Year Transition

How to move Wise Timetable from one academic year to the next — and how to keep room bookings that reach beyond it.

What the Year Transition screen is for

Every summer the same work has to be done: last year's timetable has to be kept, next year's has to be started from the same rooms and lecturers, and the databases have to end up the right way round. Until now that meant several unrelated steps in different parts of the product, and one SQL script run by hand on the server.

The Year Transition screen puts all of it in one place. It shows the databases your licence covers side by side — their academic year and how much is in each — and every action that changes a database names it, states exactly what will happen, and asks you to type the database name back before it runs.

You reach it from **Settings › Misc › General › Year transition**.

> **Nothing is lost without a copy first**
>
> Every action that changes a database writes a full XML backup of that database before it touches anything. The files go to `%LOCALAPPDATA%\WiseTimetable\yt_backups` and are named after what was about to happen and when — for example `yt_before_copy_wtt_demo_20260830_143012.xml`. The full path of every backup is also written to the event log. If anything goes wrong, open that file with File › Open and export it back. Nothing here is ever deleted automatically.

Before you start

- Make sure nobody else is editing the timetable. The screen loads and rewrites whole databases.
- Know which database is which. The screen shows the schema name and the academic year of each — read them before you press anything.
- Have the new year's dates ready: the first day of the winter semester decides which calendar week becomes week 1. You set it in **Settings › Academic Year Details**, and there is a button on this screen that takes you straight there.
- Allow time. Copying a full timetable between databases takes minutes, not seconds.

The screen, box by box

The three database boxes

One box per database in your licence. Each shows the academic year that database is set to, its name on the server, and how many schedules and bookings it holds. A database the program cannot reach says so and its buttons are greyed out — that is normal for a licence with fewer than three databases, and for a laptop that is off the campus network.

Button	What it does
Copy to DB	Replaces the whole contents of the database you pick in *Target database* with the contents of this one. Use it to seed next year's database from this year's.
Swap DB	Exchanges the contents of this database and the target. Both are written to XML first, so if the second half fails you still have both files.

Button	What it does
Save empty to file	Builds next year's starting document from this database and writes it to a file you choose. Rooms, lecturers, programmes, branches, groups, courses and settings are kept; the schedules, the bookings and the student list are removed; the year is advanced by one. Nothing on the server is changed.

New year database

The ring at the bottom left picks the database that the three buttons under it act on. Nothing is preselected on purpose.

- **Delete all schedules** — removes the timetable, leaves the bookings.
- **Delete all bookings** — removes the bookings, leaves the timetable.
- **Go to Academic year settings** — closes this screen and opens Academic Year Details, where the year and the semester dates are set.

Starting a new year: the usual sequence

1. Open the Year Transition screen and read the three boxes. Decide which database keeps the year that is ending and which one becomes the new year.
2. On the box holding the year that is ending, press **Save empty to file** and save the file somewhere you will find it — next to your other timetables.
3. If the new year needs its own database, use **Copy to DB** from the old one into it first, then delete the schedules and bookings there. Alternatively load the file from step 2 with File › Open and export it into the new database.
4. Open **Academic Year Details** and set the first and last day of both semesters for the new year. This is what defines week 1.
5. Enter the new year's courses and groups, and generate the timetable as usual.

Perpetuum bookings

The problem it solves

A Wise Timetable year is 52 weeks. Week 1 is the week the winter semester starts in, and everything the product stores — every lecture, every booking — is addressed by a number in that range. That is exactly right for teaching, and it is why a timetable can be moved from one year to the next at all.

It is wrong for one kind of request, and every school gets it: *“can we have the big hall for the conference next June?”* — asked in November, for a date that belongs to an academic year that does not exist in the database yet. Until now the answer was a note in somebody's calendar, and the hall was double-booked the following spring by whoever built the new timetable without knowing.

Perpetuum is the setting that lets one database hold weeks outside its own year, so that request can be recorded where the timetable is, and be seen by everyone who looks.

How the weeks are numbered

The extra weeks simply continue the count. Nothing is renumbered and nothing moves; weeks are added on either side of the year that already exists.

Window	Weeks that exist	What that covers
0 before, 0 after *(the default)*	1 … 52	The academic year and nothing else. This is how every database behaved before this feature, and how one behaves until somebody opens the window.

Window	Weeks that exist	What that covers
0 before, 1 after	1 … 104	This year, plus the whole of next year. Enough for the conference in June, and for an exam session that falls after the year formally ends.
1 before, 1 after	-51 … 104	Also last year, for a school that wants last year's room usage visible alongside.
0 before, 2 after	1 … 156	Two years ahead. Useful for a venue that takes long-lead bookings; it is also 104 extra week rows on every save, so take what you need and no more.

