



Omniscol user documentation

Generated on 2026-07-07

Contents

1. Getting started

- 1.1 What is Omniscol?
- 1.2 Omniscol's general philosophy
- 1.3 A five-minute guided tour
- 1.4 Architecture, modules and roles
- 1.5 Omniscol plans and options
- 1.6 Choosing the right timetable type: weekly, cyclic, calendar
- 1.7 First login
- 1.8 Set up the school account
- 1.9 Inviting and activating your users
- 1.10 Preparing your data for a mass (batch) import
- 1.11 Getting-started guided tour (the 6 steps of the Home module)

2. Core concepts

- 2.1 Data organization: subjects, teachers, classes, timetables
- 2.2 Class, group, subgroup
- 2.3 Class divisions
- 2.4 Group alignments
- 2.5 Groups of groups
- 2.6 Group hierarchy: parents, children, inherited constraints
- 2.7 Free groups
- 2.8 Courses, lessons, course types
- 2.9 Complex lessons: alternate, concatenated, associated, co-taught
- 2.10 Campuses, sites, classrooms, resources, multi-room
- 2.11 Time grid, time slots and durations
- 2.12 Classroom specialisations
- 2.13 Teacher availability and time constraints
- 2.14 Time constraints: classes, subjects, groups, classrooms and grid
- 2.15 School year, alternate weeks, holidays
- 2.16 Timeline and time navigation
- 2.17 Search and filter in lists
- 2.18 Collaboration between administrators

3. Building a timetable (Timetable management module)

- 3.1 Overview of the Timetable management module
- 3.2 Prerequisites for creating a timetable
- 3.3 Calendar mode — advanced options

- 3.4 Availability and constraints in calendar mode
- 3.5 Date windows
- 3.6 Step 1 — General settings
- 3.7 Step 2 — Sites, time grids, classrooms, resources
- 3.8 Step 3 — Assigning teachers
- 3.9 Step 4 — Creating the classes and their groups
- 3.10 Step 5 — Alignments and groups of groups
- 3.11 Step 6 — Distribute the hours and create the lessons
- 3.12 Automatic classroom assignment
- 3.13 Step 6b — Mass import of courses from a spreadsheet
- 3.14 Step 7 — Automatic generation
- 3.15 Step 8 — Publishing (activating) a timetable
- 3.16 Multiple active timetables in parallel
- 3.17 Manual placement of lessons
- 3.18 Editing a lesson
- 3.19 Off-grid lessons and classes
- 3.20 Locking the position of a lesson
- 3.21 Detecting and resolving conflicts (at generation time)
- 3.22 Diagnosing a failed generation
- 3.23 Visualize, duplicate, reorganize a timetable
- 3.24 Preparing the next school year

4. Dashboard

- 4.1 Overview of the Dashboard module
- 4.2 Using tables and charts
- 4.3 Teacher statistics
- 4.4 Classroom statistics
- 4.5 Subject statistics
- 4.6 Class statistics
- 4.7 Student and resource statistics

5. Everyday use (Timetable module)

- 5.1 Viewing and filtering timetables
- 5.2 Timetable display: grid, list, table, schedule overview, month, side-by-side
- 5.3 Ad-hoc changes to a published timetable
- 5.4 One-off events (outside the timetable)
- 5.5 Print and share
- 5.6 Share a timetable via a public link

6. Absences and substitutions

- 6.1 Overview of the Absences module

- 6.2 Declaring an absence (administrator / teacher / student)
- 6.3 Substitution policies
- 6.4 Single-lesson substitution
- 6.5 Multi-day absences
- 6.6 Class and student absences
- 6.7 Tracking and exporting absences
- 7. Staffing**
 - 7.1 Overview of the Staffing module
 - 7.2 Building a service grid
 - 7.3 Defining the tasks to cover
 - 7.4 Assigning staff
 - 7.5 Create and share rosters
- 8. Administration**
 - 8.1 Users and roles
 - 8.2 Manage administrators
 - 8.3 Custom roles for administration
 - 8.4 Managing teachers
 - 8.5 Managing students
 - 8.6 Manage staff members
 - 8.7 Managing subjects (official and custom)
 - 8.8 Types of course
 - 8.9 School year and holidays
 - 8.10 General school settings
 - 8.11 Advanced settings and customization
 - 8.12 Visibility and login restrictions
 - 8.13 Import and export
 - 8.14 Backup points
 - 8.15 Activity log (logs)
- 9. Integrations**
 - 9.1 Integrations overview
 - 9.2 iCal — subscription and dynamic link
 - 9.3 Omniscol API — authentication tokens
 - 9.4 OAuth2 / OIDC — connect a service to Omniscol
 - 9.5 API tweaks: endpoint overrides and hooks
 - 9.6 Complete data model: JSON entities, relationships and ontology
 - 9.7 MCP — connect an external AI agent to Omniscol
 - 9.8 Built-in AI assistant
 - 9.9 OIDC / SSO — sign-in via an identity provider
 - 9.10 Synchronization with external systems (ERP)
 - 9.11 OneRoster (1.2 and advanced groups)
 - 9.12 Linked accounts and shared resources
- 10. Portals**
 - 10.1 Student portal
 - 10.2 Teacher portal
 - 10.3 Guest portal (public links)
 - 10.4 Public share links
- 11. Display panels**
 - 11.1 Setting up a display panel for a lobby or a corridor
 - 11.2 Setting up a display panel outside a classroom
 - 11.3 Visual customization of display panels
- 12. Migrating from another program**
 - 12.1 Migration from another program — Overview
 - 12.2 Migrating from Hyperplanning (Index Education)
 - 12.3 Migrating from EDT / PRONOTE (Index Education)
 - 12.4 Migrating from aSc Timetables
 - 12.5 Migrating from ADE / ADE Campus
 - 12.6 Migrating from a homegrown Excel spreadsheet
- 13. Higher-education specifics**
 - 13.1 Higher education specifics — overview
 - 13.2 Sessions, cohorts, programs, tracks
 - 13.3 External instructors (adjuncts, visiting faculty)
 - 13.4 Doubled-up classrooms and multi-room exams
 - 13.5 Co-teaching and rotating instructors
 - 13.6 Calendar mode for non-recurring programmes
 - 13.7 Videoconference links per course
 - 13.8 Multi-site in higher education
- 14. Primary / secondary specifics (school)**
 - 14.1 Primary and secondary specifics — overview
 - 14.2 Half classes and electives in class divisions
 - 14.3 Study halls and supervised study (primary and secondary)
 - 14.4 Multi-grade classes
- 15. Use-case scenarios**
 - 15.1 Use-case scenarios — overview
 - 15.2 Scenarios — Timetable creation and generation
 - 15.3 Scenarios — Day-to-day placement and modification
 - 15.4 Scenarios — Multi-group, multi-room, multi-instructor
 - 15.5 Scenarios — Mass operations
 - 15.6 Scenarios — Exams and events

- 15.7 Scenarios — Absences and substitutions
- 15.8 Scenarios — Reporting and statistics
- 15.9 Scenarios — Distribution and sharing

16. **FAQ — frequently asked questions**

- 16.1 FAQ — general questions
- 16.2 FAQ — Timetable creation
- 16.3 FAQ — Data import
- 16.4 FAQ — Generation algorithm behavior
- 16.5 FAQ — Display and interface

- 16.6 FAQ — higher education use cases
- 16.7 FAQ — special cases and advanced configurations
- 16.8 FAQ — Security and hosting
- 16.9 FAQ — Pricing and licenses

Glossary

(51)

Index

Generated from the maintained help/en/ corpus on 2026-07-07. UI labels and i18n strings are resolved to their real en wording; diagrams are rendered. Interactive step-by-step walkthroughs are omitted.

1. Getting started

1.1 What is Omniscol?

Source: <help/en/overview/what-is-omniscol.md> · id: [overview.what-is-omniscol](#) · Updated: 2026-06-13

Omniscol is **online constraint-based scheduling software**, designed for institutions ranging from primary school to higher education and continuing education. It lets you create weekly, cyclic or calendar (dated, non-recurring) timetables, then publish them to the relevant users.

The product combines manual entry, conflict diagnostics and automatic generation by a [solver](#). The domain details are documented in the specialized pages.

How it works, in general

The standard workflow with Omniscol:

1. Enter the data (users, subjects, sites/rooms, classes)
↓
2. Configure the timetable (hourly volumes, alignments, constraints)
↓
3. Generate or place (automatic solver or manual positioning)
↓
4. Check and arbitrate (conflicts, adjustments, locks)
↓
5. Publish (the timetable becomes visible to users)
↓
6. Live with the timetable (daily changes, absences, substitutions)

Each step has a dedicated module. Steps 1 to 5 are occasional (before a school year starts, a semester or an overhaul). Step 6 corresponds to daily use.

Timetable types

Omniscol supports **three types** of timetable. The availability of each type depends on your plan.

- **Weekly** — recurring lessons on a model week, with or without A/B alternation. This is the standard case in primary and secondary education. Available on all plans.
- **Cyclic** — recurring lessons on a cycle of N numbered days (typically 6 or 8), different from the 5- or 7-day week. Common in North American systems. Available on Premium accounts.
- **Calendar** — lessons dated one by one, with no recurrence. The preferred mode of higher education and continuing education. Available on Premium accounts.

On a Premium account, all three types can be used. And thanks to [Multiple active timetables in parallel](#), a feature included in Premium, they can even be combined over the same periods. See [Choosing the right timetable type](#) to help you decide.

Philosophy: data first

Omniscol starts from an observation: a timetable is not a table to fill in, it is a **constraint-based optimization problem**. The software is designed around five principles:

1. **Data before computation**: sites, rooms, teachers, classes, groups, subjects and hourly volumes must be reliable before launching a generation.
2. **Separation of structure / daily life**: building a timetable and handling the day's changes are two different activities, with different interfaces.

3. **Explicit constraints:** impossibilities are flagged as **conflicts** or unplaced lessons; the user keeps the final say on business decisions.
4. **Drafts and versions:** an unpublished timetable can be tested, duplicated or rebuilt without changing what users see.
5. **Data ownership:** JSON export or export through the built-in spreadsheet, and reversibility.

Reference page: [Omniscol's general philosophy](#).

Input / output data

On the **input** side, Omniscol accepts:

- one-by-one entry in the interface;
- bulk import by copy-paste from a spreadsheet, with verification before applying;
- assisted migrations depending on the exports available on the source side;
- ERP/IT-system connectors through the [synchronization with external systems](#) (Aurion, Auriga; adding a new ERP as a project);
- the [Omniscol API](#) depending on the account's rights and options;
- [OIDC / SSO](#) depending on the contract and configuration.

On the **output** side:

- responsive web consultation;
- [iCal](#) subscriptions;
- signed, expirable public links;
- display panels;
- Excel / CSV / PDF / JSON exports depending on the screen;
- a documented REST API, with a scope limited by the authorized endpoints and options;
- [MCP](#) for compatible AI agents on Premium accounts.

Modular organization

The software is organized into two families of modules.

Day-to-day modules

- **Home** — a summary view of the day and a getting-started checklist.
- **Timetable** — consultation and one-off changes.
- **Dashboard** — occupancy and service statistics.
- **Absence management & substitutions** — unavailability declarations and assignment of substitutes.

Configuration modules

- **Timetable management** — creating, configuring, generating and publishing timetables.
- **Administration** — users, subjects, school years, settings, import / export and integrations.
- **Staffing** — task-based staff scheduling when the module is active; it can also be sold as a standalone offer.

Roles and access

Four profiles structure access on the school side. **Administrators** (scheduling managers, school leadership, IT department) configure the account and build the timetables. **Teachers** consult their schedule, enter their availability and declare their absences. **Students** consult their personal timetable. **Staff** (student supervision teams, supervisors) come into play when Staffing is used. Signed sharing links additionally provide account-free access to a precise scope.

Roles can be combined: a teacher who takes part in scheduling can be both a Teacher and an Administrator. The roles are detailed in [Architecture and roles](#).

The [Custom roles](#) option lets you restrict an administrator account's rights module by module and operation by operation.

Plans and options

Omniscol offers several plans: **Lite** (independents, solo trainers, very small institutions), **Staffing** (the Staffing module only), **Standard** (a typical school), **Standard Plus** (Standard + Staffing) and **Premium** (higher education, continuing education, complex multi-site organizations). Some features are included by default in Premium — cyclic and calendar timetable types, calendar availability, several active timetables — while others are enabled according to the contract (snapshots, logs, real-time collaboration, linked accounts, custom roles...).

The details of plans, options and how they are activated are centralized in [Omniscol plans and options](#).

Differentiating points

- **Automatic generation**: the solver places lessons while honoring the hard constraints and optimizing the preferences.
- **Manual control**: the user can always inspect, move, lock or arbitrate.
- **Multi-channel distribution**: web, iCal, exports, public links and display panels.
- **Reporting**: dashboards and exports to track hours, room occupancy and the indicators useful for audits.

Architecture and hosting

The architecture, security and hosting aspects are described in [Architecture and roles](#) and [FAQ — Security and hosting](#).

What's next

- For an interactive tour: [A five-minute guided tour](#).
- To set up your account: [First login](#) then [Set up the school account](#).
- To understand the core concepts: [Class, group, subgroup](#).
- To create your first timetable: [Timetable management module](#).
- For frequently asked questions: [General FAQ](#).

See also

[Omniscol's general philosophy](#)

[Architecture and roles](#)

[Choosing the right timetable type: weekly, cyclic, calendar](#)

[Omniscol plans and options](#)

[A five-minute guided tour](#)

1.2 Omniscol's general philosophy

Source: [help/en/overview/philosophy.md](#) · id: [overview.philosophy](#) · Updated: 2026-06-13

Before the features, a few principles structure the Omniscol experience and explain why the software makes certain decisions rather than others.

The business context

A timetable is not just a table to fill in. It is a **constrained optimization problem**:

- many constraints, often implicit;
- constraints that sometimes contradict each other;
- high operational stakes: the school year start, the semester or the period must work.

Omniscol is designed as a **constraint-based planning assistant**: it speeds up the work, flags impossibilities and leaves business decisions to the user.

What Omniscol is not

Category	What it does	Omniscol?
School ERP / SIS	Administrative records, enrollment	No. Omniscol consumes or synchronizes this data.
Digital workspace / LMS	Pedagogy, course content, communication	No.
Grades / attendance	Assessments, signatures, class diary	No.
Omniscol	Organizing time: who, what, where, when	Yes.

Omniscol is meant to interface with other systems via imports, exports, iCal, a documented REST API, OIDC / SSO depending on the contract, and [synchronization with external systems](#). It does not try to replace the business tools that cover other scopes. See also [integrations.partners](#).

The five founding principles

1. Data before computation

You cannot generate a reliable timetable without healthy data: sites, rooms, teachers, classes, groups, subjects, hour volumes, availability and constraints.

2. Separating structure from daily life

Building a timetable and living with a published timetable are two different activities. The first mostly happens in [Timetable management](#), the second in [Timetable](#), [Absences](#) and [Dashboard](#).

3. Hard constraints and optimization constraints

Some constraints make a timetable invalid and are treated as strict constraints: a teacher in only one place at a time, a class without collisions except for groups in a division, a standard room assigned to a single lesson, unavailability, capacities, material resources and inter-site travel.

Other constraints are used to improve the solution: undesirable availability, gaps, day balance, pedagogical order between subjects, hour maxima or minima, number of days of presence. They are evaluated as penalties that the solver tries to reduce.

4. Drafts and versions

An unpublished timetable has no impact on what users see. You can duplicate, test, compare, lock some lessons, run another generation and publish only when the result is right.

Snapshots, when enabled, add a safety net for going back or restoring some data.

5. The user stays in control of the data

Omniscol provides full JSON export, business exports per screen and reversibility. Automations do not replace human decisions.

Conflicts are information, not errors

In manual entry, Omniscol flags conflicts but lets the user decide. This is useful for real special cases: a room for extra-time exam arrangements, a deliberate overrun of a theoretical capacity, assigning a room outside its specialization for a local reason.

In automatic generation, the solver does not deliberately produce major collisions. If no complete solution is found, it keeps the best computed timetable and leaves the lessons impossible to place in the list of unplaced sticky notes.

[Locking a lesson](#) anchors a chosen placement: the next generation adjusts the other lessons around that lock.

Multi-timetable as a modeling principle

Omniscol is natively **multi-timetable**. Several timetables can coexist for the same school:

- distinct school years;
- semesters or periods;
- logical separation by campus, department or cycle;
- a combination of different types, for example recurring weekly and dated calendar.

In Premium, several timetables can be **active in parallel** — a specific activation is also possible on some Standard accounts. A school can have a preparatory cycle with two weekly timetables, one per semester, a graduate cycle on a non-recurring calendar and simultaneously, an 18-month ExecMBA program that starts and ends offset from the other schedules and from the school year. With Omniscol, you can create each timetable separately while benefiting from cross-conflict management during planning, then activate everything in parallel, simultaneously, so that the final result is a merged view, transparent to end users.

Optimization AI and external agents

The **solver** is a **neuro-symbolic Monte-Carlo metaheuristic** optimization AI. The engine explores placements under constraints, looks for a valid solution then optimizes the penalties, with impressive speed (e.g. less than a minute for a middle school with 16 classes). If no valid solution is found, a relaxation system sacrifices lessons intelligently, and still proceeds to optimize the result. Omniscol is an AI-first company, born from a long "deeptech" R&D phase.

Omniscol also has other symbolic-AI algorithms, to check conflicts, detect configuration inconsistencies, make improvement suggestions or configuration alerts, pre-filter the best solution from a list of proposals, or automatically allocate a subset of rooms to a subset of lessons, all in the browser, immediately.

Omniscol also exposes API tools to compatible agents via **MCP** when the option and the rights are active. This makes it possible to interact with Omniscol in natural language, to extend what the graphical interface offers, by fetching the relevant data directly from the Omniscol server. These agents act within the scope of the provided token; read-only uses are the safest, and any write action must remain verified by the user. Currently, the best AI agent for interacting with Omniscol is Claude, in its Desktop version.

See also

[What is Omniscol?](#)[Architecture and roles](#)[Choosing the right timetable type](#)[Conflict](#)[Solver](#)[Multiple active timetables in parallel](#)

1.3 A five-minute guided tour

Source: [help/en/overview/quick-tour.md](#) · id: [overview.quick-tour](#) · Updated: 2026-06-25

This guided tour takes you through the main Omniscol modules in a few minutes. If you are discovering the software, this is the right place to start.

The module menu on the left

On the left of every screen, a vertical menu lists the modules available for your role. From top to bottom, in order of daily use:

1. **Home** (the 🏠 icon) — a getting-started guide as long as no timetable is published yet, otherwise a personalized home screen.
2. **Timetable** — your main day-to-day view.
3. **Dashboard** — statistics.
4. **Absence management** — managing absences and substitutions.

5. **Timetable management** — creating and structuring timetables.
6. **Administration** — users, subjects, settings.

Depending on your contract, other modules may appear, for example **Staffing** (task and supervision grids, for student supervision teams).

The timeline

At the top of most screens, a **timeline** (by week or by month) represents the school year. It lets you navigate through time and see at a glance the periods where timetables are published (colored weeks) and the holidays (grayed-out weeks). This timeline is rich in options and uses. It can switch to a month or year view (or even, explicitly, a date-to-date view). In timetable management, it is colored according to the published timetables — the gaps are the white weeks. When a calendar-type timetable is displayed, light-blue markers at the top indicate where lessons are located on the schedule overviews shown on screen.

The ◀ and ▶ arrows at either end switch between school years.

Follow the guide!

Welcome to Omniscol. This overview introduces the software's modules and how they fit together.

1. The **help button** at the top right opens this help at any time, on any screen. It knows which screen you are on and suggests the most relevant page. Next to it, the legend button displays a description of the elements on screen.
2. **Home** (🏠 icon): until you have published your first timetable, this module guides you step by step. Follow [the steps](#), tick the boxes, and come back whenever you like.
3. **Timetable**: day-to-day consultation. All the filters (class, teacher, room, group, subject, student). Several display modes: standard grid, list, spreadsheet-style table, schedule overview, hourly schedule, day, month and side-by-side. A schedule overview only appears once a first timetable containing positioned lessons has been published.
4. **Dashboard**: statistics for the actual (operational) timetable. Hours per teacher, per class, per room, per subject. Filterable by period. Exportable to CSV and Excel. An API with even more detail.
5. **Absences**: centralized management. Declarations by teachers or students (with administrative validation), or entered directly by the administration. Assignment of substitutes to absent teachers. Class absence (work placement, school trip...).
6. **Timetable management**: creating the timetables themselves. Sites, classes, groups, teachers, subjects, lessons. Automatic generation or manual positioning. Publication, so that a theoretical timetable becomes operational.
7. **Staffing** (if the module is enabled): the scheduling of supervisors. Defining tasks and their specifics, assigning staff. A duty sheet ready to hand out.
8. **Administration**: users, subjects, school years, settings, import/export. Mostly used when the account starts out.

What's next

- To set up your account from scratch: [First login](#).
- To understand the core concepts (class, group, division, alignment, etc.): [Core concepts](#).
- To create your first timetable: [Overview of the Timetable management module](#).
- To see how Omniscol integrates with your IT systems: [Integrations overview](#).

