to remove the size requirement.
* If staff do not have consistent email addresses then part C will need changing in order to look them up from some other data source.
* If the XML needs significant updates, the “Snippets” at the top can be changed and part C may need altering so that these constant pieces of text are written into the output files at different times.

Version History

|  |  |  |
| --- | --- | --- |
| **Version iteration** | **Date** | **Reason for change** |
| 1 | 26/04/2016 | Initial output of data to output file. |
| 2 | 27/04/2016 | Outputting period 1 data to a file with correct formatting for xml |
| 3 | 28/04/2016 | Change to using a function for repeated elements. |
| 4 | 28/04/2016 | Adding in ability to output other periods. |
| 5 | 03/05/2016 | Changing function to input an array into it. |
| 6 | 06/05/2016 | Moved position of Write-Host to show correct period. |
| 7 | 13/05/2016 | Added recording of teacher email info |
| 8 | 16/05/2016 | Revamped addition of teacher email info. |
| 9 | 16/05/2016 | Adding AM/PM Data |
| 10 | 23/05/2016 | Re-ordering data output so email info comes before lesson data. |
| 11 | 24/05/2016 | Error-correcting and modifying output text(s). |
| 12 | 24/05/2016 | Ensuring output and log files are present, updating commenting. |