Per-student timetable generation

Wise Timetable — groups and turnuses built from enrolment data

◆ ◆ ◆

Wise Technologies d.o.o. · Wise Timetable desktop application · August 2026

Applies to the 2026 release and later.

Contents

1 What this procedure does

Most timetabling systems schedule a study year: you define groups by hand, you assign students to those groups by hand, and every student then reads the timetable of the group they happen to belong to.

Wise Timetable can work the other way round. You give it two flat CSV files - the teaching plan and the enrolment list straight out of the student information system - and it does the rest:

- every student becomes a group of exactly one person, named after the student ID;
- the number of turnuses (parallel sections) is derived from how many students actually enrolled, not guessed in advance;
- a turnus that would overflow is split automatically, in half, and the overflow becomes a new turnus with the same teaching load, room and lecturer;
- lectures, seminars, labs and studios of the same course are kept consistent - a student is enrolled into every execution type of the course, never just one;
- the generator then places all of it, and every student gets a personal timetable that is correct by construction.

The result is a timetable per student, not a timetable per group that students have to map themselves onto. Students see only their own hours, on the web and in their calendar.

> *This procedure has been in production at a Slovenian faculty of around 2 500 students for several years. It handles roughly 300 teaching lines and 2 700 enrolment rows in a few seconds.*

2 Before you start

The two imports do NOT invent rooms, lecturers or time labels for you - they expect the working file to be ready. Check the following first.

Must already exist	Why
Classrooms, with their real capacity	The generator matches a turnus to a room by number of seats. A room that the courses file mentions but that does not exist is created with the turnus size as its capacity and reported as a warning - which almost always means a typo.
Lecturers (teaching staff)	Lecturers are created from the courses file when missing, so a misspelled surname silently produces a second, duplicate lecturer.
Time labels	A fixed term ("day 4 at 8.00") is only accepted if the start time matches one of the time labels defined in the file.
The teaching weeks of the semester	First and last week in the courses file must fall inside the year defined for the file.
Reservations that block the semester	Exam dates, senate meetings, part-time teaching - anything the generator must schedule around has to be booked before you generate.

The two configuration files must be present in the Wise Timetable program data folder (normally C:\ProgramData\WiseTimetable):

- import_courses_misc.ini
- import_students_misc.ini

3 The files

3.1 import_courses_misc.ini

This file tells the import which column of the courses CSV carries which piece of information. Lines are positional - do not reorder them, do not delete them. A line starting with // is a comment. Save the file in ANSI encoding.

Line	Meaning	Demo value
1	comment header	// Menu->Data->CSV import->Import courses misc
2	number of columns in one CSV line	17
3	column delimiter (first character of the line)	;
4	placeholder that means "empty"	-
5	column: programme name	1
6	column: year of study, 1-9	2
7	column: first teaching week, 1-52	3
8	column: last teaching week, 1-52	4
9	column: fixed term, N:TIME LABEL	5
10	column: course name	6
11	column: execution type	7
12-13	columns: first lecturer surname, first name	8, 9
14-15	columns: second lecturer surname, first name (optional)	10, 11
16-17	columns: third lecturer surname, first name (optional)	12, 13
18	column: weekly hours ("3", or "4+4" for two blocks)	14
19	column: preferred room (optional)	15
20	column: students per turnus - the split threshold	16
21	column: skip pauses, 0 or 1	17

3.2 import_students_misc.ini

The same idea for the enrolment file, plus three lookup blocks that absorb the differences between how the student information system spells things and how the timetable spells them.