🔗 See also

[What is Omniscol?](#)

[Architecture and roles](#)

1.4 Architecture, modules and roles

Source: [help/en/overview/architecture-and-roles.md](https://help.en/overview/architecture-and-roles.md) · id: [overview.architecture-and-roles](#) · Updated: 2026-06-13

This page summarizes Omniscol's application architecture and the roles visible on the school side. Precise contractual commitments (availability, backups, retention, monitoring) belong to the approved legal or commercial documents.

Hosting and reliability

Omniscol is hosted in the European Union, more specifically on AWS Paris and Scaleway Paris. The product uses backup, monitoring and logical isolation mechanisms between school accounts; the details applicable to an account depend on the contract and the environment concerned.

See [FAQ — Security and hosting](#) for the user-facing details.

Software architecture

- **Web application**: the application loads once, then the screens are rendered dynamically with REST/JSON exchanges (Single Page Application).
- **CouchDB** as the main database.
- **Redis** as the database for application-side synchronization and logs.
- **TypeScript** for the interface (webapp).
- **Node.js / Express backend** orchestrating the internal engines (api, webapp, portal, panel, ical, mcp, oauth2, etc.).
- **C++** for the generation algorithm, run on dedicated VMs started on demand.

Security

- **HTTPS** for exposed communications.
- **Cloudflare** as the exclusive entry point (anti-DDOS protection).
- **Passwords pre-hashed client-side with scrypt**, then re-hashed and salted server-side. They are not stored or transmitted in plain text.
- **JWT authentication** with signing keys and a short lifetime; a key rotation or revocation can invalidate signed tokens.
- **Sharing links** — signed, expirable, tied to the account that generated them and invalidated by a password change or deactivation of that account.
- **API tokens** with a root key and derived tokens, endpoint scopes and expirations.
- **OIDC / SSO** available depending on the contract and the account's configuration.
- **JSON export and internal backups** for data reversibility.

User roles

Omniscol distinguishes several roles on the school side:

Role	Typical use
Administrator	Planning managers, school leadership, IT department
Teacher	Teachers, guest instructors, trainers
Student	Students, pupils, learners
Staff	Pastoral staff, education assistants, monitors if the Staffing module is active
Sharing link	Access via a signed URL to a precise scope

The mapping between these labels and the technical role identifiers is detailed in [Users and roles](#).

A single user can **hold several roles at once**. For example, a teacher who takes part in planning can be both Teacher and Administrator.

The Sharing link role does not correspond to a regular user account. These are signed links: timetable web links are read-only, while some targeted links can allow a limited action, such as entering a teacher's availability until an expiration date.

There are also internal roles reserved for Omniscol's teams: platform administration, translation, sales operations and reseller partners. Some of these accounts are allowed to sign in to school accounts, for maintenance purposes. They then act as superadministrators of the school, with more capabilities than an administrator: importing raw data into the database, enabling or disabling paid options, signing in on behalf of a user. These accesses trigger visible login alerts and their tokens expire very quickly. With great power comes great responsibility.

OPTION: CUSTOM ROLES

Custom roles

The [Custom roles](#) option restricts an administrator account's rights module by module and operation by operation. It is used to delegate part of the administration without granting all global rights.

Modules

The modules visible in a standard account:

- **Home** (`home`) — entry page and summary of the day.
- **Timetable** (`schedules`) — consultation and one-off changes.
- **Dashboard** (`dashboard`) — occupancy and service statistics.
- **Absence management** (`absences`) — declarations and substitutions.
- **Timetable management** (`timetables`) — building, generating and publishing timetables.
- **Administration** (`admin`) — users, settings, import / export and integrations.

Depending on the contract, other modules can appear, notably **Staffing** for planning pastoral and supervision tasks.

Multi-school

Each school is logically isolated on the server side and accesses its own domain. A school's data is not visible to other schools from the application. However, on request, a communication channel can be enabled between separate accounts, in order to share room and teacher occupancy conflicts. This is very useful for shared buildings, or for a group of schools sharing resources.

See also

[General presentation](#)[Guided tour](#)[Users and roles](#)[Custom roles](#)[Omniscol plans and options](#)[Security and hosting](#)

1.5 Omniscol plans and options

Source: <help/en/overview/plans-and-options.md> · id: [overview.plans-and-options](#) · Updated: 2026-06-13

Omniscol is sold under a simple model: **several plans, core Premium features**, some **contract-based activations** and a standalone **Staffing** offer. The exact price and scope depend on the contract with your school.

The plans

Plan	Who is it for?	Included
Lite	Very small schools, independents, solo trainers, trials	Timetable generation and printing. No publication, no school years, no multi-site, no student management , limited resources. Visible modules: Timetable management + minimal Administration.
Staffing	Schools that only need staff duty planning	Only the Staffing module, for supervision duties, study hall and group supervision
Standard	Typical school (primary, middle, high school), a single planning team, weekly timetable	Lite + publication, school years, multi-site, student management, daily planning, absences, dashboard, display panels, iCal, limited read-only API
Standard Plus	Schools that need Standard plus Staffing	Standard + Staffing
Premium	Higher education, continuing education, more complex multi-site or multi-program organizations	Weekly, cyclic or calendar timetables, dated availability, several active timetables, more configuration options (lesson status, off-grid placement, one-off events, lesson delivery mode, videoconference links...), full token-based API, MCP, OIDC / SSO, external synchronization, advanced options depending on the contract (collaboration, snapshots, custom roles).

Not all features are activated the same way:

- some features are **included in Premium by default** (calendar mode, calendar availability, several active timetables);
- others are **activated according to your contract**, most often on Premium accounts;
- a few cases can still be **activated exceptionally** on a Standard account when there is a specific need;
- **Staffing** also follows its own logic, since it can be sold on its own.

Standalone Staffing offer

The Staffing module is also sold **without the rest of Omniscol** for schools that only need supervision / duty management. The account then exposes only the Staffing module, absences for assignable staff and the necessary user screens. No timetables, no teaching dashboard, no display panels. See [Overview of the Staffing module](#).

Contract-based activations

Feature	Description	Activation
Custom roles	Custom roles with a fine-grained permission matrix.	According to the account's contract
Snapshots	Backup points, digest, restoration and rotation according to a contractual quota enabled by Omniscol.	According to the account's contract
Collaboration	Collaboration between administrators, real-time presence depending on configuration.	According to the account's contract
Logs	Log consultation screen with retention depth enabled by Omniscol according to the contract.	According to the account's contract
Linked accounts	Different Omniscol accounts linked to share room, teacher or class occupancy between schools that each have a separate Omniscol subdomain.	On request, after scoping
Several active timetables on a Standard account	Exceptional case for some genuinely separate school accounts (for example a self-contained middle-and-high school, or a primary-middle-high school).	Specific activation, with adapted scoping and billing
Direct assignment of students to classes and groups in timetables	Usually, schools plan without the precise list of students and their assignments. They therefore work with classes and groups through a theoretical number of expected students, specifying the organizational logic; once the timetable is finished, students are (possibly) assigned to classes and groups afterwards. However, another way of working exists, common in training organizations, where the precise list of students on each class and each group is needed upfront, when designing the timetable; the non-conflict logic between groups (mutual exclusion) is then automatically deduced by Omniscol. This changes the data model and the way the product works, and can therefore only be enabled as an option by the Omniscol team.	On request, after evaluating the specific planning approach
Staffing	Supervision / duty management module (exams, study hall). Also sold as a standalone offer.	As a standalone offer, in Standard Plus or according to the contract
Additional modules	Possible additional modules.	Depending on the module and the contract

Notable features explained

Calendar mode

Positions lessons **date by date** rather than on a typical week. It is a feature **included in Premium by default**, essential for higher education, continuing education and training centers. See [Calendar mode](#).

Groups of groups

Combine several groups — from the same class or from different classes — into an entity that can be scheduled as a single block. **Available on all accounts, whatever the plan**, and on all timetable types (weekly, cyclic, calendar); no particular activation is required. See [Groups of groups](#).

Several active timetables in parallel

Publish several timetables simultaneously over the same weeks, merged dynamically. This is included in Premium by default. Omniscol can also enable it exceptionally on some Standard or mixed school accounts when the need and billing justify it. See [Multiple active timetables in parallel](#).

Linked accounts

Link several Omniscol accounts to share the occupancy of common resources: teachers, rooms, buildings or classes depending on the case. Activation is done on request by Omniscol, because the linked accounts and shared resources must be scoped. See [Linked accounts and shared resources](#).

Custom roles

Finely restrict an administrator account's rights — module by module, with a choice of operations (view / edit / delete) and the ability to switch off an entire module in one click. See [Custom roles](#).

Snapshots

Save the state of the account at a given point in time and restore it later. Invaluable before a risky operation (massive overhaul, large import, start of the year). See [Backup points](#).

Real-time collaboration

Several administrators can work on the same account with presence indicators when the option is enabled. See [Real-time collaboration](#).

MCP

MCP lets you connect a compatible AI agent such as Claude (recommended), with its account identification via OAuth2. See [MCP](#).

Logs

When the option is enabled by Omniscol, the Logs screen lets you view and export the entries logged by Omniscol. It should not be presented as a full audit with before/after diffs or request replay. See [Activity log \(logs\)](#).

ERP synchronization

Synchronization with external systems is a contract-based activation, generally on Premium accounts, when a connector is configured. Aurion and Auriga connectors exist; adding a new ERP is handled as a project, with a documented API, a sandbox and mapping scoping. See [Synchronization with external systems](#).

Standalone Staffing

The Staffing module is available separately, including as a standalone offer (without the rest of Omniscol). See [Overview of the Staffing module](#).

How to know which plan and options are active

Visible in Settings — a section at the bottom of the panel, which displays the active plan and the subscribed options. The plan + the options determine which modules are available in the left-hand menu.

See also

[Multiple active timetables in parallel](#)

[Linked accounts and shared resources](#)

[Calendar mode](#)

[Custom roles](#)

[Snapshots](#)

[Real-time collaboration](#)

[MCP — connect an external AI agent](#)

1.6 Choosing the right timetable type: weekly, cyclic, calendar

Source: help/en/overview/timetable-modes.md · id: [overview.timetable-modes](#) · Plan: premium · Updated: 2026-06-13

👑 PREMIUM

PREMIUM

On a Premium account, you can choose between three timetable types at creation time. The fundamental distinction is between **recurrence** (Weekly and Cyclic: a pattern repeats) and **non-recurrence** (Calendar: each lesson is dated individually). This choice shapes everything that follows; it is therefore important to understand it before creating a timetable.

Automatic generation by the solver is available for all three types; the advantages and limitations specific to each type are described below.

In one sentence

Type	Recurrence	When to use it
Weekly	Recurring (model week)	Recurring lessons on a model week . Primary / secondary education.
Cyclic	Recurring (N-day cycle)	Recurring lessons on an N-day cycle (different from the 5- or 7-day week). North American systems, international schools.
Calendar	Non-recurring	Lessons dated one by one , with no recurrence. Higher education, continuing education.

Weekly — the standard case

This is the default mode. It suits any school whose lessons follow a **recurring model week**, with or without A/B alternation.

Advantages:

- Simple logic: you enter a model week, then choose the weeks where it applies.
- Holidays removed automatically.

Limitations:

- Not suited to institutions where every week is different (one-off lessons, alternating external instructors).
- Lessons that take place only once (a conference, an exam on a specific date) must be added as one-off changes or through a separate timetable.

Cyclic — for non-weekly cycles

The cyclic mode suits institutions that organize lessons on an **N-day cycle** (typically 6 or 8) different from the 5- or 7-day week — common in North American systems and in some international schools.

It works much like the weekly mode, with a cycle of N numbered days (Day 1, Day 2...). The cycle ↔ weekday mapping is built dynamically from the **publication of the timetable**, that is, when choosing the weeks where the cycle applies.

Calendar — for higher education and continuing education

The calendar mode differs significantly from the other two: each lesson is positioned on a **specific date**, with no recurrence.

It is the preferred mode of:

- higher education (business schools, engineering schools, universities),
- continuing education,

- training centers where lessons do not recur every week.

Compared with a classic agenda or an ERP/school administration software, you then benefit from the features typical of higher education (see [calendar mode](#) for details):

- **Very complete entry assistance** — real-time conflicts, room filters, consolidated availability...
- **Consolidated availability in real time** — changes in instructors' availability are immediately reflected as potential or actual conflicts.
- **Binary publication** — the timetable is either published or not, with no choice of ranges.

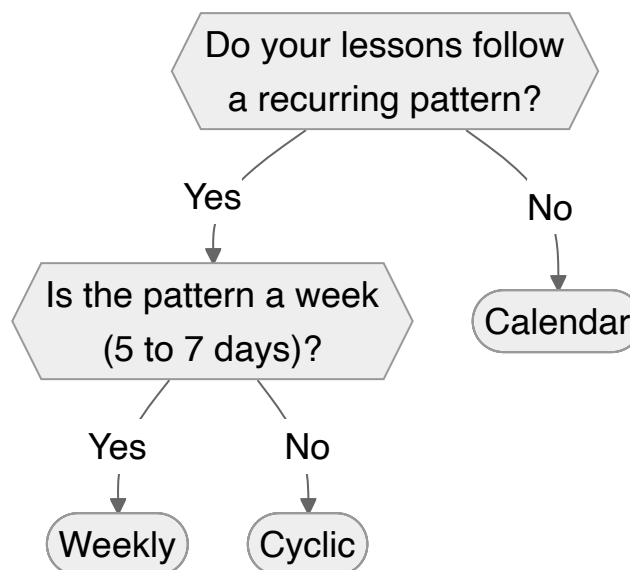
Current limitation — **semi-recurrence**: some schools have weeks that are broadly similar but punctuated with irregularities (lessons that end before the others, public holidays, occasional instructors...). Buttons let you **manually duplicate** a lesson over a recurring pattern, but the solver does not yet handle this case; automatic generation remains fully available on calendars with no underlying recurring pattern, or conversely, for fully recurring (weekly) timetables.

Partial generation — automatic generation can be restricted to a subset of the scope through the options panel. Three filters are available:


- **Classes**: selecting the classes to generate.
- **Subjects**: selecting the subjects to generate.
- **Date range**: specific to the Calendar mode. Lets you handle one period at a time (for example, the next term).

This is particularly useful in Calendar mode: you can build the timetable **through successive iterations** — one term, then the next; one program, then the others; and so on.

Choosing your mode: a quick guide



Converting a timetable to another type

If you want to switch an existing timetable from one type to another, you can **duplicate it while converting it** using the [Duplicate](#)  action in the timetable list, in the Timetable management module. All six conversions are supported: Weekly ↔ Cyclic, Weekly ↔ Calendar, Cyclic ↔ Calendar.

Depending on the target type, an additional parameter is requested:

- **Calendar**: the date range (start / end) over which to roll out the lessons. The dialog suggests by default the first and last weeks of the school year.
- **Cyclic**: the cycle length, in days (12 by default).
- **Weekly**: no additional parameter.

What gets remapped during the conversion:

- **Lesson positions** — each position is translated from the source system to the target system (weekday ↔ cycle number ↔ date).
- **Weekly → Calendar**: each lesson is multiplied across all matching dates of the target range, and the **subjects' hourly volumes** are multiplied by the number of weeks to preserve the total. The **A/B alternation** is applied automatically to each materialized date.
- **Time availability**: kept and migrated only from a **Weekly** source. A Cyclic or Calendar source drops it — dated or cycle-numbered availability does not reproject cleanly onto a model week or another cycle.

The result is a **standalone timetable**, independent of the original — you can rework it separately, publish it alone or alongside the original version.

Combining modes

You can **publish several timetables of different types simultaneously** over the same weeks.

Typical cases:

- **Recurring common core + one-off events**: a weekly timetable for the regular lessons + a **calendar** timetable for masterclasses, conferences or dated exams.
- **Different rhythms for the same classes**: a weekly timetable for the mornings (very regular core subjects) + a calendar timetable for the afternoons (sports, clubs, workshops that change from week to week).
- **Different operation depending on the class**: a weekly integrated preparatory cycle, vs a non-recurring graduate cycle with many external instructors teaching the same course to the same students several days in a row.

If several published timetables share a teacher or a room, Omniscol **dynamically merges the views** on the consultation side. When editing a timetable while another timetable is already published, Omniscol takes into account, by default, the teachers, rooms and classes occupied by that published timetable (this feature can be disabled).

See also

[Weekly](#)[Cyclic](#)[Calendar](#)[Overview of the Timetable management module](#)[Multiple active timetables in parallel](#)

1.7 First login

Source: [help/en/getting-started/first-login.md](#) · id: [getting-started.first-login](#) · Updated: 2026-05-10

To sign in to Omniscol, go to your school's URL — usually of the form `https://<your-school-name>.omniscol.com`. The login page asks you for an identifier and a password.

Credentials

Your credentials were given to you by your school. The default syntax is `firstname.lastname` or `lastname.firstname` (depending on the school's configuration in the [General settings](#)), but it can also be a numeric registration number.

Forgotten password

Click [Password forgotten?](#) Enter your login identifier. If an email address is on file, you will receive a reset link. Follow the instructions.

If you did not receive an email:

- check that the identifier you entered matches the one defined in Omniscol (when in doubt, contact your school's administrator),
- look in your junk/spam folder,
- check that your school has entered your email address in the user record of your account.

First login

If this is your first login, the system may immediately ask you to **change your password**. Choose a strong, unique one (Omniscol has no way to recover it — it is made unreadable before it is even transmitted).

Signing in via an identity provider

If your school uses [OIDC / SSO](#) (Google Workspace, Azure AD/Microsoft Entra, Keycloak, an internal SSO provider), a dedicated button appears on the login page. Click it and authenticate through your usual identity service.

Profile-based restrictions

The administrator can decide to **block login** for certain profiles (for example, disabling student login during the school year, or disabling a specific teacher account). If your profile is restricted, you see a message telling you so. See Application login restriction.

See also

[Set up the school account](#)

[Inviting and activating your users](#)

[OIDC / SSO](#)

1.8 Set up the school account

Source: [help/en/getting-started/setup-school.md](#) · id: [getting-started.setup-school](#) · Audience: admin · Updated: 2026-05-10

When your Omniscol account is created, the home screen guides you step by step. Follow the suggested order: it is the smoothest path. This page recaps the sequence from **right to left** in the Administration menu (the less often an entry is used, the further right it sits).

Step 1 — Settings

The Settings module contains the entire account configuration. Many values are pre-filled according to your school's **country**, based on the local practices observed by Omniscol — you usually have only a few things to adjust.

Check these first:

- School name — displayed on the login page.
- **Class levels** (Grade 6, Grade 7, Year 1, Year 2...) — they serve as a reference everywhere. Adapt them to your own nomenclature.
- **Teacher availability entry mode** — weekly, calendar (on Premium accounts), calendar + weekly (on Premium accounts), or disabled.
- **Timetable visibility restriction for students** — how many weeks in advance students can see their timetable (useful for delaying the release of a timetable that is not yet final).
- **Login identifier** — the format used to generate logins automatically (`prenom.nom`, `nom.prenom`, registration number...).
- First day of the week — important for countries where the week starts on Sunday (Arab countries, Israel).
- Alternate weeks (for weekly timetables) — A/B, 1/2 (that is, a display mode using letters or numbers), or disabled.
- **School logo** — displayed instead of the Omniscol logo on the login page and in the top banner.

For details, see [General settings](#).

Step 2 — School year

Before you can publish a timetable, you must have at least **one school year** declared in School years.

Fill in:

- **Name** — by convention, the dates: "2025-2026".
- **Start and end dates**.
- **Holidays** — either by copy-pasting from a spreadsheet, by importing the pre-filled dates for your country (offered at creation), or manually.
- **Current year** — designate the year currently in use. It is the one shown by default and the only one published.

See [School year and holidays](#).

Step 3 — Subjects

If Omniscol already pre-fills a **base of common subjects** for your country, you may not need to do anything. Otherwise, or for the disciplines specific to your school:

- Create **custom subjects** in [+ Create](#).
- Define the **course types** (for example lesson, workshop, exam, lecture) in [+ Create](#) if your teaching nomenclature uses them.
- Optional: configure **subject families** to group related subjects.

⚠ **Mind the spelling** of custom subjects: Omniscol makes an internal copy when a subject is assigned to a timetable, and a later correction does not propagate to timetables already configured in the past (only to current, future and unpublished timetables).

Step 4 — Users

This is where you create teachers, students and administrators. See [Inviting and activating your users](#) for the details.

At the end of this step, you will be able to create your first timetable. Then follow [Overview of the Timetable management module](#).

See also

[General settings](#)[School year](#)[Managing subjects](#)[Inviting and activating your users](#)[Preparing your data for a mass import](#)

1.9 Inviting and activating your users

Source: [help/en/getting-started/inviting-users.md](#) · id: [getting-started.inviting-users](#) · Audience: admin · Updated: 2026-06-13


Four school-side user roles structure the onboarding: **Administrator**, **Teacher**, **Student** and **Staff**. Each role gives access to a different scope.


Which screen for which role

Role	Screen	Required?
Administrator	Administrators	At least the main administrator.
Teacher	Teachers	Essential, so that lessons can be created and published with the teachers.
Student	Students	Optional: Omniscol can operate without knowing the students and without giving them access to the viewing portal.
Staff	Staff	Useful only if the Staffing module is used.

The detailed rights of each role and their technical mapping are described in [Users and roles](#).

Single creation or mass (batch) creation

Single: the  Add button on the screen of the role concerned. You open the form for a single user and fill in their basic information.