A worked example. A school whose year 2026/2027 starts on Monday 21 September 2026 has week 1 = 21–27 September 2026 and week 52 = 13–19 September 2027. Open the window one year forward and week 53 = 20–26 September 2027, week 92 is the last full week of June 2028, and the conference can be booked there today.

What may go outside weeks 1–52, and what may not

Allowed outside 1–52	Stays inside 1–52
Room bookings — a booking that names one or more rooms and nothing else.	Lectures (the timetable itself), and any booking that also names a lecturer, a group or a subject.

The reason is worth understanding, because it is not an arbitrary restriction. A room is the one thing that survives a year transition unchanged: the same hall, the same name, the same internal number. Groups are re-formed every September, subjects are re-entered and lecturers' engagements are rebuilt — and the transition gives all of them new internal numbers. A booking made two years out against “BCIT2-1” would arrive attached to whichever group inherited that number, which is worse than not having been allowed to make it.

The web application enforces the same rule when somebody books there, and says so on the form as soon as the dates leave the year — so nobody fills in a long recurrence only to be refused at the end.

Opening the window — Convert to perpetuum

Set **Years before** and **Years after**, pick the database in **New year database**, and press **Convert to perpetuum**. Nothing is deleted and nothing moves: the extra weeks become available, in this application and on the web timetable, and rooms can be booked in them from that moment.

The window is stored **with the data** — in the database's own settings, and in the XML when the document is saved to a file. It survives every save, every export and every copy between databases, and it does not have to be re-applied. Up to five years either side.

Rolling it forward — Load new from file

This is the step that makes a perpetuum booking worth making, and it is the one that is easiest to misunderstand, so here it is in full.

The conference booking was entered against *this* year's numbering — somewhere above week 52. Next September that same date is an ordinary week of the new year. So when the new year's document is ready, the window slides one year to the right and every kept booking moves 52 weeks down with it:

In the old database	In the new one	
week 53	week 1	the first week of the new year
week 92	week 40	the conference next June, now an ordinary week

In the old database	In the new one	
week 104	week 52	the last week of the new year
week 105 and beyond	week 53 and beyond	still in the future — kept if the window is still that wide

Years after drops by one in the process, because one year of the window has been used up by the transition itself. Everything else in the target database is replaced by the file. Bookings that would land in the past, or beyond the new window, are dropped — and they are still in the backup XML written before the step began. The screen reports both numbers: how many were carried over, and how many were dropped.

Bookings that name a lecturer or a group are not carried at all. They were never allowed above week 52, so there are none out there to carry.

Closing the window — Convert from perpetuum

Puts the database back to weeks 1–52. Bookings outside that range have nowhere left to live and are removed; the confirmation says exactly how many, and the backup XML written first still contains every one of them. Use it when a school decides it does not want long-lead bookings after all, or to clean up a window that was set too wide by mistake.

A year in the life of one booking

1. **November 2026.** The window is opened one year forward on the live database. Weeks 1–104 now exist.
2. **November 2026.** The conference is booked into the big hall for 12–14 June 2028 — week 92. Only the hall is named; no group, no lecturer.
3. **Any time after that.** Anyone looking at the web timetable for that week sees the hall taken, including whoever is building next year's timetable.
4. **July 2027.** The new year's document is prepared — Save empty to file, then the courses and groups for 2027/2028.
5. **August 2027.** Load new from file, into the same database. The conference moves from week 92 to week 40 and is still in the big hall on 12–14 June 2028. Years after is now 0.
6. **September 2027.** The new timetable is generated. The hall is already occupied that week, so nothing is scheduled over it.

Where perpetuum weeks are visible

This application's week views work with the current 52 weeks, as they always have — booking a room two years out is done here, from this screen's settings and the ordinary booking screens. Looking one up is done on the **web timetable**, which reads the full week range from the database and lets anybody page into it. The mobile app is unchanged and shows the current year, which is what a student needs.

If something goes wrong

What you see	What to do
“The name does not match.”	The database name has to be typed exactly as it is shown in the box above. It is not case-sensitive. Nothing was changed.
A database box says it is not available.	That database could not be reached. Check the network and that the server is running. A licence with fewer than three databases will always show empty boxes.

What you see	What to do
“The export did not complete.”	The target database may be half-written. The event log names the backup XML written just before it (see the first page for the folder). Open that file with File › Open and export it into that database.
An exchange stopped halfway.	The message names both XML files that were written before it started. Open the one you need and export it into the database it belongs to.
Bookings disappeared after closing perpetuum.	They were outside weeks 1–52 and were removed on purpose. They are in the backup XML written before the conversion.

Wise Technologies d.o.o. · Ljubljana, Slovenia · wise-tt.com · For help with a transition, contact support before you start rather than after.