Line	Meaning	Demo value
2	number of columns in one CSV line	10
3	column delimiter	;
4	placeholder that means "empty"	-
5	new turnus percentage - see chapter 5	33
6-7	columns: student surname, first name	1, 2
8	column: student ID - becomes the group name	3
9	column: programme of the student	4
10	column: year of study	5
11	column: course name	8
12	column: alternative course name	9
13	courses to ignore, each in quotes, comma separated	"Diploma","Practical training"

Line	Meaning	Demo value
15	programme dictionary, all on ONE line: long name,"SHORT",long name,"SHORT", ...	see below
16	the literal marker line Courses dictionary:	Courses dictionary:
17+	course dictionary, one mapping per line: name as the SIS writes it,"name as the timetable writes it"	see below

The programme dictionary is what lets the enrolment file keep the full official programme name while the teaching plan uses a short code:

```
bachelor study programme (1st cycle) Design,"BDE",bachelor study programme (1st cycle) Media
Technology,"BMT",master study programme (2nd cycle) Design,"MDE"
```

The course dictionary does the same for course titles - including the case where two differently named enrolments belong to one joint course:

```
Courses dictionary:
Colour Metrics,"Colour Metrics and Management"
Colour Management,"Colour Metrics and Management"
Entrepreneurship for Designers,"Entrepreneurship"
```

3.3 The courses CSV - the teaching plan

One line = one execution (one course, one execution type, one programme, one year). This is the file that defines what has to be taught; it says nothing about who attends.

```
program;year;first week;last week;fixed term;course;execution type;tutor 1 surname;tutor 1 name;...;weekly
hours;room;students per turnus;no breaks
BDE;1;21;36;;Design Basics 2;P+S;Novak;Ana;;;;;3;;125;1
BDE;1;21;36;;Design Basics 2;V_BDE;Novak;Ana;Vidmar;Sara;;;2;L-201 (Design Lab);15;1
BDE;1;21;36;4:8.00;Drawing 2;P+S+V;Horvat;Eva;;;;;3;S-301 (Studio);25;1
```

Read the second line as: for programme BDE, year 1, weeks 21 to 36, the course "Design Basics 2", execution type "V_BDE", taught by Novak and Vidmar, 2 hours a week, preferably in L-201, in turnuses of 15 students, with no breaks between hours.

Column	Notes
Fixed term	Format N:TIME, where N is the weekday 1-7 and TIME must match a time label. "4:8.00" means Thursday at 8.00. Leave it empty to let the generator choose freely. A day without a time fixes only the day.
Weekly hours	"3" means one block of three hours a week. "4+4" means two separate blocks of four. Empty means nothing will be generated for this line.
Preferred room	Must match the room name exactly, including any text in brackets.
Students per turnus	The heart of the automation - see chapter 5. Use the real room capacity for labs (15, 16, 25) and a number larger than the cohort for lectures (125).
No breaks	1 = treat the hours as one continuous block, using the wtt.fromschmap / wtt.toschmap mapping.

3.4 The students CSV - the enrolment list

One line = one student enrolled in one course. This is normally an unmodified export from the student information system. Both course-name columns are tried, so an export that carries a long internal title and a short printable title works without editing.

```
LastName;FirstName;StudentId;Programme;Year;Mode;Enrolment;CourseTitle;CourseTitleShort;Semester
Ambroz;Hana;32026001;bachelor study programme (1st cycle) Design;1. year;full-time;first;Design Basics 2
BDE (S);Design Basics 2;summer
Ambroz;Hana;32026001;bachelor study programme (1st cycle) Design;1. year;full-time;first;Colour Metrics
BDE (S);Colour Metrics;summer
```

Import the first cycle and the second cycle files one after the other. They are two files only because the SIS exports them separately - the import merges them into the same timetable.

4 The procedure, step by step

4.1 Prepare the working file

1. Open the last version of the WINTER semester of the same academic year.
2. File > New file. Untick: classrooms, blockades and classroom comments; lecturers; lecturer blockades and comments. You now have a new file that already carries the room reservations from the winter semester - typically the exam dates.
3. Check that there are no other reservations inside the teaching weeks of the new semester. Analysis > Overview of classrooms is the fastest way to see this.

4.2 Book everything the generator must schedule around

1. Reserve the senate room and the senate meetings.
2. Reserve, by hand, the rooms in which part-time (evening) teaching takes place. Part-time timetables are imported LAST - if you let the generator work around them from the start it will interleave full-time and part-time teaching and the result is unusable.
3. Reserve the lecturers who teach part-time.
4. New rooms you do not want the generator to fill: give them one dummy equipment item (R works well) and tick "strictly respect". Cabinets and offices are the usual case.