Mass (batch): the  Import data button on the screen of the role concerned. You create or edit several rows in a table, from a file prepared in Excel, Google Sheets, Numbers or Calc. See [Preparing your data for a mass import](#).

Activating a user


An active user can log in, appear on the screens that concern them and receive assignments.

To make a user inactive without deleting them, use their record or the mass action. An inactive account is blocked from logging in and from being assigned to new timetables, but its history remains available.

Inviting by email



The  Invite user button, on a selection of users, sends each recipient an email with a first-login link. First check that the email addresses are filled in; otherwise the invitation cannot be sent.

Setting initial passwords

If the email invitation is not suitable, you can set passwords manually with  Change password. Best reserved for pilot accounts, service accounts or situations where the school itself handles distributing the credentials. A password set manually this way (possibly with the help of the random generation button) is actually temporary: at their first login, the user will have to enter a new, personal one.

Placing students in classes and groups

Placements are managed from the Students screen:

1. On Students, select the students concerned.
2. Click  Assign to a class.
3. Choose the school year, the class and the dates if necessary.
4. Use  Groups for groups (languages, electives, practicals, half-classes).

Placements can be done in several waves, for example by level or as enrollments come in.

There is an option for students to be assigned directly to classes and groups at the timetable level, upfront, rather than in the Administration module, after the fact. Contact the Omniscol team if you are interested in this way of working.

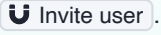
OPTION: CUSTOM ROLES

Custom roles

The **Custom roles** option restricts an administrator account's rights module by module and operation by operation. See [Custom roles](#).

☰ How-to

Setting up users at the start of the school year


1. Prepare the user files or tables: administrators, teachers, students and staff if needed.
2. Import or create the accounts from the Administration module screens.
3. Check the emails, the logins and the roles.
4. Send the invitations with .
5. Place the students in their classes and groups once a first timetable is published on the school year.
6. At the end of the year, deactivate the accounts that should no longer log in, rather than immediately deleting the history.

🔗 See also






[Set up the school account](#)[Preparing your data for a mass import](#)[Users and roles](#)[Managing teachers](#)[Managing students](#)

1.10 Preparing your data for a mass (batch) import

Source: <help/en/getting-started/preparing-data.md> · id: [getting-started.preparing-data](#) · Audience: admin · Updated: 2026-06-03

Manually entering several hundred or thousand rows (teachers, students, rooms, lessons...) is rarely wise. Omniscol offers **mass imports** by copy-paste from any spreadsheet (Excel, Google Sheets, Numbers, Calc...), in CSV or TSV format. These imports are performed by clicking the  buttons, usually at the top of entity listings. A few of them only export data, but most accept imports.

What can be imported?

Entity	Screen	Typical volume
Teachers	 Export data in CSV format	10 - 1000
Students	 Export data in CSV format	50 - 10,000
Rooms	 Export data in CSV format	10 - 500
Custom subjects	 Export data in CSV format	10 - 1000
Lessons of a timetable	 Export data in CSV format	100 - 10,000

Preparing your files

Golden rules:

1. **One row = one entity.** No merged rows, no multi-line headers.
2. **Follow the column order expected by Omniscol.** The import screen shows the template with the columns in the target order, and you can **reorder these columns** on the Omniscol side to match the order of your source spreadsheet — which makes the copy-paste from Excel or Google Sheets clean, without rewriting your file.
3. **Some columns are read-only.** They are then greyed out, intended only for exporting the data. The first column is usually the identifier of the data already stored in the database.
4. **Columns marked with an "*" are required.** If a required field is missing, the row is ignored (not imported at creation, or deleted if it already existed in the database).
5. **Separators inside a cell are free** in lists (multi-teacher, multi-group, multi-room lessons): comma, semicolon, slash.

6. **The built-in AI will clean the data**, but only up to a point: try to provide clean, unambiguous data, otherwise you may be in for a few unpleasant surprises.

Typical fields per entity

Teachers

- First name, Last name (required).
- Email (recommended for invitations).
- Registration number / external ID (optional but useful for traceability).
- Subjects taught (list, free separators).
- Service hours (planned number of weekly hours).
- Main site (if multi-site).

Students

- First name, Last name (required).
- Email (recommended).
- SIS registration number (useful to avoid mixing up namesakes).
- Level / Class.
- Groups (elective, subgroup).
- Status (active, pre-enrolled, inactive).

Rooms

- Name (required).
- Site (required if multi-site — otherwise Omniscol assigns the default site).
- Capacity.
- Specialisations (list).
- Building (free text).
- Tags / comments.

Lessons

- Class + subject (at minimum).
- Duration (otherwise deduced from the start/end times).
- Day, start time, end time (to pre-position; otherwise leave empty and position later).
- Group (optional).
- Lesson type (tutorial, practical, lecture, exam...).
- Teacher(s) (free separators).
- Room(s) (free separators — yes, [multiple rooms](#) on the same lesson are supported).
- Resource(s).
- Alternating weeks (format `A/B`, `1/2`, etc.).
- Free-text comment.

After preparation

The import flow depends on the target screen. Generally, the screen fully refreshes with the new data.

For the **lessons of a timetable**, see [Mass import of courses from a spreadsheet](#), which documents the dedicated four-step wizard.

Exporting

The same built-in spreadsheet mechanism extracts the data. Several options for this:

- Copy-paste to an external spreadsheet (TSV format, automatic).
- Use the buttons at the bottom to export directly to a PDF, CSV or Excel (XLSX) file.

Re-importing

Importing is not limited to an initial import. It is possible to mass import in several passes. The first column, labeled "id", holds each entity's read-only identifier in the database, which can also be found in the URLs. If you edit a row matching a given identifier, the entity's data is updated.

To add new data, just go to the very bottom of the spreadsheet, where there are always a few empty rows with no identifier attached. Pasting new data, as in the initial import, grows the table accordingly.

Re-imported entities may carry complex data that is not eligible for import/export (sometimes a read-only column exists solely to export that data). This is for example the case for time constraints on teachers, rooms, classes, etc. In that case, the identical re-injection is automatic.

This system thus lets you mass import in several passes without losing enriched data on existing entities, while still being able to create new ones or delete others.

Mass deletion

As seen for re-importing, the first "id" column identifies the different entities. If you clear a row — at least the required columns marked with an "*" — the import will delete the emptied entities. Since you can select several rows and columns, as in a standard spreadsheet, just press the Delete key to clear as many rows as you want, and thus delete the corresponding entities from the database, in bulk.

See also

[Import and export](#)[Mass import of a timetable's lessons](#)[Migrating from another software](#)

1.11 Getting-started guided tour (the 6 steps of the Home module)

Source: [help/en/getting-started/onboarding-tour.md](#) · id: [getting-started.onboarding-tour](#) · Audience: *admin* · Updated: 2026-05-10

When you log in for the first time on a blank account, the **Home** module offers a **six-step checklist** — each step ticked automatically when its precondition is met (green ✓) or marked as blocking (red ✗) while nothing has been entered yet.

The steps follow a strongly recommended logical order, but you can always skip a step and come back to it later. At one point or another, a missing piece of data will force you to retrace your steps.

This help page walks through the same sequence, step by step; a **clickable guided tour** recaps it at the end of the page.

Overview — welcome message


Welcome to Omniscol, the online timetable generation and management software! If you see this message, this is your first login or you have not yet created your first timetable. Follow the steps below to get started with the software. The order shown is the recommended one, but you can go back to a previous step at any time.

Organization tip: the Home module hides itself automatically as soon as your **first timetable is published**. While it is visible, it is your compass. The [display the configuration steps](#) link brings it back if needed after publication:

Your account has been configured and is now functional. You can still display the configuration steps.

Step 1 – Enter the users

First add all users: teachers and students. Adding students is optional, Later on, you can create timetables by specifying only the number of students in each class and group. You can also add users later, and only then assign them to classes and groups.

Hint: you can add large amounts of data by pasting from Excel (or other spreadsheet software) by clicking the  icon.

Action:

- Input teachers (*required* — red cross while no teacher has been created)
- Input students (*optional* — no red cross, just a circle if not filled in)

See [Inviting and activating your users](#) for the details (single creation vs mass import, email invitations, initial passwords, mass operations).

Step 2 – Fill in the settings

Please, fill your account settings, including class grades.

Action: Specify settings

See [General settings](#). The first things to adapt: class levels, availability mode, first day of the week (for countries whose week starts on Sunday), identifier syntax. Most of the other values are pre-filled for your country.

Step 3 – Create a first school year

A school year is a period between start and end dates, including holidays, during which timetables will be published. It is required to access the daily-use modules (timetable viewing, dashboards, absence management, etc.).

Action: Create your first school year

See [School year and holidays](#). Give it a name ("2025-2026"), start and end dates, and import the pre-filled holidays for your country if offered.

This is the step that unlocks access to publication in the timetable management module.

Step 4 – Create the custom subjects (optional)

Define specific subjects on the curriculum that are not included in the default subject database. This step is optional.

Action: Create custom subjects

See [Managing subjects](#). The database of common subjects per country is already pre-filled; here you only create the subjects **specific to your school** that are not in the database.


Step 5 – Create (then configure) a timetable

Step 5 has two variants depending on the account's state:

5a – While no timetable exists yet

Create a new timetable. It is recommended to proceed step by step, filling in each tab from left to right. Once finished, click the generation button.

Action: Create a timetable

Click  [Create timetable](#) in the [Timetable management](#) module. If your account offers several modes, choose the [timetable mode](#) (weekly, cyclic, calendar).

5b — Once the timetable is created, while it is being configured

A timetable is being configured but has not yet been generated. We advise you to complete the setup step by step by filling in each tab from left to right. Once finished, click the generation button.

Action: Resume timetable configuration

Work through the timetable's tabs **from left to right**: [General](#) → [Sites](#) → [Teachers](#) → [Classes](#) → [Alignments](#) → [Hours distribution](#) → [Generation](#).

Step 6 — Distribute (publish) the timetable

Once the timetable has been configured and generated, it must be allocated to some or all weeks of the school year. Click the allocation button in the management screen. This screen also allows you to find timetables being configured or already generated.

Action: Manage timetables

See [Publishing \(activating\) a timetable](#). This is the step most often forgotten: until the timetable is distributed over the weeks of the school year, it remains a draft invisible to end users. Publication is required for the everyday-use modules (viewing, dashboards, absences) to be fully operational. It also activates any real-time exports to the outside of the application: iCal, connected external software, etc.

What comes next

Once the checklist is green from end to end, with a timetable fully configured and published:

- **Assign the students** to their classes / groups (see [Inviting and activating your users](#)).
- **Invite the users** by email ([Invite user](#)).
- **Set up the display panels** on the premises (see [Display panels](#)).
- **Set up the iCal subscriptions** of teachers and students (see [iCal](#)).
- **Test student access** from a test account.
- **Configure the integrations** (school-life software, ERP...) if applicable (see [Integrations overview](#)).

See also

[A five-minute guided tour](#)

[First login](#)

[Set up the school account](#)

[Inviting and activating your users](#)

[Overview of the Timetable management module](#)

[Publishing \(activating\) a timetable](#)

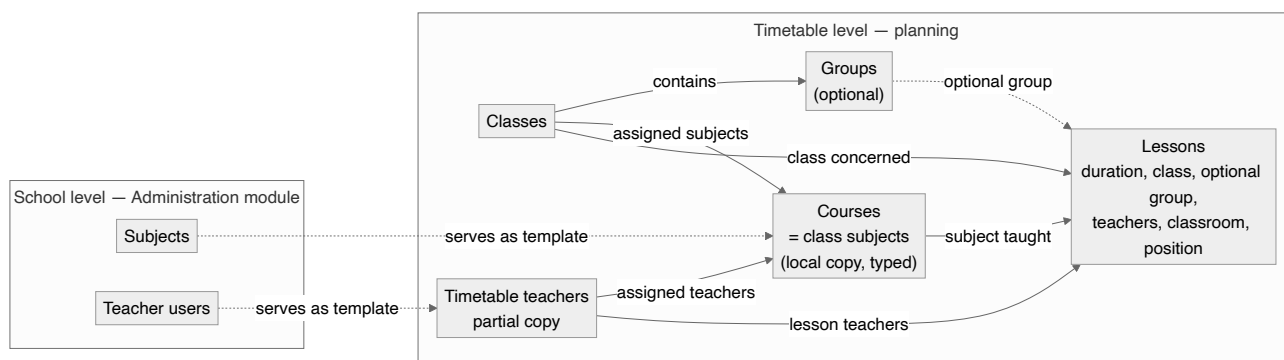
2. Core concepts

2.1 Data organization: subjects, teachers, classes, timetables

Source: [help/en/core-concepts/data-model.md](https://help.en/core-concepts/data-model.md) · id: [core-concepts.data-model](#) · Audience: [admin](#) · Updated: 2026-06-13

Omniscop distinguishes two levels of organization: a **school repository** carried by the Administration module (subjects, users, school years...), which describes your institution over the long term, and **timetables** that draw on its elements to build a consistent plan over a given period. Understanding this duality is the key to handling changes calmly (renaming a subject, updating a teacher, archiving a timetable) without breaking the history.

Overview: two levels, two logics



School level: a global repository with a long lifespan, managed in the Administration module. It contains what is true for the institution independently of any particular timetable — the calendar of school years, the subject catalog (subjects common to the country plus the school's custom ones), and the directory of users (with their roles).

Timetable level: a consistent plan over a period. Each timetable has its own list of teachers, classes, groups, subjects attached to classes, and lessons. These objects are not mere references to the school repository: they are **local copies**, which can be enriched without modifying the repository. Conversely, the school-level repository can evolve (subjects or teachers being deleted) without breaking past timetables.

The rest of this page details each type of link.

The school year: the time frame

The **school year** defines a `start date` → `end date` range and the list of holidays (see [School year](#)). It does not directly "contain" timetables: it is the timetables that **unfold** over weeks (weekly / cycle mode) or dates (calendar mode) belonging to a school year.

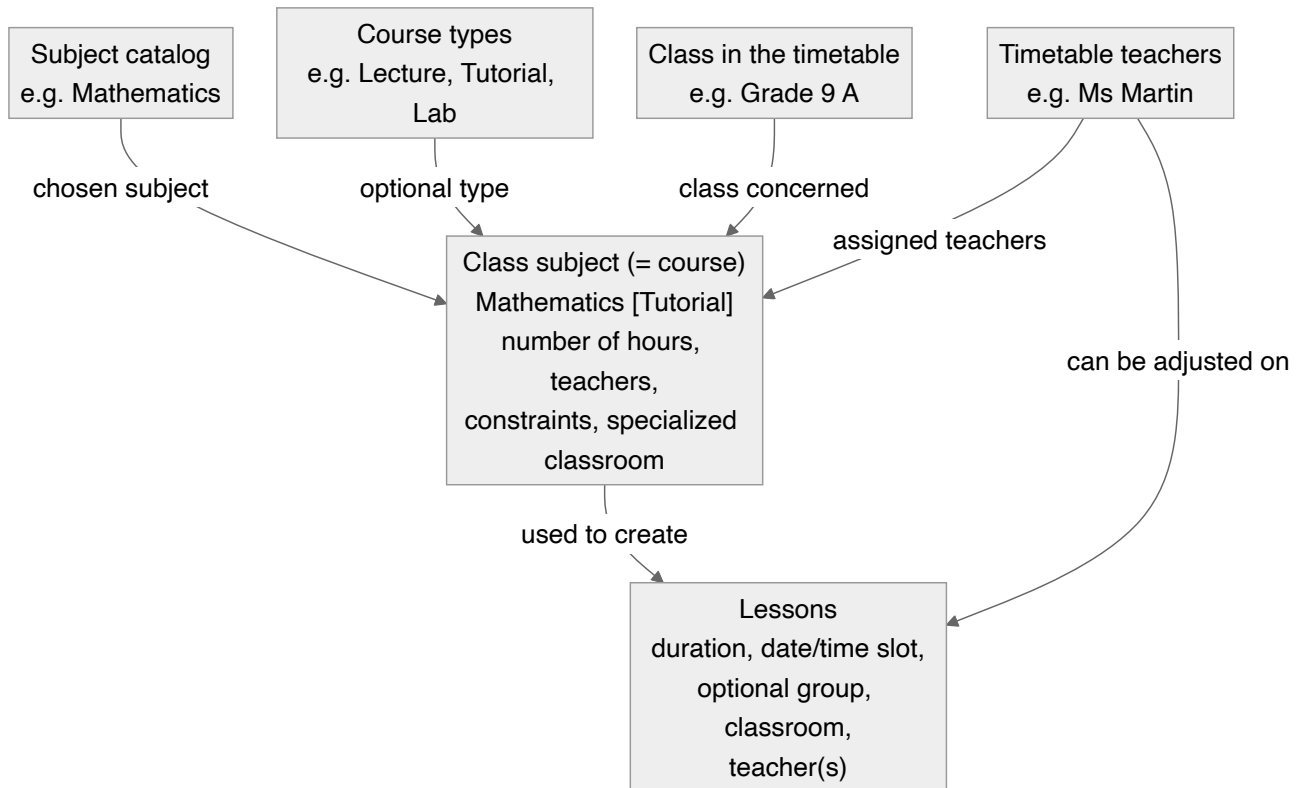
This allows several configurations:

- **A single timetable per school year** (the standard case): the timetable covers all the working weeks of the year.
- **Several successive timetables** within the same year: one timetable per term for example, each handing over to the next at the changeover dates.
- **Several timetables in parallel** over the same period (feature included in Premium; on some Standard accounts it can be enabled under a suitable contractual arrangement), for example:
 - a timetable for the recurring core curriculum
 - a calendar timetable for masterclasses, merged when viewed.

See [Multiple active timetables in parallel](#).

The school year therefore remains a frame — not a container. Changing its start and end dates or its holidays updates the calendar used by the timetables attached to it, without modifying their content.

Subjects: from the school catalog to a class's lessons



Level 1 — The catalog in the Administration module

At school level, two origins coexist:

- **Common subjects:** Omniscol's reference set for the configured country (official codes, standardized labels). This catalog is read-only for the school.
- **Custom subjects:** what the school creates itself, managed from [the Subjects screen](#). Each custom subject has a name, a (short) code, and optionally: a short name, a specific color, a parent subject, a family.

This catalog says nothing about which classes teach these subjects — it only says that they exist.

Subject families can also come from a shared per-country reference set, read-only, or from a custom reference set local to the school.

Level 2 — Assignment to a class (with or without a type)

When you **assign a subject to a class** in a timetable, a **local copy** of the subject is created in the class, enriched with planning-specific fields (target number of hours, pedagogical weight, incompatibilities, default teachers, classroom specialization...).

If you assign the subject **with a course type** (Lecture, Tutorial, Lab, Exam — see [Types of course](#)), each type creates a **separate entry** in the class:

Subject in the class	Origin	Type
Mathematics	Mathematics	none
Mathematics [Lecture]	Mathematics	Lecture
Mathematics [Tutorial]	Mathematics	Tutorial

This is intentional: each variant (Mathematics lecture, Mathematics tutorial) becomes a separate local copy, with its own number of hours, its own teachers and its own incompatibilities. The link with the catalog is not a hard

dependency: the local copy keeps the subject's origin as long as the subject still exists in the catalog. If the subject is deleted from the catalog, the local copies in the timetables remain valid.

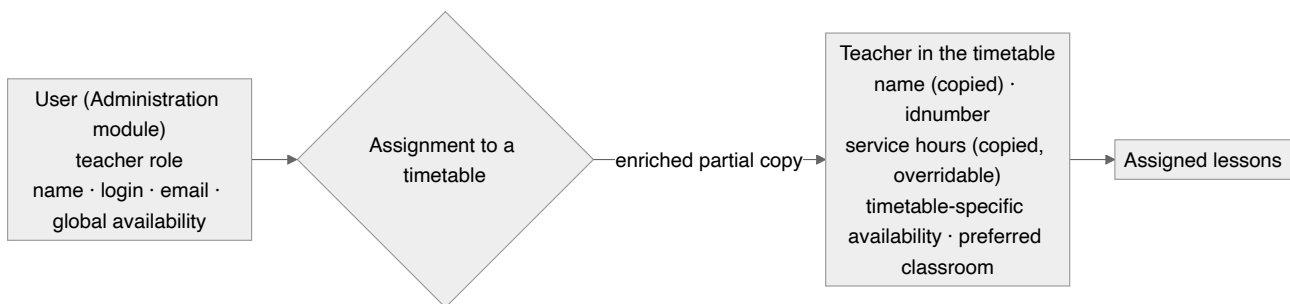
Level 3 – From course settings to lessons

In Omniscol, what is commonly called a **course** generally corresponds to a subject assigned to a class, possibly associated with a course type (Lecture, Tutorial, Lab...), then enriched with planning information: number of hours, teachers, groups, placement constraints, classrooms or resources.

This definition states what needs to be scheduled. **Lessons** are the occurrences actually placed in the timetable, on a given week or date.

Example: a class subject such as **Mathematics [Tutorial]** can define a number of hours, teachers, groups, placement constraints and classroom preferences. Lessons are then created from this definition. See [Courses, lessons, course types](#).

Teachers: school repository → enriched timetable copy



The teacher at school level

At school level, a teacher is a **user** holding the `teacher` role (the same user can hold several roles — see [Users and roles](#)). Managed from [the Teachers screen](#), this user record carries the durable identifying information: last name, first name, login, email, identification number, as well as the teacher's **global availability** (recurring) and reference service hours.

The teacher assigned to a timetable

When you assign this teacher to a timetable, Omniscol creates an **enriched partial copy** in the timetable. The copy carries over only a selection of fields from the school repository and adds planning-specific fields to them:

Field	School level (repository)	Timetable level (copy)
Identifier	the user's identifier	identical (implicit link)
Last name, first name, middle name	source	copied at assignment
Identification number	source	copied at assignment
Email, phone, login	source	not copied (remains only in the Administration module)
Reference service hours	source	copied, overridable per timetable
Preferred classroom	not managed in the repository	timetable-specific
Availability	the user's "global availability"	recurring mode: local copy, validated or relaxed; calendar mode: shared source, viewable and editable from Administration or from the timetables concerned

This duality matters: the **same teacher** can have **different service hours** depending on the scope being planned, or a **different preferred classroom** at a given time (for example because of a temporary disability).

For availability, the logic depends on the planning mode:

- in recurring mode, the availability entered by the teacher goes through an acceptance phase by the administration. It can be adapted locally in the timetable, for example by turning a strict unavailability into an undesired time slot;
- in calendar mode, the teacher's availability is treated as a single source of truth, in particular for external instructors. It can be viewed and edited from the user's record or from any calendar timetable the teacher is assigned to.

In both cases, some settings remain timetable-specific and can locally enrich this availability.

Virtual teachers

A variant: you can create a **virtual teacher** in a timetable — a position to be filled, with no real user behind it ("the future math teacher"). It has no counterpart at school level. At recruitment time, you can replace this virtual teacher with a real teacher user (see [Assigning teachers to a timetable](#)).

The local-copy principle, in summary

The same logic governs subjects in a class and teachers in a timetable: **a local copy is created at assignment time**, and it is this local copy that carries the timetable-specific fields.

Why this design choice:

- **Temporal independence.** A timetable closed last year must not change when you rename a subject or a teacher this year. Local copies guarantee **historical integrity**: what was planned stays exactly as it was at the time.
- **Contextual enrichment.** A teacher does not have the same availability or the same service hours depending on the scope being planned; a subject does not have the same weight or the same incompatibilities depending on the class teaching it. The local copy is the natural place to carry these variations.
- **Cleanup without breakage.** Deleting or overhauling a subject at school level does not erase the timetables that used it — their local copies remain valid.

As a trade-off, school-side renames do not propagate silently everywhere. That is the subject of the next section.

Practical consequences: renames, deletions, history

When you modify an entity at school level, Omniscol applies a clear propagation rule:

Action at school level	Past (closed) timetables	Current / future timetables
Rename a custom subject (name, short code, code)	Unchanged — history preserved	Name propagated to the matching local copies
Change a custom subject's color	Unchanged	Color updated if the previous color had not been overridden locally
Rename a teacher (last name, first name)	Unchanged	Name propagated to the local copies
Change a teacher's email or phone	No effect (not copied)	No effect (not copied)
Change a teacher's global availability	No effect	Recurring mode: update indicator if the availability had already been accepted; calendar mode: the shared availability source is updated
Delete a custom subject	Local copies preserved	Local copies preserved; the subject only disappears from the catalog
Disable a user's teacher role	Unchanged	The copies in the timetables stay in place

The distinction between "past" and "current or future" is made against **today's date**: a published timetable whose last active week is before today is considered past and is no longer modified by renames.

When to propagate manually

Automatic propagation stays **deliberately narrow**: subject name, code and color; teacher name and identifier. The name and code of a locally copied subject cannot be edited in the timetable; only its color can be redefined locally. The other fields (incompatibilities, pedagogical weight, service hours, per-timetable availability...) are not synchronized, because they are **by nature timetable-specific**.

If a deeper change must be applied to several existing timetables (for example: revising a teacher's service hours on all active timetables, or adding a new subject incompatibility everywhere), you have to act timetable by timetable. For bulk operations, the fastest tool is the **copy-paste import from a spreadsheet** on the relevant screen of each timetable.

See also

[School year](#)

[Class, group, subgroup](#)

[Course \(glossary definition\)](#)

[Courses, lessons, course types](#)

[Managing subjects](#)

[Managing teachers](#)

[Types of course](#)

[Overview of the Timetable management module](#)

2.2 Class, group, subgroup

Source: [help/en/core-concepts/classes-and-groups.md](https://help.en/core-concepts/classes-and-groups.md) · id: [core-concepts.classes-and-groups](#) · Updated: 2026-05-20

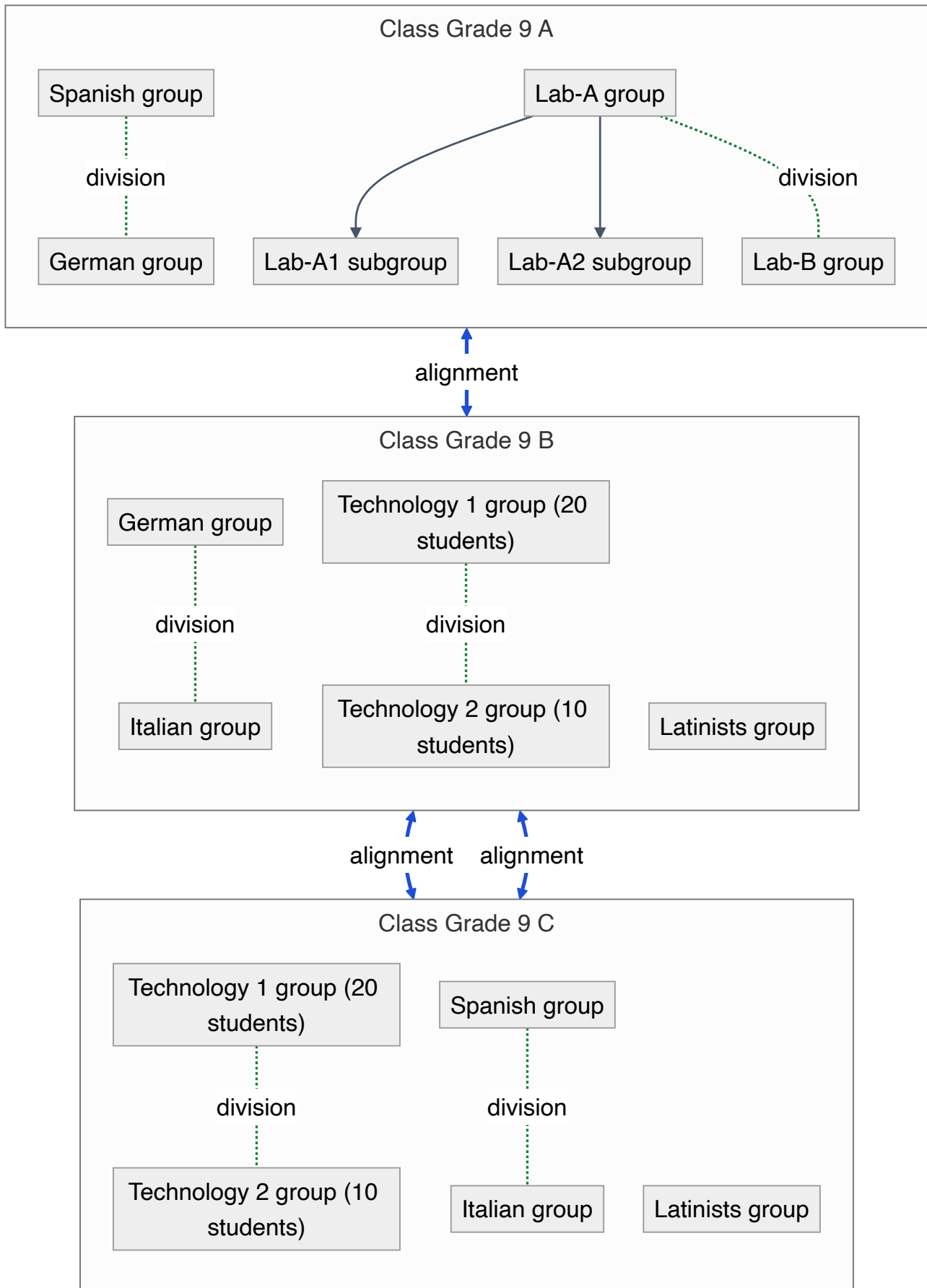
Classes represent separate sets of students who share the same curriculum and usually attend their courses together (same subjects, teachers and classrooms), with a single exception: division into groups, that is, custom subsets with a marginally adapted curriculum (subdivision by options, ability levels, gender, alphabetical order...). Two different classes never have students in common.

In Omniscol, you always start from the **class**, then, inside each one, you create **groups** whenever the whole class does not follow the same course. **Subgroups** refine this split when a group must itself be subdivided: a group

attached to another group, in a parent-child relationship, becomes a subgroup.

Class divisions, alignments and **groups of groups** are not extra levels in the hierarchy. They are ways of organizing or linking groups depending on the use case.

Overview



Read it as follows:

- a **class** contains groups;
- a **group** can contain **subgroups**;
- a **class division** organizes groups of the same class that share out the students;
- an **alignment** links groups from different classes for a shared course;
- a **group of groups** builds an explicit, reusable grouping, especially when the grouping must evolve.

This diagram shows several classes, with several groups, some in a division, others aligned, sometimes both. The second foreign language is an exclusive option: a student attends either the German, the Spanish or the Italian course. That is why the different language options are declared as a division: they can be scheduled simultaneously (provided different teachers teach each of the languages). Few students take German in Grade 9 A and Grade 9 B: they are therefore brought together to follow the same courses — that is the purpose of the alignment; mirrored lessons are created in each class, so they are scheduled simultaneously. The same goes for the Latinists of Grade 9 B and Grade 9 C.

This diagram also illustrates a classic resource-saving pattern, "**3 groups across 2 classes**": classes Grade 9 B and Grade 9 C are each divided into Technology 1 (20 students) and Technology 2 (10 students). By **aligning** the two Technology 2 groups, the school merges the two small groups and runs only **three technology courses** instead of four — all of equivalent size (20 students), with one fewer teacher and one fewer room needed.

Class: the base level

A **class** is the default reference entity. Depending on the context, it can represent:

- a school class,
- a year group,
- a cohort,
- a session,
- a track.

When a course concerns **the whole class**, no extra group is needed: the course is simply attached to the class.

Group: a subdivision of a class

A **group** is always a **subdivision of a single class**. All its students remain students of the parent class.

Classic examples:

- `Group A / Group B` for an alphabetical split;
- `Lab-A / Lab-B` for lab-session half-groups;
- `Spanish / German / Italian` for exclusive second-language options;
- `Advanced English / Intermediate English / Basic English` if the school works with ability groups;
- `Boys / Girls` in the contexts where this split exists;
- `Elective Marketing, Elective Finance, Elective Data` in a higher-education elective set.

Strong recommendation: create one group per clear pedagogical use, even if the students are sometimes the same. For example, `Latinists` and `Hellenists` are better than reusing a generic `options` group. Otherwise, timetable reading, diagnostics and cross-class groupings quickly become ambiguous.

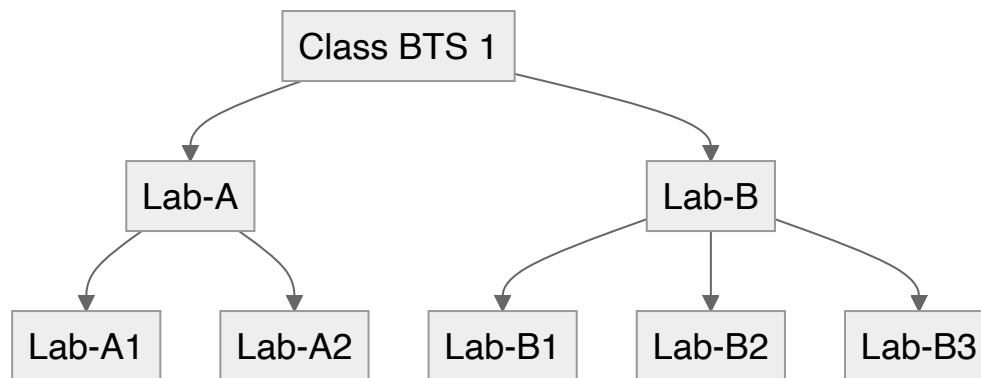
PREMIUM

Subgroup: a subdivision of a group

A **subgroup** is a child group attached to a parent group.

Example:

- the class is split into `Lab-A` and `Lab-B`;
- then `Lab-A` is itself split into `Lab-A1` and `Lab-A2` for finer rotations.



Important consequences:

- subgroups stay within the scope of the class;
- they inherit their parent's logic;
- they are useful when you have several levels of subdivision;
- they only make sense if the structure stays readable.

Subgroups are very handy when they are also used with class divisions:

- groups A and B in a division;
- groups A1 and A2 in a division;
- groups B1, B2 and B3 in a division.

Automatically, A1 and B3 are in a division, just like A2 and B1, and so on. This relationship is deduced from the combined logic of divisions and parent-child links.

Subgroups are groups that were dragged and dropped onto a parent group. The page [Group hierarchy](#) goes further into advanced uses: inherited divisions, inherited constraints and time masks.

Class division: splitting one class into exclusive groups

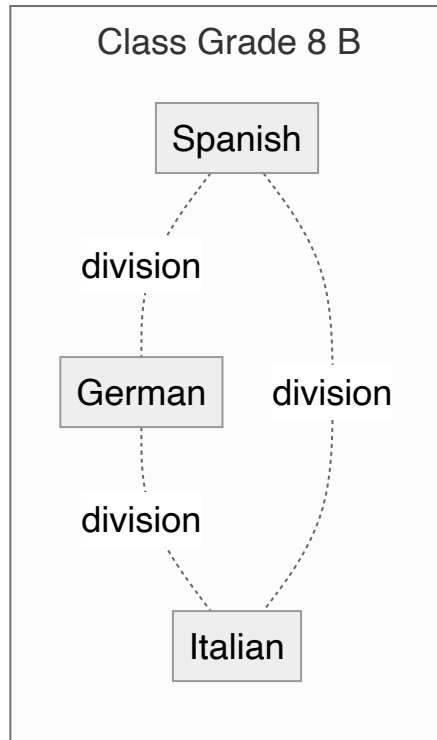
A **class division** is used between groups of **the same class** when the students are spread across several mutually exclusive groups. Omniscol is constraint-based scheduling software, so by default groups are assumed to potentially share at least one student: declaring a division between groups lifts that constraint. Explicitly, groups of the same division are not in conflict.

The idea to remember:

- every student must belong to **only one (or none)** of the division's groups;
- these groups can then carry different courses in parallel;
- Omniscol knows this is not a student conflict.

Examples:

- `Lab-A` and `Lab-B` run two different lab sessions at the same time;
- `Spanish`, `German` and `Italian` share the same option time slot;
- `Group A` and `Group B` alternate simultaneous activities in two rooms;
- `Advanced English`, `Intermediate English` and `Basic English` follow three different courses in parallel.



Within a single class, the division is therefore the right tool when you want to say: "these groups share out the students, they can be placed in parallel".

Note: Omniscol has an optional mode of operation where students are directly assigned to classes and groups at the timetable level. In that case, divisions can be deduced automatically (no student overlap between groups = division).

Subgroups and class divisions

With subgroups, the division logic propagates along the hierarchy. If `Lab-A` and `Lab-B` are in a division, the subgroups of `Lab-A` and those of `Lab-B` inherit this separation between branches.

Example:

- `Lab-A` and `Lab-B` are in a division;
- `Lab-A1` and `Lab-A2` are subgroups of `Lab-A`.

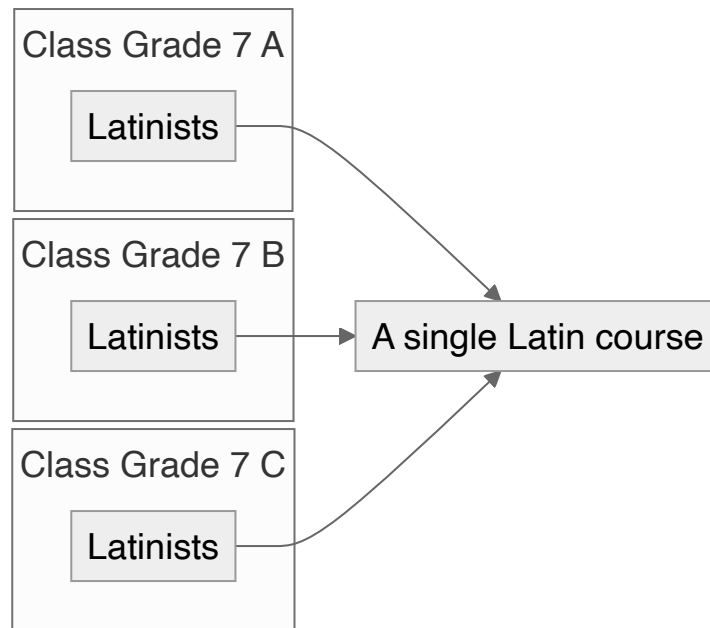
Omniscol then knows that a course on `Lab-A1` does not conflict with a course on `Lab-B`, without you having to redeclare every combination by hand.

However, if you want to subdivide `Lab-A` into several branches that also share out the students, you must explicitly define this subdivision in the hierarchy and, if needed, its own division.

Alignment: one course shared across several classes

An **alignment** is used when several groups from **different classes** actually follow **the same course**, in the **same time slot**, with the **same teacher** and in the **same classroom**.

Classic example: the `Latinists of Grade 7 A`, `Grade 7 B` and `Grade 7 C` share a single Latin course.



Alignment is very useful for cross-class options or electives, but it is more rigid than a simple group: everything that is aligned lives together.

Important condition: it must be the **same subject** across the classes involved.

In other words, if you want a shared course across several classes, those classes must actually share the subject used by that lesson.

Example:

- `Marketing` shared across several classes: yes;
- `Marketing B3` in one class and `Marketing M1 elective` in another: no, not if they are two distinct subjects in the timetable structure.

In higher education, this often leads to avoiding subjects too specific to a single program when you know courses will be shared. It is better to plan a subject common to the classes that must share lessons.

Alignments are better suited to secondary schools: the point is to create mirrored lessons, and to declare the dynamic grouping logic. The drawback is having to respect this mirror: you must create as many lessons as there are groups in the alignment, with the right groups assigned in each class, and everything must be symmetrical: teacher(s), classroom, resources, etc. If a lesson carrying a group of an alignment is positioned, a mirrored lesson must exist in each class, with its corresponding aligned group.

The advantage is the ability to create complex courses with alignments. You can have a shared Latin course on week A, and a different, non-grouped course in each class.

If these constraints are not offset by the benefits, consider groups of groups instead.

Group of groups: an explicit, editable grouping

A [group of groups](#) gathers several groups into a named, editable and reusable grouping. The group of groups is then used like any other group.

It is particularly useful:

- in higher education and continuing education;
- when the composition of a grouping may change;
- when you want to keep a readable, traceable structure.

Example: a shared seminar brings together students from **M1 Marketing**, **M1 Finance** and **Elective Data**. The lessons are scheduled with the group of groups, then, later, a fourth group joins the arrangement: simply add it to the group of groups and every lesson is attached to it.

As with alignments, the shared lessons must rely on a **common subject** across the classes involved. The group of groups brings audiences together; it does not replace the consistency of the subject used for the course.

If you need to bring together **whole classes**, first create in each class a group representing that whole class, then use these groups in the alignment or the group of groups. Classes are never aligned directly without going through groups.

Assigning several groups directly to a lesson

It is possible to assign **several groups directly to a lesson**, without first creating a named group of groups.

This is useful:

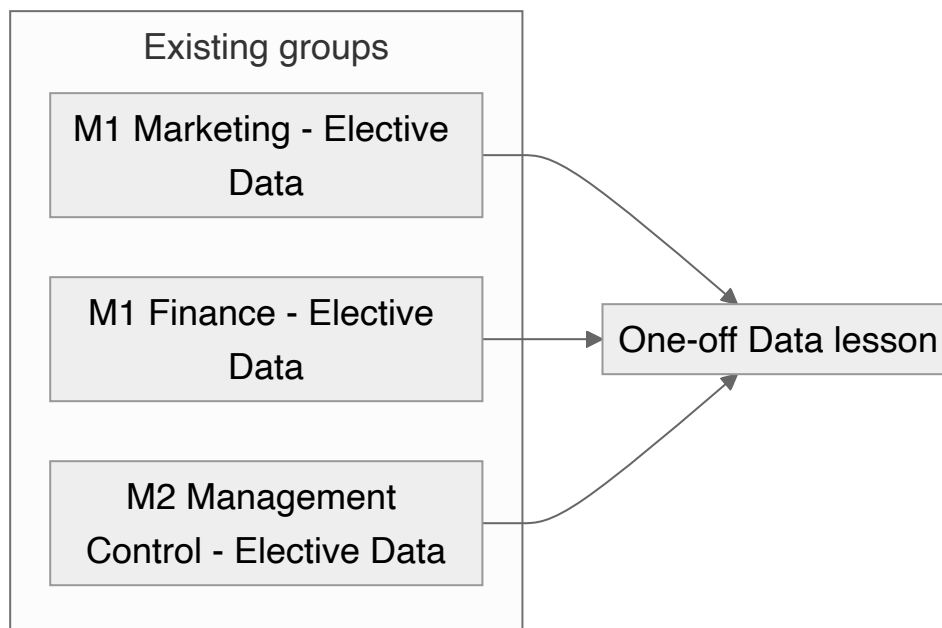
- for a one-off need;
- to quickly test a grouping;
- for an exceptional lesson that does not justify a dedicated structure.