4.3 Save an empty version - twice

1. File > Save as, under the final name. You now have a file that can be re-generated as often as you like.
2. BEFORE YOU GENERATE, SAVE AN EMPTY VERSION. Then import the CSV files into a new, second version. If a generation run goes wrong you go back to the empty version instead of rebuilding everything.

This is the single most important habit in the whole procedure. Generation is fast; rebuilding the prepared file is not.

4.4 Import the teaching plan

1. Data > Import data from CSV file > Import courses - misc, then choose the courses CSV.
2. Read the summary popup. It reports the number of lines and the number of warnings, and every warning is listed in one window: unknown rooms, unusable fixed terms, missing weekly hours, invalid years. Fix the CSV and re-import rather than patching by hand in the application.

4.5 Import the enrolments

1. Data > Import data from CSV file > Import students - misc, then choose the first cycle file.
2. Repeat for the second cycle file.
3. Read the summary popup again. This is where data inconsistencies surface: a course a student is enrolled in that the teaching plan does not define, or a course that is defined for another programme than the one the student is registered on. Chapter 7 lists every message.

The courses import must run first, in the same session. It writes the programme / execution-type table the student import needs. If you skip it the student import says so and stops.

4.6 Lock and generate

1. Lock the timetables that must not move.
2. Start the generator (the clock button).
3. When it finishes, work through everything that was not allocated.

4.7 Part-time timetables last

Import the part-time (evening) timetables only after the full-time timetable has been generated and accepted. The rooms and lecturers you reserved in step 4.2 are what kept the space free for them.

5 How the automatic grouping works

This is the part worth understanding, because everything else follows from it.

One student, one group

Each student ID in the enrolment file becomes a group containing exactly that one student, with the student's name kept as the group note. Nothing is shared, so nothing has to be reconciled later.

Turnuses grow out of the enrolment count

Each line of the teaching plan starts as one turnus whose nominal size is the "students per turnus" column. As enrolments are read, each student is put into the turnus that currently has the fewest students. A turnus keeps accepting students until it reaches its nominal size plus the new turnus percentage from the ini file - 33 % by default.

Nominal size	Accepted before a split	Effect
15	19	a lab of 15 seats tolerates 19 before a second lab session is created
25	33	a studio group
125	166	in practice never splits - this is how you say "one lecture for everybody"

The split

When the fullest-but-least-loaded turnus can no longer take a student, the turn with the fewest turnuses is split: half of its students move into a brand-new turnus that inherits the teaching load, the preferred room and the lecturers, and the new student joins the original one. From then on the two turnuses fill evenly.

In the demo data set, 22 first-year Design students meet a lab declared at 15 students per turnus. The first 19 fill turnus 1; the 20th triggers a split; the final state is two lab sessions of 11 students each - which is exactly what a room with 16 seats needs.

Every execution type, every time

A student enrolled in a course is enrolled into ALL execution types of that course - the lecture, the seminar and the lab - not only the one that happened to be listed. That is why the timetable of an individual student is complete.

After the import

When the enrolment file has been read, the actual number of students in every turnus replaces the nominal number. From that moment the room matching works with real figures, so the generator will not put 11 students into a lecture hall for 150.

6 Naming rules that decide whether it works

Almost every failed import is a naming problem, not a software problem. Four rules cover all of them.

6.1 Course names must match exactly

The name in the teaching plan and the name in the enrolment file must be identical, or the enrolment name must be mapped to the teaching-plan name in the course dictionary. "Colour Metrics" and "Colour Metrics and Management" are two different courses unless you say otherwise. Trailing spaces are removed automatically; nothing else is.

6.2 The programme must actually teach the course

A course is attached to the programme and year of the lines that created it. If the teaching plan defines "Digital Design 3" only for BDE year 2, then a BDE year 3 student enrolled in it cannot be placed - the import will say so by name. Either add the line to the teaching plan or correct the enrolment.