But it is not the best choice as a long-term model:

- it is less readable over time;
- it is less traceable than a named grouping;
- it becomes harder to understand the pedagogical intent if this pattern repeats often.

In practice:

- for lasting use, prefer an explicit **group of groups**;
- for a one-off need, directly assigning several groups is very handy.



However, avoid mixing, on the same lesson, a **parent group** and one of its **subgroups**: this is generally not a clean model of the audience involved.

Summary table

Concept	Scope	What it is for	Example
Class	Reference set	Carry the courses everyone attends	Grade 6 A , BTS 1 , M1 Marketing
Group	Subdivision of a class	Isolate a subset of students	Lab-A , Latinists , Spanish
Subgroup	Subdivision of a group	Refine an already split organization	Lab-A1 , Lab-A2
Class division	Groups of the same class	Spread mutually exclusive students over parallel courses	Lab-A / Lab-B , German / Spanish
Alignment	Groups from different classes	Attend a single shared course	Latinists from several classes
Group of groups	Several groups, same or different classes	Build an explicit, editable grouping	Shared seminar across programs
Multiple groups	Several groups, same or different classes	Assign several groups dynamically, on the fly	One-off lesson for varied audiences
Free group	Open membership	Handle enrollments that are not fixed	Workshop, club, open activity

See also

[Class divisions](#)[Group alignments](#)[Groups of groups](#)[Group hierarchy](#)[Free groups](#)[Higher education use cases](#)

2.3 Class divisions

Source: <help/en/core-concepts/class-divisions.md> · id: [core-concepts.class-divisions](#) · Updated: 2026-05-20

A **class division** declares to the solver that several **groups** of the same **class** can share the **same time slot** without conflicting — because no student belongs to more than one of these groups.

Why it is necessary — the "everything conflicts by default" philosophy

Omniscol is **constraint-based** scheduling software. Direct consequence: **anything that can be a source of conflict is treated as one by default**, unless the user explicitly states otherwise.

- Between **two distinct classes**: no students in common by assumption, so no conflict to report. (Transverse / shared courses are modelled separately with **alignments** or **groups of groups**.)
- Within **one and the same class**, the solver has **no logical way** to guess which groups can coexist in the same time slot without overlapping. It therefore treats two groups of the same class as **potentially having students in common**. Without a declared division, scheduling two lessons in the same time slot for two groups of the same class raises a conflict.

It is up to the timetable author to **tell the solver**: "these groups are mutually exclusive, you may schedule them simultaneously". The class division is that declaration.

The solver will try as much as possible to position the lessons of groups in a division simultaneously, but it may not manage to because of other constraints (classroom shortage, incompatible time constraints between teachers or subjects, a shared entity such as a specialised classroom or a teacher common to the lessons). In that case, the solver will detect that part of the class has no lesson and apply a proportional penalty (default option: reduce gaps

for students), pushing it to schedule those lessons at the very start or very end of the day, or during the lunch break when possible.

Typical use cases

- **Alphabetical half-classes:** groups A and B (by alphabetical order or random draw) — a student is in one OR the other, never both.
- **Half-classes by level:** Advanced English vs Standard English groups; a student is in one OR the other.
- **Half-classes by sex** (a special case in some schools, for physical education for example).
- **Exclusive electives:** German / Spanish / Italian second-language options — a student chooses **only one** of the three languages.
- **Bundles of simultaneous electives:** a set of electives where the school decides to schedule all the options in the same blocked time slot (`Friday 2pm-4pm = electives slot`), and students choose their option from the bundle — all exclusive in the sense that "a student takes a single option".
- **Practicals / labs in half-groups:** Lab-A and Lab-B in the same time slot in two different classrooms with two teachers.

Creation

Steps:

1. Go to a class's groups page (the `Groups` tab).
2. Select the groups concerned.
3. Click `+ Add class division`.
4. Confirm.

Several divisions can coexist within the same class: one for science practicals (Lab-A, Lab-B), one for languages (German, Spanish), one for speciality electives (Philosophy, History-Geography, Mathematics). Each division is independent of the others.

When a group belongs to a division, it appears on all relevant screens with an  icon.

Default safeguard: one group in a single division

By default, the interface applies a simple safeguard: a group belongs to only one division. This is the most common case, and it prevents many mistakes.

This filter can however be lifted when the actual student mix justifies it.

Example:

- only the students in `German` also take `Latin`;
- the students in `Spanish` do not take `Latin`.

You can then create:

- a division `German / Spanish`;
- a division `Latin / Spanish`.

The benefit is to give the engine more freedom to place certain lessons on the same time slot and reduce gaps for part of the class.

Use this with care: the actual distribution of students must be fully under control.

Validity criterion

No student may belong to two groups of the same division. This is the mathematical definition of a partition — the sets are disjoint.

If a student appears in two groups of the same division, Omniscol raises a consistency alert (the solver cannot send them to two places at the same time).

Effect on the algorithm

With a declared division, the solver:

- **allows** the groups of the division to be scheduled simultaneously (whereas this would otherwise be a conflict),
- **still checks the other resources**: you need as many different classrooms as simultaneous groups, and as many different teachers. The division only lifts the student conflict — not the classroom / teacher conflicts, which remain strict.

Frequent mistake — three courses in the same time slot in one class

If you want **three different simultaneous courses** in one class (Philo, Hist-géo, Maths spé) because your students have exclusive electives, create **three groups** (Philo, HG, Maths-spé) and put them in a division. Not three courses without a group — otherwise the solver will have no information about the simultaneity constraints, and each will be placed separately, at different times.

Difference from other constructs

You want...	Use
Several groups of one and the same class in the same time slot (different courses, disjoint students)	Class division (this page)
Several groups from different classes in the same time slot (same course, same teacher, same classroom) via a logic linking simple groups	Alignment
Same as an alignment, but easily adjustable afterwards, by creating a standalone grouping entity	Group of groups
A group whose membership is not fixed (workshop, open enrolment)	Free group

See also

[Class, group, subgroup](#)[Alignments](#)[Groups of groups](#)[Free groups](#)[Class division](#)

2.4 Group alignments

Source: <help/en/core-concepts/alignments.md> · id: [core-concepts.alignments](#) · Updated: 2026-05-20

An **alignment** links several [groups](#) belonging to **different classes**: they must take **the same course**, in the **same time slot**, in the **same classroom**, with the **same teacher**.

Important condition: the alignment must be based on the **same subject** in the different classes. If two classes have distinct subjects, even pedagogically very close ones, an alignment is not a good way to model the situation.

In practice, where a [class division](#) stays inside one class, an alignment spans several classes.

Use cases


Alignments are used to create transverse courses where several classes each have a small number of students to bring together (align) for one specific shared course, which they attend together. This is for example the case for:

- **Latinists** in 5A, 5B and 5C who take the Latin course together.
- **Cross-class electives** in high school: political science, art history, a third language, etc.
- **Speciality courses**.
- **Multi-programme core curriculum** in higher education: an introductory course shared by several master's programmes.

If you need to align **entire classes**, first create in each class a group representing that entire class, then align those groups.

Creation

1. First create, in each class concerned, **a group**, preferably with the same name (example: "Latinistes" in 5A, in 5B, and in 5C).
2. Go to the [Group alignment](#) tab.
3. Click [+ Add an alignment](#) and select the groups from the different classes.

When a group is part of an alignment, it appears on all relevant screens with a link icon .

Consequences

Once aligned, the group in each class is **tied** to the shared course. In practice:

- The course's lessons must be created in the classes as mirror images, with in particular the **same duration**, the **same teacher** and, where applicable, the **same classroom**.
- If a lesson of a class with an aligned group is positioned on a given day and time, then all the other classes in the alignment must have the same mirrored lesson, each with its corresponding aligned group, in the same position.
- The hourly volumes and the way they are split into lessons with the various aligned groups must be **strictly identical** for the subject concerned — otherwise Omniscol raises an inconsistency diagnostic.
- A change (moving a lesson, changing the classroom, changing the teacher, adding a memo) made on a lesson **must be propagated** to all aligned classes (the system tries to guess which lessons mirror each other to make this as automatic as possible).

Diagnostics — frequent inconsistencies

Alignments are sensitive to discrepancies. Typical diagnostics:

- **Diverging hourly volume:** the "Latinistes" group in 8A has 3 one-hour Latin lessons, the one in 8B has 2. Either harmonise (3 everywhere), or unalign (2 aligned hours, one hour on another, non-aligned Latinist group).
- **Different classroom on an aligned lesson:** an alignment implies a single classroom. If you manually force a different classroom on one class for its lesson, the diagnostic is raised.
- **Different teacher:** the same applies.

Tip — one group per course/subject

Because alignments are very sensitive to the requirement of perfectly mirrored creation, it is strongly recommended to create one group per subject or course concerned in each class to be aligned. In other words: do not reuse a generic group (typically: A/B groups) for different subjects when a strong alignment logic is involved. Otherwise you get ambiguous alignments that the solver cannot disambiguate. Instead, create dedicated groups, which you can link to each other without fearing unwanted side effects.

See also

[Class, group, subgroup](#)[Class divisions](#)[Groups of groups](#)[Transverse course](#)[Group alignment](#)

2.5 Groups of groups

Source: <help/en/core-concepts/groups-of-groups.md> · id: [core-concepts.groups-of-groups](#) · Updated: 2026-05-20

A **group of groups** is a super-group that gathers several **groups** — which can come from the **same class** or from **different classes**. Other scheduling software uses the term "grouping" for the same concept.

As with an alignment, the shared lessons must rely on a **common subject** across the classes involved.

Unlike alignments, which express a logic between groups, the group of groups is a standalone entity that can be used like a group. You can see it as a named, dynamic container of groups.

How it is more flexible than an alignment

Aspect	Alignment	Group of groups
Composition	Can be changed afterwards but requires creating the lessons in mirror	Can be changed afterwards very easily
Timetable mode	Well suited to the weekly and cyclic modes of secondary education	Usable in all timetable types; particularly convenient when groupings evolve
Taking a lesson out	Requires a non-aligned clone of the group	Direct edit of the lesson, whose ownership is then fixed to one class; the lessons of the other classes involved must be recreated
Use case	Recurring shared course, for the curriculum of parallel classes	Transverse courses, one-off groupings

The group of groups is designed for higher education and continuing education, where:

- the composition of groupings changes often (one track that joins another, a subgroup that splits off for a project),
- shared courses are dated individually rather than recurring,
- guest speakers or visiting professors teach a composite audience on specific dates.

A group of groups is more flexible and easier to track than an alignment. Its limitation lies in complex courses, with asymmetric week alternations. It only requires well-structured naming (for example: "Marketing transverse Master").

Creation and use

Creation happens from the **Group of groups** tab of the timetable. The group of groups can be assigned to a course **as if it were an ordinary group** — the difference is that it aggregates several member groups.

If you want to combine **entire classes**, first create in each class a group that represents the whole class, then use these groups in your group of groups.

A course assigned to a group of groups appears in **all** the parent classes of the member groups. And you can edit it from any of them.

Groups of groups appear on all relevant screens with a specific icon .

Changing the composition afterwards

Member groups can be added or removed at any time. The courses already assigned adapt automatically (the scope widens or narrows according to the groups added / removed).

This is what makes the concept valuable: you can start a semester with one composition, adjust it along the way, without breaking the structure.

Technical considerations

Technically, a lesson with a group of groups is stored once, transversally to the classes, so it is not assigned to a specific class. Whereas a lesson without a group of groups (including one with aligned groups) is stored in its class. With an alignment across 3 groups, 3 lessons are stored, one in each class; with a group of groups, a single lesson is stored apart from the classes, but displayed in each class.

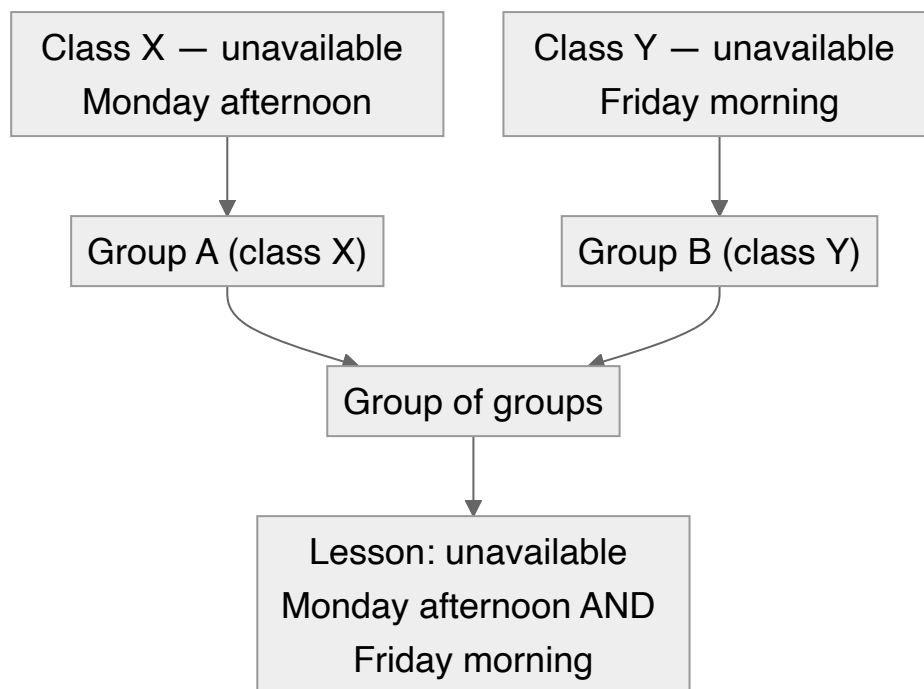
When a group of groups is removed from a lesson, the system guesses or asks which class the lesson should be attached to.

It is easier to edit the other fields of a lesson with a group of groups than with an alignment: single storage lets you change the classroom or the teacher immediately, with no "mirrored" propagation.

Consolidating unavailabilities

A group of groups **consolidates the unavailabilities** of all its member groups. An unavailability set on **any** member group — or on that group's class — applies to the shared lesson. The rule holds both during automatic generation and on screen while editing. Soft levels (Undesired, red) consolidate the same way; and when several levels meet on the same slot, **the strongest wins** — Unavailable (black) overrides Undesired (red). See [Time constraints \(general system\)](#) for the editor and the levels.

Because a group of groups often spans several classes, this consolidation gathers the unavailabilities of **every class it covers**: a class closed on Friday afternoon closes that slot for every lesson of the group of groups that includes it, even when the other classes stay free.



On the group of groups itself, the app shows a **consolidated, read-only preview** of these unavailabilities: you see all the members' constraints at a glance. You **edit them on the member groups** (or on their classes), where they carry their own meaning — the group of groups has no separate constraint editor.

PREMIUM

Each member group brings its **own** group constraints, as well as those it **inherits** from its class and its parent groups (see [Group hierarchy](#)). The consolidation takes them all on: the more members a group of groups aggregates, the tighter the placement window. To reopen a slot despite an inherited constraint, the **Allowed lessons** level re-allows it locally, without changing the source.

On screen, these consolidated constraints show up **at placement time**: the candidate slots and the diagnostic panel flag the conflict and name the **member group** and the **class** responsible, so you can trace it back to its source (see [Conflicts and diagnostic](#)).

☰ How-to

Creating a transverse group for a guest lecture

1. **A guest lecture for a composite audience** (students from three cohorts, two options): the matching Omniscol concept is the **group of groups**.
2. **Prerequisite:** the member groups already exist within their respective classes. Identify the groups to aggregate (M1 marketing, M1 finance, Entrepreneurship option).
3. **Open the** `Group of groups` **tab** of the timetable. Click the `+ Add a group of groups` button. Give it a meaningful name: `Guest lecture Dr Lambert – M1 students + electives`.
4. **Add the member groups:** select the groups of the three classes / options from the list. There is no composition limit. Member groups are not required to be mutually exclusive (at worst, a student who belongs to two groups will see the course only once).
5. **Create the course** (the lecture) from one of the classes involved, and **assign it the group of groups** as if it were an ordinary group. A single entry, and the course appears in **all the parent classes** of the member groups.
6. **Changes afterwards:** if the composition evolves (one group withdraws, another joins), edit the composition of the group of groups — the assigned courses adapt automatically (the scope widens or narrows). This is what sets this concept apart from a classic alignment (fixed composition). See also [Alignments](#) for the weekly case.

A more flexible alternative: direct multiple group assignment

Sometimes, for one-off lessons, creating a dedicated group of groups turns out to be a tedious operation. In that case, if the lack of data structure is not an obstacle, you can directly associate several groups from one or more classes with a lesson.

Internally, this is handled like an anonymous group of groups, with single storage transverse to the classes, but you do not have to worry about it.

📌 See also

[Alignments](#)[Time constraints \(general system\)](#)[Transverse course](#)[Calendar mode](#)[Group of groups](#)

2.6 Group hierarchy: parents, children, inherited constraints

Source: [help/en/core-concepts/group-hierarchy.md](#) · id: [core-concepts.group-hierarchy](#) · Audience: [admin](#) · Plan: [premium](#) · Updated: 2026-05-20

👑 PREMIUM

Premium feature. Group hierarchy and group time constraints are included in the Premium features.

PREMIUM

Beyond the flat "class → groups" model, Omniscol lets you organize a class's groups into a **parent / child hierarchy** and attach **time constraints** that propagate through inheritance. This is useful when the class is structured in several successive levels of subdivisions.

Building a hierarchy by drag and drop

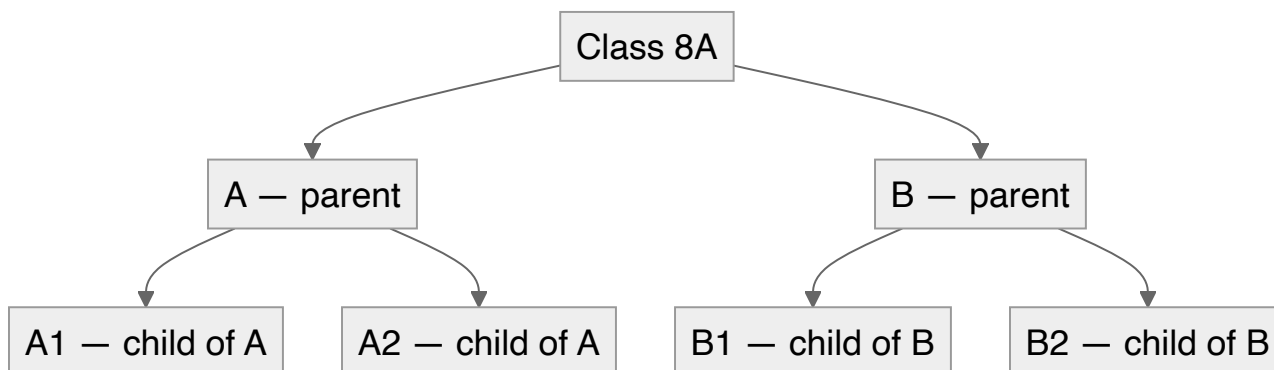
On a class's groups page, drag and drop defines the hierarchy:

- **Dragging one group onto another** → the child is attached to the parent.
- **Dragging a child onto the head of the group list** → the child is taken out of its hierarchy and becomes a direct group of the class again.

There is no depth limit. You can have parent → child → grandchild chains if your structure calls for it.

Canonical example

A typical hierarchy:



Declared divisions:

- (A, B) — A and B are mutually exclusive at the parent level.
- $(A1, A2)$ — under A, A1 and A2 are mutually exclusive.
- $(B1, B2)$ — under B, B1 and B2 are mutually exclusive.

Automatic consequence: Omniscol allows groups belonging to different branches to be scheduled simultaneously — for example $A1$ and $B2$ can share the same time slot, because no student belongs to both ($A1 \subset A$, $B2 \subset B$, and $A \cap B = \emptyset$).

There is no need to declare the division $(A1, B2)$: inheritance computes it.

Inheritance of time constraints

Beyond divisions, you can attach **time constraints** (availability, incompatibilities) to a group. With the hierarchy, these constraints are **inherited** from parents down to children — through a **mask** system that children can override.

Example: a time slot reserved for PE

A typical case:

- Class $8A$ is **unavailable on Monday afternoon** (a global slot blocked for collective activities).
- Through inheritance, **all the groups** of the class are unavailable on Monday afternoon.
- **Except** the $sport$ group: you **invert** its mask to allow Monday afternoon and forbid everything else. All of this group's sports activities will then automatically land on this protected slot.

The mechanism is powerful: **one parent mask + one child exception** is enough to model dedicated time slots cleanly, without duplicating the constraint on every course.

Another example: an $Electifs$ parent that carries the allowed slots

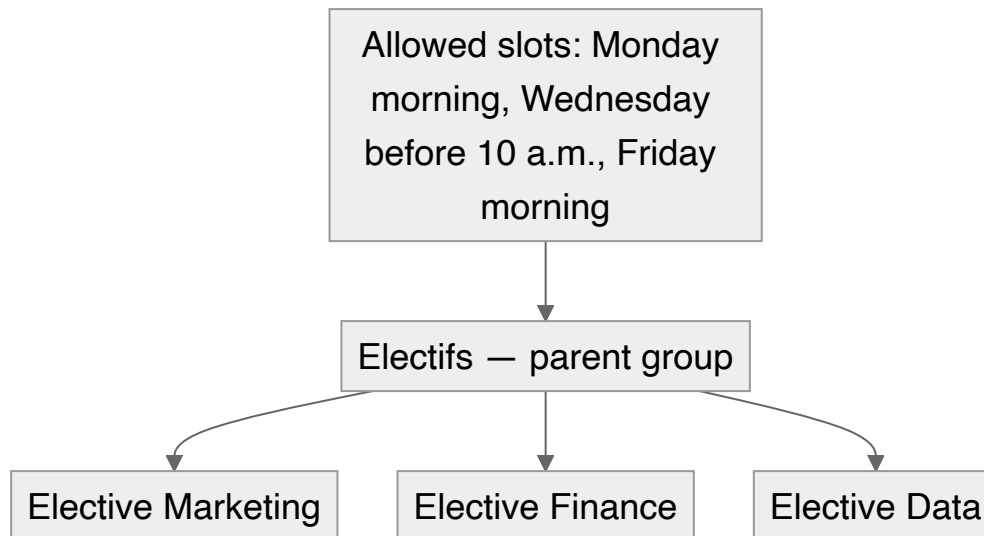
Another very useful case: the institution wants all electives to land only on a few reserved time slots, for example:

- Monday morning,
- Wednesday before 10 a.m.,
- Friday morning.

You can then:

- create a parent group `Electifs` that will not itself be assigned to courses;
- express the allowed slots on this parent by painting the time constraints;
- create below it as many subgroups as needed: `Elective Marketing`, `Elective Finance`, `Elective Data`, and so on.

All these subgroups then inherit the same allowed slots. If the rule changes, you edit a single place: the `Electifs` parent.

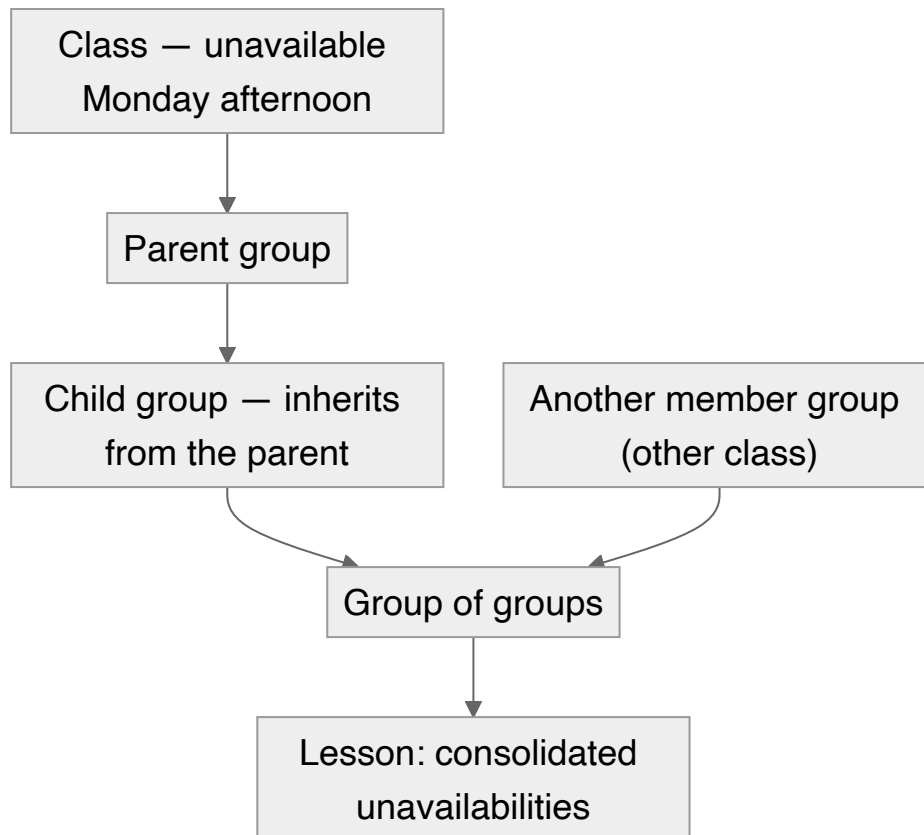


Inheritance rules

- A child group **inherits by default** its parent's time constraints.
- The child can **override** these constraints: add its own constraints or invert the mask.
- The **parent's divisions** apply automatically to the children: if A and B form a division, the children of A are mutually exclusive with the children of B without any explicit declaration.

Inherited constraints and groups of groups

A group can also be a member of a **group of groups**. Its constraints — its own as well as those inherited from its class and its parent groups — are then **consolidated** with those of the other member groups: the group-of-groups lesson respects all these unavailabilities combined. Inheritance flows down the hierarchy, then consolidation gathers the members of the group of groups.



See [Groups of groups](#) for the details of the consolidation.

When to use a hierarchy

- **Option bundles:** a class with languages structured in levels (Advanced English > A1-Conversation, A1-Civilisation, A2-...).
- **Practicals with sub-rotations:** two half-classes for the practical, each subdivided into subgroups for the experiments.
- **Differentiated tracks:** a main track with distinct optional modules that are scheduled within the track.
- **Sport:** a global group gathering all the students, with a dedicated mask for the sports slots.
- **Electives on reserved slots:** a parent group `Electifs` carries the allowed slots, and its subgroups correspond to the actual electives.

When not to use it

- For simple classes without hierarchical subdivision — the hierarchy adds complexity.
- To mix students from different classes — that is the role of [alignments](#) or [groups of groups](#), not of the intra-class hierarchy.
- For open enrollment — use [free groups](#).

☰ How-to

Building a parent/child hierarchy

1. **Group hierarchy** organizes a class's groups into parents and children, with **inheritance** of time constraints.
2. **First create the groups flat** on the class's groups page: A , A_1 , A_2 , B , B_1 , B_2 , plus a cross-cutting `sport` group. At this stage, they are all at the same level, attached directly to the class.
3. **Drag and drop to build the hierarchy**: drag A_1 onto $A \rightarrow A_1$ becomes a child of A . Do the same for A_2 , B_1 , B_2 . You get two branches $A \rightarrow A_1, A_2$ and $B \rightarrow B_1, B_2$. There is no depth limit.

To take a child out: drag it onto the **head of the list** of the class's groups.

4. **Declare the divisions**: (A, B) at the parent level (mutually exclusive), (A_1, A_2) under A , (B_1, B_2) under B . No need to declare (A_1, B_2) as well: through inheritance, Omniscol works out that $A_1 \subset A$ and $B_2 \subset B$ are compatible (schedulable on the same time slot).
5. **Set the parent mask**: on the class, mark **Monday afternoon as unavailable**. All the groups inherit it automatically. Check by viewing each group's grid: the slot is blocked everywhere.
6. **Invert the mask on `sport`**: on the Sport group's record, invert the time constraint — allow Monday afternoon, forbid everything else. All sports activities automatically land on this dedicated slot, without having to declare it again on every course.
7. **You now have an active hierarchy** with inheritance of divisions and a dedicated time mask. The solver uses it during automatic generation, and inherited conflicts are reported in real time.

🔍 See also

Class, group, subgroup

Class divisions

Free groups

Availability, time constraints and incompatibilities

2.7 Free groups

Source: help/en/core-concepts/free-groups.md · id: `core-concepts.free-groups` · Plan: `premium` · Updated: 2026-05-12

👑 PREMIUM

PREMIUM

A **free group** is a **semi-autonomous working group**: part of the class moves forward on its own work alongside the main lesson, often in the same room and with the same teacher — a **satellite group** of the course. It sits on the grid without generating any conflict, and its membership does not have to be fixed in advance.

It is Omniscol's **wild card**: use it when the strict model of a class plus subgroups in a division does not fit. Very common in:

- **art schools** (open-enrollment workshops, cross-disciplinary projects),
- **semi-autonomous projects** — a small group of students who work together under the **light** supervision of a teacher who mainly looks after their own class nearby and regularly stops by to check on their progress,
- **cross-class lectures**, masterclasses, guest seminars,
- voluntary **remedial sessions** and **clubs**,
- **research projects and short internships** with open enrollment.

Behavior

A free group is **attached to a class** like any group, with a headcount and, depending on the account configuration, a list of students. What changes is how Omniscol checks for conflicts.

On a free group's lesson, Omniscol **disables three types of checks**:

- conflicts with the rest of the class on the same time slot,
- conflicts with another lesson of the same teacher,
- conflicts with a classroom that is already occupied.

This is what lets the free group work as a **satellite group** on the main class's grid: the main lesson takes place, and the free group's lesson sits alongside it without triggering any alert.

A free group's lessons **appear explicitly** in the timetable of the students concerned, like every other lesson.

When to use a free group vs a regular group

Situation	Regular group	Free group
Membership fixed in advance	✓	✓ (like an ordinary group: headcount, optional student list)
Class / teacher / classroom conflict detection	✓ strict	— disabled
Modeling electives in a class division	✓	— no
Satellite lesson, alongside the main lesson	—	✓


Usage precaution — the real limit

First and foremost: prefer real subgroups in a division whenever you can. The free group is a **wild card to be used sparingly**, reserved for cases where no other clean solution works.

The reason: in practice, the parallel placement must be **managed by the human planner**. The automatic generation algorithm has no way of knowing that a given semi-autonomous group must be scheduled **alongside one specific lesson of the class's main course**. If you let it run, the free group's lessons can land at any time — regardless of what the rest of the class is doing.

Good practice: manually position the free group's lesson **on the same time slot as** the targeted main lesson, then **lock its position** so that automatic generation does not touch it. See [Locking a lesson](#).

Creation

When creating the group (using **+ Add group**), check the **Free group** option. An  icon then marks these groups in the lists so they are not confused with ordinary groups.

Effects of disabling conflicts

Concretely, on a free group's lesson:

- if Alice (a student in the class) attends this lesson while another main lesson takes place for the rest of the class on the same time slot, Omniscol raises **no alert** (even though she is in theory expected in two places at the same time);
- the teacher can be assigned to another lesson on the same time slot, again without any alert;
- the classroom can already be occupied by the main lesson without an alert.

This tolerance is **deliberate**. It is up to you (the planner) to ensure that this coexistence makes pedagogical sense (typically: the teacher just stops by from time to time to check on the semi-autonomous group; the group's students step out of the main lesson without causing any problem; the room hosts both activities).

See also

[Class, group, subgroup](#)

[Class divisions](#)

[Locking a lesson](#)

[Free group](#)

2.8 Courses, lessons, course types

Source: <help/en/core-concepts/lessons-and-types.md> · id: [core-concepts.lessons-and-types](#) · Updated: 2026-06-13

Three related but distinct objects to tell apart in Omniscol: the **course** (the abstract teaching unit), the **lesson** (a concrete instance of the course, on a specific date), and the **course type** (the classification: lecture, tutorial, practical...).

Course

A **course** is the atomic teaching unit: a subject, a class or a group, a teacher, a classroom, a duration, a number of weekly occurrences (or a calendar of dates in calendar mode).

Example: *"Mathematics in Grade 6 A, taught by Mr. Durand, 4 hours per week in room B204"*.

The course carries the **teaching logic**: who learns what with whom. This is what you build in the Timetable management module. See also the glossary definition: [Course](#).

Lesson

A **lesson** is a concrete **occurrence** of a course, with a date and a time. From the course *Mathematics in Grade 6 A, 4h/week*, Omniscol generates 4 lessons per week over the active weeks (for example Monday 8-9 a.m., Tuesday 10-11 a.m., Wednesday 2-3 p.m., Friday 9-10 a.m.).

The lesson is what is visible in the Timetable module, in the portals, and in the iCal exports. One-off changes (moving, canceling, substituting) also apply to the lesson.

It is the term the interface uses for the unit placed on the grid: what you place, move, lock, edit, cancel or substitute is a **lesson** (or "course lesson" when the extra precision helps).

Course type

A **course type** classifies each course beyond its subject: lecture, tutorial, practical, exam, remote, field trip. The type is used for:

- **business logic** (an exam is not handled like a tutorial),
- **statistics** (tutorial hours vs lecture hours per teacher),
- **readability** (display in the timetables).

See [Types of course \(tutorial, practical, exam, lecture, etc.\)](#) for detailed type management.

Where courses live in the structure

Depending on how complex the class structure is, courses are attached either **to their class** (the standard case) or **transversally to the classes** (when groups of groups are used or when a lesson has several groups at once). This is transparent for the user — Omniscol switches automatically according to the structure. See [Groups of groups](#).

Simple vs complex courses

A course can be **simple** (one teacher, one class, one subject, one classroom, a regular recurrence) or **complex**: A/B week alternation, concatenations (double lessons), associations (rotation between groups), co-teaching (several teachers). See [Complex lessons](#).

See also

Lesson / Session

Type of course

Types of course (tutorial, practical, exam, lecture, etc.)

Complex lessons

2.9 Complex lessons: alternate, concatenated, associated, co-taught

Source: help/en/core-concepts/complex-lessons.md · id: [core-concepts.complex-lessons](#) · Updated: 2026-06-13

Many lessons are simple: one audience, one teacher, one room, a recurring time slot. But Omniscol also handles more sophisticated configurations, which can be combined with one another. Complex lessons are lessons of the same class linked together by a special configuration. There are several types.

1. Alternate lessons (weeks A/B/...)

Lessons that do **not recur every week**, but alternate. The classic case: one lesson in week A, another in week B, in the same time slot.

This configuration is very popular in France and in the school systems inspired by it, but very rare elsewhere.

Prerequisite: enable alternate weeks in Settings (Administration module). You can choose the naming scheme (letters A/B/C or numbers 1/2/3).

Offset caused by holidays: the alternation can be realigned after a holiday period. This is set on the timeline of the [school year](#) screen.

Creation:

1. Hover over the lesson to alternate. A **+** icon appears at the top right of the lesson; click it.
2. A free slot is added for the alternate week.
3. Create the lesson corresponding to this new slot.
4. Position it with its pin button **📌**, then click the desired day/time slot among the colored placeholders.

You can repeat this to add more weeks (alternation over 3, 4, or more).

2. Concatenated lessons (consecutive)

Two lessons that must **follow each other within the day**, with nothing in between.

Typical cases:

- back-to-back practical sessions for two different groups (the teacher knows they will be in for a full half-day),
- a lecture followed by a tutorial (an introductory lecture, then a tutorial to apply it),
- a double exam session.

Creation: drag and drop one lesson under another in the hours distribution view. The two lessons are then stuck together, and are treated as a single block.

You can concatenate as many lessons as you like, but of course the block might not fit in the time grid if the total duration is too long.

Detaching: a "scissors" button appears on hover between the two concatenated lessons. It splits the concatenation.

Why not a single, longer lesson? Because the two lessons can have:

- different **types** (a lecture, then a tutorial),
- different **teachers** (the main teacher for the lecture, an assistant for the tutorial),
- different **rooms**.

If all the attributes are identical, lengthening a single lesson is simpler — concatenation brings flexibility in the other cases.

3. Associated lessons (alternating half-groups)


Two simultaneous lessons whose **half-groups swap** with each other, forming four lessons in total. The archetypal case:

```
Slot 1: group A in biology,      group B in physics
Slot 2: group A in physics,      group B in biology
```

After the two time slots, both half-groups have covered both subjects, but in a different order. The swap therefore requires two teachers (one for biology, one for physics) who "rotate".

This configuration is fairly popular in middle schools in France and very popular in French-speaking countries in Africa. It seems much rarer elsewhere, but you can come across it in Brazil, for example.

Creation:

1. Create two **concatenated** lessons (one under the other).
2. Hover over the boundary between the two: an  button appears.
3. Click it and designate the **two groups** that must alternate. Ideally, the two groups should be declared as a **class division**.

It is possible to go beyond two lessons/two groups, but this case remains more theoretical than actually encountered in real life.

4. Co-teaching

Two or more teachers delivering **the same lesson**, in the **same room**, **simultaneously**.

Typical cases:

- reinforced supervision (main teacher + assistant),
- a *visiting professor* paired with the main teacher for a few lessons,
- a multidisciplinary module (a physician + a computer scientist for "computing for healthcare"),
- co-lead teachers over a whole semester.

Creation: on the screen for assigning teachers to the course, select **several instructors**. Each one is credited with the lesson in their statistics and their service hours.

5. Combinations

These complexities can be **combined**. Examples:

- **Alternate + concatenated** — the week-A lesson is a concatenated double block, and in week B it is another double block.
- **Alternate + co-taught** — week A with the main teacher, week B with the visiting professor (alternating instructors).
- **Concatenated + associated** — half-groups alternating over a double time slot (the most common case: science practicals in half-classes).

In France, this is typically how schools declare the extra half-hours of French and mathematics while also solving the three-hours-of-sports problem (achieving 1 hour on average through 2 hours every other week, on top of 2 hours every week).


Week A	Week B
French	Sports
Mathematics	Sports

Unusual cases

If your need does not fit into any of these categories, contact Omniscol support — it can most likely be modeled, but it may take a little help to identify the best approach.

☰ How-to

Create a practical session with associated half-groups

1. **The archetypal case:** a science practical over two time slots, where half-group A does biology then physics while half-group B does physics then biology. Concatenation + association.
2. **Prerequisite:** on the class concerned, declare the two half-groups as a **class division**, for example (A, B). See [Class divisions](#). This is what allows Omniscol to treat them as mutually exclusive and therefore swappable.
3. **Create the first lesson** (for example biology for half-group A) in the first time slot, in the hours distribution view. Biology teachers and room.
4. **Drag and drop the second lesson** (physics for half-group B) **directly under the first** (a slot appears as you release the lesson, which ends up stuck to the first).
5. **Hover over the boundary** between the two concatenated lessons: an  Association of lessons button appears. Click it. Designate the two groups (A and B) that must alternate.

Omniscol automatically creates the **rotation**: slot 1 → A in biology + B in physics; slot 2 → A in physics + B in biology. Two simultaneous teachers, two rooms, two half-groups rotating, four lessons in total.

6. **To adjust:** a scissors button between the two concatenated lessons splits the association. You can also combine this with **A/B alternation** (a different practical in week A vs week B) or **co-teaching** (two teachers on the same lesson). See [Distribute the hours and create the lessons](#) for the full context.

🔗 See also

[Alternate lessons](#)

[Concatenated lessons](#)

[Associated lessons](#)

[Co-teaching](#)

[Distribute the hours and create the lessons](#)

2.10 Campuses, sites, classrooms, resources, multi-room

Source: help/en/core-concepts/sites-rooms-resources.md · id: [core-concepts.sites-rooms-resources](#) · Updated: 2026-06-13

This page brings together the **organizational and physical** modeling of a school in Omniscol: which campuses structure the classes, where lessons take place, in which classrooms, with which movable equipment.

PREMIUM

Campuses — the organizational level

A [campus](#) distinguishes several locations, faculties or divisions within a single account. It is an internal organization concept: it groups classes according to the structure of the school and makes certain filtering easier.

This concept is **optional**. It becomes most useful when the site concept does not match the logical organization of the school well, or when you want to keep both in parallel.

The campus cuts across sites. A site describes a geographical or physical reality: location, time grid, classrooms, travel times. The two concepts do not necessarily overlap. That is precisely when they become most useful. For example:

- a faculty can use several sites;
- a physical site can host several campuses or divisions;
- several faculties can share the same buildings;
- several schools in a group can share several sites in a criss-cross pattern;
- a campus can serve mainly as an analysis filter, without changing travel times.

Even when campuses and sites overlap almost completely, keeping both can remain useful for filters, groupings and certain analyses.

Campuses are created in the [general settings](#), below the class levels. They can then be set on classes. The conflict diagnostic can use them to filter alerts by organizational scope.

Sites — the geographical level

Each [site](#) represents a distinct physical or geographical location. At least one site is required.

If all your lessons take place in the same location, create a single site. If your school is multi-site with instructors or students who move around, create as many sites as needed **and enter the travel times** between them.

Each site carries its own **time grid** (slot start/end times, breaks, lunch, closures) — sites can have different grids (for example a main building and a satellite location with different hours).


Multi-site setups are a common practice in **higher education** (multiple campuses, regional branches), but also in **primary and secondary education** (a middle school + a high school in the same school group with separate buildings, a school with a preschool annex, etc.).

Multi-site is not included in the **Lite** plan. It is available from the Standard plan upwards.

Multi-site: policy and travel times

Typical case: a teacher finishes a lesson on site A at 10:00, must be at site B at 10:30, and the trip takes 15 minutes. On a time grid with a 30-minute break between lessons, this works. With only a 10-minute break, it does not fit.

Entering travel times

On the **Sites** screen of the Timetable management module, the  **Distance between sites** button opens a popup that presents travel times as a **triangular half-matrix**: N sites = $N \times (N-1) / 2$ cells to fill in (no diagonal, no duplicates — the matrix is symmetric). Enter the durations in minutes for each pair, then confirm with **Save**. Without this input, the solver assumes that no travel is needed between the sites — it may teleport teachers, which obviously does not work in practice.

Teacher policy: automatic site changes

A **teacher** can change site automatically from one lesson to the next — the algorithm handles it. There is nothing to declare on the teacher side: the inter-site travel times just need to be entered.

Class policy: default site + manual override

A **class** is attached to a site by default (its enrollment site). All its lessons take place on this site by default.