6.3 Programme-specific execution types

When two programmes attend the same course but not the same sessions, add the programme code to the execution type after an underscore:

Execution type	Means
P+S	lecture and seminar, everybody enrolled in the course
V_BDE	laboratory, students of programme BDE only
P+S_BMT	lecture and seminar, students of programme BMT only
V_BDE_2	laboratory, programme BDE, year 2 only

> *The suffix after the underscore must be exactly the programme short name. "V_BDE3" is not "BDE year 3" - it is an unknown programme called BDE3, and no student will ever be placed into it. Write V_BDE_3.*

6.4 Rooms are matched by their full name

"L-201 (Design Lab)" and "L-201" are two different rooms. If the import has to create a room, it says so - treat that message as a spelling error until proven otherwise.

7 Messages and what to do about them

Both imports finish with one summary window listing everything that was not clean. Nothing is silently skipped any more. The same text is written to wtt_event.log.

Message	Meaning and fix
no course named 'X' or 'Y'	A student is enrolled in a course the teaching plan does not define. Either add it to the courses CSV, or map the name in the course dictionary of import_students_misc.ini.
course 'X' is not taught in PROG year N	The course exists but not for that programme and year. Usually a student repeating a year, or an enrolment recorded on the wrong programme. Add the missing line to the teaching plan or correct the enrolment.
course 'X' has no execution type valid for PROG year N	Every execution type of the course is reserved for another programme. Check the _PROGRAMME suffixes - see 6.3.
programme 'X' does not exist in this file	The programme dictionary did not translate the long name, or the teaching plan never mentioned that programme.
room 'X' did not exist and was created with N seats	Almost always a typo in the room column. Correct the CSV and re-import into the empty version.
fixed term 'X' - start time is not one of the time labels	The time does not match any time label. Both 8.00 and 08:00 are accepted; 8h15 in a file whose labels are quarter-hourly is not.
course 'X' (T) has no weekly duration	The weekly hours column is empty, so nothing will be generated for that line.
column N is missing (M columns expected)	The line has fewer columns than the ini file declares. Usually a stray line break or a semicolon inside a name.
Import the courses file first	The student import was run before the courses import in this session.

8 Checklist

- rooms, lecturers, time labels and semester weeks are in the file
- exam, senate and part-time reservations are booked
- the prepared file is saved
- AN EMPTY VERSION IS SAVED
- courses CSV imported, warnings read and the CSV corrected
- first cycle enrolments imported, warnings read
- second cycle enrolments imported, warnings read
- timetables locked
- generated
- unallocated items resolved
- part-time timetables imported last

Appendix A - The demo data set

Four files ship with this manual. They describe a fictional Wise University of Design and Media: three programmes (BDE, BMT, MDE), 29 teaching lines, 92 students and 274 enrolment rows. They import without a single warning and they exercise every feature described above:

File	Shows
import_courses_misc.ini	the column map for the teaching plan
import_students_misc.ini	the column map, the ignore list and both dictionaries
DEMO_courses_summer_2026.csv	fixed terms, 4+4 double blocks, three lecturers on one line, preferred rooms, programme-specific and year-specific execution types, a course taught jointly to two programmes
DEMO_students_courses.csv	92 students whose enrolments make one lab split into two turnuses of 11, and whose Diploma and Practical training rows are ignored

What the import produces from them:

Course	Type	Per turnus	Result
Design Basics 2	P+S	125	1 turnus of 22
Design Basics 2	V_BDE	15	2 turnuses of 11 - split automatically
Colour Metrics and Management	P+S_BDE	125	1 turnus of 22 - BDE only
Colour Metrics and Management	P+S_BMT	125	1 turnus of 18 - BMT only
Entrepreneurship	P+S+V	125	1 turnus of 20 - BDE and BMT together
Digital Design 3	V_BDE_2	15	1 turnus of 12 - BDE year 2 only