But you can **manually attach a classroom from another site** to one of the class's lessons — for example "PE lesson in the annex gymnasium". In that case:

- the **preassigned classroom** is kept by the algorithm (it does not try to replace it with a classroom from the default site);
- the solver computes the **travel times** relative to the lessons before and after (based on their site: the class site by default, otherwise the site of the classroom forced on those lessons).

If **no classroom is specified** on a lesson, the algorithm picks a classroom from the **class site**. This is the default behavior.

A blocking constraint, to the minute

The inter-site constraint is **blocking** in the automatic generation algorithm. If even one minute is missing between the end of a lesson on one site and the start of a lesson on another site, the placement is **rejected** — there is no automatic relaxation.

The check is precise on the **effective time grid**: if two slots follow each other with a 15-minute recess and the trip takes 10 minutes, it fits; if the trip takes 16 minutes, it does not.

Lunch break on a site change

When a site change falls on the lunch break, the travel time is **subtracted from the actual lunch time** available. The algorithm takes a minimum lunch duration into account for teachers and for students (set per site); if the trip reduces this time below the minimum, the placement is rejected.

Practical consequence: declaring travel times and per-site lunch durations correctly avoids timetables that are theoretically valid but physically impossible to follow.

Impact on the diagnostic

If you see "impossible" on multi-site placements, check first:

- the declared **travel times** (no teleporting, no absurd overestimates),
- the **breaks between lessons** on the time grid,
- any **lunch break** on the slot concerned,
- the **preassigned classrooms**: if a class has a classroom forced on another site, that classroom sets the constraint.

In manual placement, the constraint appears in orange, and you can always force it. But unless the lessons are locked, running the automatic generation algorithm afterwards will move the lesson(s) to resolve this conflict.

Use case: off-site activities

Beyond permanent campuses, the multi-site mechanism naturally covers **recurring off-site activities**:

- **hotel or restaurant visits** for a hospitality / catering school;
- **museum or workshop visits** for an art or design school;
- recurring **company visits** (work-study visits);
- **field work** on an external site (construction site, farm, partner laboratory).

Create a dedicated site for each recurring external location, with its travel time from your main site. The blocking constraint applies as usual: if the trip does not fit in the time available between two lessons, the placement is rejected.

Going further: higher education

Multi-site setups are especially common in higher education (multiple campuses, regional branches, instructors moving around). Specific cases (one timetable per site thanks to multiple active timetables — included in Premium —, virtual sites for videoconferencing, etc.) are covered in [Multi-site in higher education](#).

Classrooms — the premises

[Classrooms](#) are attached to a site, which carries their physical reality: location, time grid, travel times. A classroom carries:

- a **name**,
- a **capacity** (number of students; a critical field for the solver),
- optionally a **specialisation** (free label: chemistry, computing, sport...),
- optionally a **maximum number of simultaneous classes**, for the **large rooms** able to host several lessons in parallel (exam room, gymnasium, swimming pool, outdoor space); this field only appears once a specialisation is set,
- optionally free **tags** or comments,
- optionally a **building**,
- optionally specific **opening hours**.

Multi-room: one lesson in several classrooms

Omniscol **supports** assigning several classrooms to the same lesson.

Use cases:

- **Exams split across lecture halls** — a 200-candidate exam spread over three lecture halls (capacity 70 + 60 + 80) with a single instructor in charge. The total capacity is computed as the **sum** of the assigned classrooms.
- **Broadcast lecture** — a lecture in a main lecture hall, streamed by videoconference to a satellite room (on another site, or even to a fully remote cohort).
- **Split practicals** — a practical session of 30 students spread over two neighboring rooms (15 + 15) with the same teacher moving between them.

Limitation: if the sum of the capacities remains **below** the group headcount, Omniscol displays a conflict. It is up to the administrator to decide how to resolve it (by adding a classroom, reducing the group, or accepting the conflict if it is deliberate — for example, when you know that not all enrolled students will attend).

Classroom specialisations

Specialisations (chemistry, computing, sport, gymnasium, multimedia, supervised-study room) are optional **free** labels defined by the school. You create the specialisations that match your naming scheme, to single out certain rooms that are dedicated to specific lessons (a gymnasium for sport, a swimming pool for swimming, a laboratory for a chemistry practical, etc.).

The solver strictly respects specialisations: if a subject requires "chemistry", only the classrooms carrying exactly this specialisation will be used.

Each classroom carries **at most one specialisation**. For a room shared between two specialised uses, use an umbrella label (for example "computing-multimedia") and associate this same label with the subjects concerned. See [Classroom specialisations](#).

Classroom dedicated to a class

Many primary/secondary setups associate a classroom with each class (lessons take place there by default, only the teachers move). This is configurable on the class — the engine then prioritizes this classroom for the class's lessons (with a priority higher than the teacher's preferred classroom, if one is defined).

Classrooms in two virtual sites

If you have two virtual sites for a single physical location (typical of a middle school + high school sharing the premises), a classroom can only belong to one site at a time. To make it usable in both contexts, duplicate it in both sites and enter exclusive opening hours ("free in the morning for the middle school, in the afternoon for the high school") to avoid double bookings.

Resources — movable equipment

A **resource** is a piece of **movable** equipment not attached to a particular classroom: three portable projectors, a case of tablets, a microphone kit, etc.

Each resource carries:

- a **name**,
- an **available quantity** (the case counts as 1, not 30 — enter the number of cases, not individual tablets).

The solver guarantees that on any given slot, the number of lessons requesting the resource does not exceed the available quantity. There is no point in modelling resources you "always have enough" of — do it only for real shared limits.

Special case — videoconferencing and per-course links

The format of a lesson — in person, remote, hybrid or self-study — is carried by its **modality** (Premium): it determines whether the lesson uses a physical classroom. For a remote or hybrid course, you can attach a **videoconference link** to the course (Zoom, Teams, Meet...). See [Videoconference links per course](#).

This is finer-grained than the videoconference link on the class, which is a general default link on all its lessons.

See also

[Classroom specialisation](#)

[Sites, time grids, classrooms, resources](#)

[Multi-room exams in higher education](#)

[Videoconference links per course](#)

[Modality](#)

2.11 Time grid, time slots and durations

Source: help/en/core-concepts/timetable-grid.md · id: core-concepts.timetable-grid · Audience: admin/teacher/staff · Plan: standard · Updated: 2026-06-13

The **time grid** describes a site's time slots: start times, end times, breaks, lunch break and closures. It serves display, availability checks and automatic generation alike.

Standard storage: the time slot, not the written time

By default, a placed lesson is attached to a **slot of the grid**. Omniscol stores the position in the grid, then recomputes the displayed times from the site's grid.

A useful consequence: if one day the starting bell changes from `08:15` to `08:10`, the lessons placed on that slot follow the new grid without having to edit each lesson one by one.

Grid display mode

The **display mode** determines how the hours column is presented, without changing lesson placement in any way:

- **Lesson hours** — each row carries the slot's real times (8:15 – 9:10, 9:15 – 10:15...), as defined by the site's grid.
- **Periods** — each row carries a period number (P0, P1, P2...), a widespread convention in English-speaking schools.
- **Calendar** — calendar-style display on round hours (8:00, 9:00, 10:00...), independent of the grid: each lesson sits at its real time on a continuous time background.

Base lesson duration

The **base lesson duration** is the reference duration of a time slot. It can include the gap between lessons if the institution counts it that way.

Examples:

- 55-minute lessons with a 5-minute gap between them: base duration of `60` minutes;
- regular 1.5-hour lessons: base duration of `90` minutes;

- mostly one-hour lessons, with a few 1.5-hour lessons: base duration of 60 minutes and a suitable lesson unit division.

This duration is used for the configured hour volumes and for generation. Real start and end times can vary within the grid: a recess can make a displayed slot last 55 minutes while remaining attached to a reference duration of 60 minutes.

In the interface, this setting is named **Time slot duration** and is defined on the **General** tab of the timetable editor (see [General settings](#)).

Typical durations, in order of popularity, are: 60, 45, 50, 120, 90, 30, 180.

Sometimes, in higher-education institutions, lessons always last several hours, but they may start at either 8:00 or 9:00, or last 6 or sometimes 5 hours. In that case the greatest common divisor prevails, and it will usually be a base duration of 60 minutes.

Lesson unit division

The **lesson unit division** indicates how many sub-parts the base duration can be cut into. It is the minimum time step. A few examples:

- 1 : only lessons of a full base duration;
- 1/2 : a 30-minute step if the base duration is 60 minutes;
- 1/4 : a 15-minute step if the base duration is 60 minutes.

Choose the simplest granularity that covers the actual needs. A finer division allows more flexibility, but increases the number of possibilities to test during generation.

In the interface, this setting is adjusted on the **General** tab through **Time slot division**.

In most school systems around the world, the time grid has no subdivision. In France, it is usually 1/2 in secondary education (a few lessons last 1.5 hours). In some training organizations, precision can go down to 10 minutes: that is a base duration of 60 minutes with a 1/6 division.

Position and duration: storage in time steps

A lesson is not recorded with "fixed" times but as a **number of time steps** on the grid. The **step** is the base duration divided by the lesson unit division: a base duration of 60 minutes with a 1/2 division gives a 30-minute step. A one-hour lesson then occupies 2 steps, a 1.5-hour lesson occupies 3.

Three notions coexist without merging:

- the **reference duration** (the base duration) is used to count hour volumes and to **display the duration** of a lesson;
- the displayed **real times** (start and end) come from the **site's grid**, slot by slot;
- **storage** happens in **time steps**, at the granularity of the lesson unit division.

This is why the same slot can display 8:15 – 9:10 (real times) while counting as 1 hour (reference duration).

Global slot bans

Some slots must remain **banned for the whole institution**: Wednesday afternoon, a meeting slot, a reserved range. These bans are set **directly on the site's grid**, by marking the slot as **unavailable**: no lesson will be placed there, neither in display nor in generation.

This is a **time constraint carried by the grid**. On Premium accounts, the same mechanism also allows soft constraints (slots to avoid without closing them). See [Time constraints \(general system\)](#).

Lunch break and offsets

The lunch break is described by a **range** (for example 12:00 – 14:00) and a lunch **duration** to preserve. When the duration to preserve is **shorter** than the range, Omniscol allows a lesson **on part** of the break, as long as the lunch

duration remains possible — before or after the inserted lesson. It is very common for the lunch break to last two hours while a one-hour lesson may still be positioned on it — usually electives where the whole class is not present, or transverse courses where synchronizing several classes is complex. The parts of the class that have no lesson are not considered to be in study hall (for the algorithm, it is not a gap).

The lunch break is also the only case where an irregular slot is treated as a set of distinct time steps. With a 30-minute time step, a base lesson duration of 1 hour and a break range from 12:00 to 13:30, the software considers it as 3 × 30 minutes. You can declare either a single 1.5-hour slot, or 1 hour + 30 minutes, or 30 minutes + 1 hour — it does not matter. In that configuration, the afternoon lessons are shifted by half an hour. If you then declare a minimum lunch duration of half an hour, this leaves the possibility of placing a one-hour lesson on the break, letting the students concerned have lunch either before or after the lesson.

You can also indicate that the **canteen opens just before** the official break: a "gap" on that slot then does not count as lost time for students.

What matters is modeling the intended rule: open slots, closed slots, flexible lunch break, availability or unavailability. The grid must reflect what the institution actually accepts.

PREMIUM

Custom times on a lesson

A lesson can receive **custom times**: a precise start and end that do not necessarily match the exact boundaries of a slot. This case exists in every timetable type.

The lesson is displayed off the grid, with its real times: it can start or end outside the boundaries of the displayed slot. Omniscol uses these custom times to compute its actual overlap. Every entity assigned to the lesson — teacher, room, class, group or resource — is considered busy as soon as the custom times encroach on a slot.

For automatic generation, a lesson with custom times is treated as unmovable: its position is locked automatically.

PREMIUM

Off-grid class

The **off-grid class** is another case: the class's lessons are stored with precise times, independently of the site's grid. This feature only concerns calendar-type timetables.

It mainly serves continuing-education courses or very flexible programs that share premises and teachers with initial training, without following the same time grid. The institution then chooses a default time step, for example a quarter of an hour, and lessons are stored with their start and end times.

PREMIUM

Custom durations

A lesson can also carry **custom durations**:

- **Calculated duration**: duration deduced from the lesson's custom times;
- **Actual duration**: duration given priority in dashboards;
- **Accounted duration**: duration used for billing or the teacher's pay.

Example: an exam occupies a room from 08:30 to 11:30 because the room must be prepared beforehand and cleared afterwards. The dashboard may count only 2 hours of exam as actual duration, while the accounted duration of the supervising teacher can be higher if the institution includes compensation for grading.

See also

General settings

Sites, time grids, classrooms, resources

Off-grid lessons

2.12 Classroom specialisations

Source: <help/en/core-concepts/classroom-specializations.md> · id: `core-concepts.classroom-specializations` · Updated: 2026-06-13

Not all classrooms are equal: a chemistry lab cannot host a sports lesson, a computer room only makes sense for the subjects that use its workstations, a gym is reserved for sports. **Classroom specialisation** in Omniscol lets you declare these affinities so that automatic generation and diagnostics respect the physical constraints.

How it works: a label shared between classrooms and subjects

Specialisation relies on a **free-form label**, defined by the school according to its own naming conventions, which links two declarations:

- **On the classroom side** — each classroom carries **at most one specialisation** ("chemistry", "computing", "gym"...). Most classrooms have none.
- **On the subject side** — a subject assigned to a class can require a specialised classroom through its Special classroom field. The value must match **exactly** the label carried by at least one classroom.

The link is strict in both directions: a subject that requires "chemistry" is only placed in a classroom carrying that label, and a specialised classroom is **reserved** for the subjects that request its label — it is not used for ordinary lessons.

The classroom's **capacity** remains a separate check: whatever the specialisation, the expected headcount must fit in the room.

Effect on generation

During automatic generation, for each course, the algorithm only considers **compatible** classrooms: the specialisation label required by the subject (if any) and enough capacity for the headcount.

If no compatible classroom is available, generation reports it and the diagnostic identifies the subject, the class and the specialisation involved. The decision is yours: add the label to another classroom, remove the requirement carried by the subject, or review the headcounts.

Multi-purpose classrooms

A classroom carries only **one** specialisation label. For a room that must serve two specialised uses — for example computing and multimedia — create an umbrella label ("computing-multimedia") and set that same label on all the subjects involved.

Generic classrooms

Generic classrooms (without a specialisation) host all the subjects that do not require a specialised room. This is the most common case; add a specialisation only when you want to restrict.

Large room: several lessons at once

A specialised classroom normally hosts **one lesson at a time**. You can turn it into a **large room** by setting its **maximum number of simultaneous classes**: it then hosts several different lessons in parallel (distinct teachers and groups), as long as the combined headcount fits within its capacity. This is the case for an exam room, a gym, a swimming pool or an outdoor area. This setting only appears **after** the classroom has been given a specialisation.

Best practices

- **Start broad** — during initial setup, do not add too many constraints, or you may block generation.

- **Refine after the first diagnostic** — once you see the generation result, you identify the cases where a missing specialisation would have prevented a bad placement.
- **Document it for the team** — teachers and room staff must understand why a given room is restricted; otherwise they work around it by hand.

☰ How-to

Setting up a science lab

1. **Set up a specialised classroom** (science lab, computer room, gym) so that automatic generation gets it right.
2. **Open the classroom** in the timetable's Sites / Classrooms / Resources screen. Fill in its **specialisation**: pick an existing label or create one (for example `science`). A classroom carries only one label.
3. **Capacity**: number of seats. Generation checks that the assigned group's headcount stays \leq capacity. \triangle Be realistic — too strict blocks generation, too loose leads to overbooking.
4. **On the subject side**: on the relevant subject(s) of each class (Life and earth sciences, Biology...), fill in the Special classroom field with the same `science` label. The match must be exact — it is what reserves the room for these subjects.
5. **Start broad** at initial setup — fewer constraints = less risk of blocking generation. You will **refine after the first diagnostic** when you see the cases where a missing specialisation would have prevented a bad placement.
6. **For an unrestricted (generic) room**, leave the specialisation empty: the room then hosts all subjects with no particular requirement. This is the simplest default — add the specialisation only when you want to restrict.

🔗 See also

[Classroom specialisation](#)

[Large room](#)

[Sites, classrooms, resources, multi-room](#)

[Managing subjects](#)

2.13 Teacher availability and time constraints

Source: <help/en/core-concepts/wishes-and-availability.md> · id: `core-concepts.wishes-and-availability` · Updated: 2026-06-21

A teacher's **availability** tells the software which time slots are **impossible**, **undesirable** or **preferred** for them. The solver strictly respects the impossibilities and tries to optimize the other preferences. In secondary education these are also called **wishes** — the same concept.

Four levels

Color	Meaning	Effect
Black	Impossible	Hard constraint: the solver never places a lesson there
Red	Undesirable	Soft constraint: the solver avoids it when possible
Green	Preferred	The solver favors this slot when several are valid

An unmarked range is **neutral** — available with no preference.


Enabling availability entry

The entry mode is chosen in the [general settings](#), on the Teachers' availability setting:

- **Disabled** — no availability entry by teachers.
- **Weekly** — availability entered on a typical week. For weekly and cyclic timetables.
- **Calendar** — availability entered date by date. For calendar timetables, on Premium accounts.
- **Calendar + Weekly** — both entry modes coexist, on Premium accounts, when recurring and calendar timetables live side by side, or simply to give teachers a simpler way to state that they are never available on a given day of the week, whatever the date (exceptions can still be entered).

Entry by teachers themselves

Teachers can enter their own availability from their account . A major time saver for the administration.

If the school prefers that the teacher not sign in, you can also **send them a direct share link** to their own availability screen, in edit mode, generated from  Download with an expiration date. The teacher enters their availability even if their account has no sign-in permission.

In practice, this link is often sent in a personalized message with the planned subjects, the hour volume, travel or specific constraints, and an explicit deadline. Omniscol does not provide a dedicated mail merge for this case: each link slots into the communication already prepared for the corresponding teacher.

Administrative validation in the timetable

In a weekly timetable:

- Availability entered by the teacher is by default **pending validation**.
- The planner checks and **validates** (potentially after edits — for example removing an availability entry that is too restrictive, or turning black into red).
- Any later change by the teacher goes back to pending.

In secondary education, availability entry can be opened several weeks before planning. Entries are thus made on a generic time grid, and the validation phase adapts them to the timetable's actual lesson time grid: the validation phase also serves to carry out this transposition correctly.

Availability of virtual teachers is saved directly. This makes it possible, for example, to grant Wednesday off by default.

In calendar mode (**Premium**):

- Availability is **consolidated in real time**.
- Conflicts are detected directly.

In calendar mode, the grid is indeed considered less constrained (since there are many possible dates), and teachers are autonomous — sometimes external to the school, juggling positions and responsibilities: this is not a wish but actual availability.

As an exception, teacher availability data in calendar mode can therefore be edited both from the administration module's screen and from a timetable.

Availability conflicts at generation



If a teacher's availability is **too restrictive** to allow their lessons to be positioned, generation fails with an explicit diagnostic. See [Diagnosing a failed generation](#).

Solutions:

- Relax the availability (turn a range red rather than black).
- Redistribute the courses (a teacher with 18 hours and only 12 free slots cannot make it work).
- Recruit (extreme case — but sometimes it is the right diagnosis).

☰ How-to

Collect and validate teacher availability

1. **Availability** tells the solver which time slots are **impossible** (black — hard constraint), **undesirable** (red — soft) or **preferred** (green). The solver strictly respects black and optimizes the rest.
 2. **Enable the entry mode** in the general settings (Teachers' availability setting): **disabled**, **weekly** (availability on a typical week, the classic mode), or **calendar** (on Premium accounts, availability date by date — combinable with weekly). Choose according to your timetable type.
 3. **Open the entry** to teachers. Two options:
 - they sign in and enter it from their accounts  ;
 - you generate a **direct share link** through  Download with an expiration date — no need for the teacher to sign in to their account — and you send it to each person concerned.
 4. **On the teacher's side**: they sign in or click the link, see their grid, mark slots black (impossible), red (undesirable), green (preferred). Saving is immediate. The whole action is quick.
 5. **Validation on the administrator's side (weekly mode)**: entered availability arrives as **pending validation**. Check then validate — you can edit along the way. Any later change by the teacher goes back to pending. In calendar mode, everything is consolidated in real time.
- ⚠ If a teacher marks too much black, the solver will not be able to position their lessons — the diagnostic will report it (see [Diagnosing a failed generation](#)).

🔗 See also

[Time constraints \(general system\)](#)

[Teachers' availability \(wishes\)](#)

[Incompatibility \(between subjects\)](#)

[Solver](#)

[Calendar-mode availability](#)

2.14 Time constraints: classes, subjects, groups, classrooms and grid

Source: [help/en/core-concepts/time-constraints.md](#) · id: [core-concepts.time-constraints](#) · Audience: [admin](#) · Updated: 2026-06-27


Time constraints tell the solver which slots to **avoid** or to **favor**. The same editor is used for a **teacher's availability** and for the constraints carried by a **class**, a **subject**, a **group**, a **classroom** or the **grid** itself. Each surface only exposes the levels that make sense for it.

For the details of the teacher case (entry by the teacher, sharing link, validation), see [Teacher availability](#).

The four levels

A slot is painted with one of these levels, each identified by a color:

Color	Level	Effect on generation
● Black	Unavailable	Hard constraint: the solver never places a lesson there
● Red	Undesired	Strong soft constraint: the solver avoids it as much as possible
● Orange	Slightly disliked	Light soft constraint: avoided if everything else allows it
● Green	Preferred	Positive preference: the solver favors this slot

An unpainted slot is **neutral**. The  Eraser clears a painted level and returns it to neutral.

Who carries which levels

Not all entities give access to all four levels:

- **Teacher** — Unavailable, Undesired, Preferred (black, red, green). This is their **availability**.
- **Class** (🕒 Class time constraints), **subject** (🕒), **classroom** (🕒 Availability) — Unavailable and Undesired (black, red). They describe the slots where you do not want — or would rather not have — the class working / the subject placed / the classroom used.
- **Group** (🕒 Availability) — the same black and red levels, a Premium feature (see below).
- **Time grid** (the site) — the **only** surface that offers the **orange** level (Slightly disliked). See below.

Access is always through the **small clock** 🕒 shown on the entity (on each class, each subject of the class, each classroom...). The clock takes on a **color** when constraints are already entered.

Constraints set on the grid

On a site's grid, you paint the **global constraints** directly, valid for the **whole school**:

- painting a slot **black closes** it for everyone — this is how a slot is banned globally (Wednesday afternoon, a break, a meeting slot...).

PREMIUM

With Premium, the grid additionally accepts **soft** constraints: **red** (Undesired) and **orange** (Slightly disliked), to discourage a slot without closing it.

See [Time grid](#), [time slots and durations](#) for the structure of the grid (slots, durations, breaks).

The constraint editor

The editor is shared by all entities, and everything is done by clicking.

First choose a level in the top bar (the levels offered depend on the entity). It becomes the active "brush" and applies to every slot you touch:

- **Click a cell** to paint that single slot; **click then drag** across several cells to paint them in one gesture.
- **Click a day header** to paint **the whole column** (the entire day); **click an hour header** to paint **the whole row** (that slot on every day). This is the fastest way to cover large areas.
- The 🧼 Eraser is a brush like any other: select it, then click or drag to return to neutral.

Typical week and calendar view

For a weekly or cyclic timetable, constraints are entered on a **typical week**.

PREMIUM

For a calendar-type timetable (Premium), a navigation bar lets you switch between **weeks**, **months**, **whole year** and **typical week**: you enter dated constraints, and you can also declare a recurring constraint on the typical week. The [date windows](#) additionally restrict the periods when certain lessons can be scheduled.

On the **typical week**, each slot **consolidates** the constraints of all the matching dates in the period (every Monday at 8:00, for example). When these dates diverge, the slot displays a **proportional color band** — the width of each color reflects the number of days concerned — and its **tooltip** gives the count per level (how many days unavailable, undesired, etc.). **Clicking this slot forces the chosen level on all these dates** at once: this is the generic way to decide for the whole typical day.

Additional options

PREMIUM

Below the editor, a class or group constraint can carry **caps** (Premium):

- [Maximum number of lesson hours in a day](#) — in HH:MM format;
- [Maximum number of lesson hours in a week](#) — in calendar mode.

The solver respects these caps as optimization constraints.

On the teacher side, a free **comment** lets them share a remark about their availability (travel, a specific constraint).

Inheritance between levels

PREMIUM

A **group** inherits the constraints of its **class** and of its **parent groups** (see [Group hierarchy](#)). When a slot must be locally **allowed** despite an inherited constraint, the **Allowed lessons** level neutralizes the inherited constraint **without changing the source**.

Effect on generation

Black constraints are **blocking**: a lesson will never be placed on an unavailable slot. The red and orange levels are **penalties** the solver tries to minimize; green is a **bonus**. If the constraints make a lesson impossible to place, the generation reports it — see [Diagnosing a failed generation](#).

See also

[Teacher availability](#)

[Group hierarchy](#)

[Time grid, time slots and durations](#)

[Date windows](#)

[Teachers' availability \(wishes\)](#)

2.15 School year, alternate weeks, holidays

Source: [help/en/core-concepts/school-year.md](#) · id: [core-concepts.school-year](#) · Plan: [standard](#) · Updated: 2026-05-12

The **school year** is the time frame within which timetables run. It raises three structuring questions: when the year starts and ends, when the holidays are, and — if the institution uses alternate weeks — how their alternation aligns with this calendar. It is also called the academic calendar.

School year and boundaries

A school year has a **start date** and an **end date**. Over this range, the weeks are numbered. Publishing a timetable applies to a subset or all of these weeks. A calendar-type timetable can even be published over a period longer than the school year (typically an 18-month EMBA), but only the dates declared on the school years are taken into account.

Without an active school year, Omniscol does not deploy the daily modules (timetable, dashboard, absences) — it is a prerequisite. See [School year and holidays](#).

School years cannot share dates. Usually, a holiday period separates them, but they can perfectly well run back to back. Some training institutions are not bound to an academic cycle: the school year is then simply the calendar year, from January 1 to December 31.

Holidays

Holidays are **inactive** date ranges: no lessons, no events, no hour counting. Two sources:

- **Common holidays** — taken from the country's official calendar (All Saints', Christmas, winter, spring). Omniscol can suggest them based on the configured country; you choose which ones to import.
- **Specific holidays** — unique to the institution (staff training day, exams, local bridge day). You add them by hand.

Both types combine without distinction. However, holidays are global to the whole institution. If only some classes are concerned, use a class absence, or a class unavailability period (potentially reusable across several classes, for example classes that are abroad).

Alternate weeks (A/B and beyond)

Some schools have courses that run only every other week (or even less often): PE as a double lesson, homeroom on alternate weeks, options. Omniscol handles this by **counting the weeks**: the weeks of the year are not labeled, they are counted continuously. Each course runs on its own cycle — every other week (A/B), one week in three (A/B/C), and so on. The same time slot then hosts different lessons depending on the week's rank in the cycle, for example one lesson on odd weeks and another on even weeks, on the same day and at the same time. Since the computation is dynamic, A/B courses and A/B/C courses can coexist in the same class.

The **activation** of alternate weeks and the **display format** of the alternation (A/B , 1/2 ...) are set in the [general settings](#). The [school year](#) screen is then used to **force, on the timeline, the points where the count restarts at week A** — for example to resume cleanly after holidays; to stay readable, the timeline only displays the alternation as A/B. To **declare alternate lessons** on a time slot, see [Complex lessons](#).

Several years in parallel

Omniscol manages **several school years in parallel** in the same account: for example the current year 2026-2027 and the next year 2027-2028 in preparation. The notion of the **current school year** determines what users see day to day; the others remain accessible for administrative management / viewing.

See also

[School year](#)[Alternate lessons](#)[School year and holidays](#)[Publishing a timetable](#)[Complex lessons](#)

2.16 Timeline and time navigation

Source: [help/en/core-concepts/timeline-navigation.md](#) · id: [core-concepts.timeline-navigation](#) · Updated: 2026-05-12

Every Omniscol screen that displays a schedule or statistics carries a **timeline** (the "banner") at the top of the page. It is how you **navigate through time**: choosing the school year, the displayed period, and — on some screens — switching from a week view to a month or year view.

This page explains the shared principles; each module has its own specifics, covered on its dedicated page.

Timeline buttons

The timeline exposes **left / right arrows** on its two opposite sides:

- the **left arrow** ← goes back to the **previous school year**,
- the **right arrow** → moves forward to the **next school year** (the next school year being prepared, for example).

The year (or period) **currently displayed** is usually named explicitly above the timeline.

The classic oversight: the right arrow

The **Timetable management** module displays **the current school year by default**. When a planner prepares the next school year, they must **explicitly click the right arrow** of the timeline to switch to the next school year before they can distribute / publish their timetable on it. On this specific screen, you can also move from one year to another by clicking the school year. The current one is shown in green, the next one in orange.

Time views by module

Not all modules expose the same granularities:

Module	Default view	Other views
Timetable	Week	Day, month
Dashboard	Week	Month, year, free date-to-date range
Absence management	Week	Month, year, free date-to-date range
Staffing	Week	Day
Timetable management	Entire school year	
Editing a calendar timetable	Week	Month
Calendar-based availability	Week	Month, school year, typical week

The **view toggle buttons** sit next to the timeline, usually above it on the left. The `year` and `free date-to-date range` views are mostly found in the **Dashboard**, where they serve cross-period statistics.

School year displayed by default

The **current school year** (configured in [Administration](#)) is **displayed by default** in every module for every user.

- When viewing their timetable, teacher and student accounts see **only the current year**. No access to past or preparatory years on the portal side. This is deliberate: for them, the timetable is about today.
- **Absences** can be declared **only for the current school year**. You cannot retroactively declare an absence for a past year, nor declare one in advance for the next year.
- **Administrators**, however, can **navigate** between years through the timeline — hence the value of understanding the arrows.

The current school year is also the one used for exports and synchronization with external software.

When you switch the current school year

Typical sequence:

1. **Year N in progress**. You work in parallel on year N+1 in [Timetable management](#), through the right arrow. All users still see year N on their portals and their absences.
2. **Final preparation** for the start of year N+1: the N+1 timetable is published on the corresponding weeks (but the current year remains N).
3. **Just before the school year starts**, when you are ready for everyone to switch to N+1: change the current school year in [Administration](#) → [School year](#).
4. **Immediate effect**: every user sees N+1 by default from that moment on. The teacher and student accounts access their new timetable. Absences are declared on N+1.

The exact moment of the switch depends on your institution — often **the day before school starts**, once everything is validated, sometimes **the morning school starts** to keep a margin for checks. No rush: you keep read access to the past year as history.

School year ≠ calendar year

A **school year** in Omniscol does not necessarily run September → August. Frequent special cases:

- **Continuing-education providers** — often aligned with the **calendar year** (January 1 → December 31). Define your school year with these dates in [Administration](#) → [School year](#); everything works the same.
- **Universities** with shifted semesters or long cycles (intensive modules over 18 months) — you can define school years of non-standard length.

See also

School year

School year and holidays

Publishing (activating) a timetable

Preparing the next school year

2.17 Search and filter in lists

Source: [help/en/core-concepts/search-and-filter.md](https://help.en/core-concepts/search-and-filter.md) · id: `core-concepts.search-and-filter` · Audience: `admin/teacher/staff` · Plan: `standard` · Updated: 2026-06-27

The same search bar appears in most lists in Omniscol (classrooms, teachers, classes, subjects, resources, students...): it combines a **text** search and, wherever a numeric value exists, a **numeric** condition. On **classrooms**, this condition applies to the **capacity** — the most complete use, and the one detailed here. You will find this bar in the `Classrooms` tab of the Dashboard as well as in the **classroom selector** of a lesson, when you assign or change a classroom in the timetable (`Assign a classroom`).

Text search

Text search works on **all these lists**. On classrooms, it covers the classroom name, the site, the building, the specialisation or the tags entered in Omniscol; on other lists, the displayed fields (name, code, etc.).

Examples:

- `nord` displays the classrooms whose information contains `nord`.
- `laboratoire chimie` displays the classrooms that match both terms.
- `amphi, gymnase` displays the classrooms matching `amphi` or `gymnase`.

Capacity filter (classrooms)

On classroom lists, the numeric comparators filter on the classroom capacity:

- `>50` : capacity strictly greater than 50;
- `>=50` : capacity of at least 50;
- `<25` : capacity strictly less than 25;
- `<=100` : capacity of at most 100;
- `=30` : capacity exactly equal to 30;
- `50-100` : capacity between 50 and 100;
- `50-` : capacity equal to or greater than 50;
- `-25` : capacity equal to or less than 25.

Useful combinations

Spaces add criteria that must all be met. Commas separate alternatives.

- `>50 site nord` : classrooms with more than 50 seats linked to the north site.
- `<=30 laboratoire` : small laboratory-type classrooms.
- `amphi, >100` : classrooms containing `amphi` or classrooms with more than 100 seats.
- `bâtiment A informatique` : classrooms linked to building A and to computing.

Why keep classroom data complete

Filters become more precise when sites, buildings, specialisations and tags are kept up to date. This information improves searches in the timetable, classroom statistics and building occupancy analyses.

See also

[Classroom statistics](#)

[Using tables and charts](#)

[Sites, classrooms and resources](#)

2.18 Collaboration between administrators

Source: <help/en/core-concepts/collaboration.md> · id: [core-concepts.collaboration](#) · Audience: [admin](#) · Feature: [collab](#) · Updated: 2026-06-13

PREMIUM

The **Real-time collaboration** option lets several administrators work simultaneously on the same Omniscol account: each one sees the others' presence live, and the server merges concurrent actions.

Real-time presence

At the top of the screen, an indicator shaped like a **disc with initials** appears for each other administrator signed in to the account at the same time as you (yours is light blue, and the discs overlap when many of you are connected). Hover over them: a tooltip shows each person's name and **the screen they currently have open**.

An indicator turns **red** when another administrator opens the **same editing screen** as you — timetable construction (sites, classes, groups, hours distribution), reorganization, staffing, users, subjects or settings. It signals a **risk of simultaneous modification**.

Nothing blocks you, though: the server **merges concurrent actions** as far as possible, so that everyone keeps their changes without any silent overwrite. However, editing exactly the same field of a user or a lesson, or importing bulk data, inevitably leads to a conflict: the last change wins. The red indicator is an invitation to coordinate so you do not work concurrently in the same place, on the same data, at the same time.

When to use it

- **Large school with several managers** — each one handles a department, a cohort or a site, but all of them work in Omniscol at the same time.
- **Busy periods** — building the start of the school year, switching to an exam session, handling a wave of absences.
- **Coordination during a meeting** — the team gathers around a decision and several participants make changes from their own computers: the presence discs make it easy to divide up the areas.

Synchronization limit

The presence discs are relayed in real time. On the other hand, the working view another user has open is not necessarily refreshed automatically after you save. If a colleague has just changed a piece of data you are viewing, reload the view or reopen the object before making a decision based on it.

Best practices

- **Define scopes:** who handles what. The scope can be informal (by e-mail), or formalized through [custom roles](#).
- **Communicate outside the app** — keep a separate coordination channel for business decisions; the presence indicators are no substitute for work instructions.
- **Avoid editing the same structure at the same time:** if two of you touch the student list of a class at the same moment, the risk of conflict increases. Split the subtasks by scope.

☰ How-to

Coordinate a school-year start with three administrators

1. **In large schools**, building the start of the school year involves several administrators in parallel. The **Real-time collaboration** option adds real-time presence and the merging of concurrent actions.
2. **Define the scopes** beforehand (in a meeting or by e-mail): who manages which department / cohort / site. You can complement this organization with **custom roles**, which restrict the modules and operations available to each account; the split by department or by site remains a team convention.
3. **Everyone signs in** to the Omniscol account. At the top of the screen, **the indicators with the initials** of the other connected administrators appear in real time. On hover, each person's name and current screen.
4. **Conflict signal**: an indicator turns **red** when two administrators open the same editing screen (the same classes, the same reorganization...). It signals a risk of simultaneous modification, visible to all the administrators involved.
5. **The server merges** concurrent actions: nothing blocks you while editing, and everyone keeps their changes, as far as possible. The red indicator invites you to coordinate before touching the same place.
6. **Best practices**: keep a separate coordination channel and split the subtasks by scope (one administrator per department). Avoid having two people edit the same structure (a class's student list, for example) at the same moment.

After a change made by a colleague, reload the affected view before treating the displayed data as up to date.

📌 See also

[Omniscol plans and options](#)

[Users and roles](#)

[Custom roles](#)

3. Building a timetable (Timetable management module)

3.1 Overview of the Timetable management module

Source: [help/en/timetables/overview.md](#) · id: [timetables.overview](#) · Audience: [admin](#) · Updated: 2026-06-13

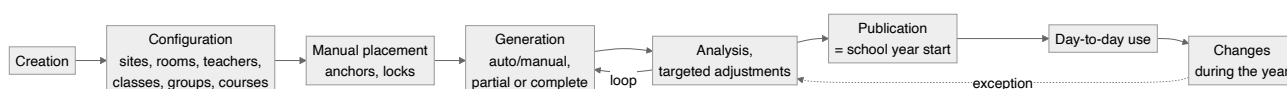
The **Timetable management** module is the heart of Omniscol: this is where timetables are created, configured, generated and published.

For a recurring timetable (weekly/cyclic), it is used **occasionally** (typically before the start of the school year, when creating a different term or semester, and for structural changes), not daily (for that, use the [Timetable](#) module).

For a calendar-type timetable, in higher education, it is conversely the main working screen of the whole planning team, every day.

Planning flow

Building a timetable is not a straight line: after creation and configuration, the core is a **cycle** — place a few lessons, generate, analyze, adjust, loop again — before a linear sequence that starts at **publication** (the start of the school year).



You can go back at any time, save along the way, duplicate a timetable to create a variant. The heart of the cycle is detailed in [Automatic generation](#) and [Manual placement](#); after the school year starts, touch-ups go through [Ad-hoc changes](#).

Several timetables in parallel

You can have several timetables in your account:

- unpublished **drafts** (to prepare the next school year while the current year is running),
- timetables for different **periods** (S1, S2...),
- with Premium, timetables **active simultaneously** on the same weeks (for example, a weekly morning timetable + a calendar afternoon timetable) — see [Multiple active timetables in parallel](#). This possibility can also be opened contractually on some Standard accounts.

Detailed steps

Each one has its own dedicated page:

The **prerequisite**: [Prerequisites for creating a timetable](#). Then, eight steps:

1. [General settings](#)
2. [Sites, time grids, classrooms, resources](#)
3. [Assigning teachers](#)
4. [Creating the classes and their groups](#)
5. [Alignments and groups of groups](#)
6. [Distribute the hours and create the lessons](#)
7. [Automatic generation](#)
8. [Publishing \(activating\) a timetable](#)

At any time, you can:

- **View** a timetable (read-only mode), for checking.
- **Reorganize** a timetable (quick edit mode), to adjust lesson positions.
- **Duplicate** a timetable (create a draft from an existing timetable).

- **Import** or **export** the course configuration as a spreadsheet (see [Mass import](#)).

Quick start

If this is your first timetable:

1. First check that your [account is set up](#) (settings, school year, subjects, users).
2. Click [Create timetable](#) at the top right.
3. Choose the [timetable mode](#) (weekly, cyclic, calendar).
4. Follow the steps in order.

If you already have data in a spreadsheet (Excel, Google Sheets, an export from another program), go directly through [mass import](#): Omniscol creates the structure for you, and you finalize afterwards.

See also

[Choosing the right timetable type](#)

[Preparing your data for a mass import](#)

[Higher education use cases](#)

[Special cases and advanced configurations](#)

3.2 Prerequisites for creating a timetable

Source: [help/en/timetables/prerequisites.md](#) · id: [timetables.prerequisites](#) · Audience: [admin](#) · Updated: 2026-05-10

Before creating a timetable, three elements must exist at the school level:

1. At least one school year

Without a school year, there is no timeline, and therefore no publication possible. Created in [School years](#).

See [School year and holidays](#).

2. The subjects you use

Omniscol pre-fills the base of common subjects for your country. For specific disciplines, create the **custom subjects** in [+ Create](#).

△ Mind the spelling: Omniscol makes an internal copy when a subject is assigned to a timetable, and a fix does not propagate to timetables already published in the present or the past.

See [Managing subjects](#).

3. The teachers

Create all teachers in Administration > Teachers (one by one or in bulk via [import](#)). Fill in at least first name, last name and e-mail. The subjects taught and the service hours are optional but useful.

See [Managing teachers](#).

Optional but useful

- **Students** — not needed to create the timetable, but useful for individual timetables after publication.
- **Course types** (tutorial, practical, lecture...) — if your teaching nomenclature uses them, create them in [+ Create](#).
- **Levels** — check that Administration > Settings contains the class levels you will use.

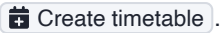
You can start even without everything

Omniscol is **flexible**. If you do not have all the teachers or all the subjects when creating the timetable, you can start and fill in the rest along the way. It is less linear, but it works.

However, the **school year** really is essential before publication.


Getting started

Once these prerequisites are (at least partially) in place:



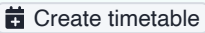
1. Go to the Timetable management module.
2. Click .
3. Choose the [timetable mode](#).
4. You land on the [general settings](#) page.
5. Give it a label (e.g. "Timetable 2025-2026", "S1 2025") and a description.

How-to

Checking the prerequisites before creating a timetable

1. **Three elements must exist at the school level** before a timetable can be created: a school year, the subjects, the teachers.
2. **School year**: open School years. Check that at least one year is defined, with its start and end dates and the holidays. Without a year, no timeline, no publication. See [School year and holidays](#).
3. **Subjects**: Omniscol pre-fills the base for your country. If your institution has specific subjects (Home economics, Robotics, Advanced Mandarin), create them via .

△ Mind the spelling: Omniscol copies the name into each timetable, and a fix after creation will not propagate to publications of the past and the present.

4. **Teachers**: open Teachers. For each teacher, at minimum first name, last name, e-mail. Subjects taught and service hours are optional but useful. For many teachers: mass import (see [Preparing your data](#)).
5. **Optional but useful**: course types (tutorial, practical, lecture...) via  if your nomenclature uses them. Class levels to check in the general settings ().
6. **You are ready**: **Timetable management** module,  button. Choose the mode ([timetable modes](#)), give a label (Timetable 2026-2027, S1 2026) and a description. Next step: [general settings](#).

See also

[Set up the school account](#)

[Next step — General settings](#)

3.3 Calendar mode — advanced options

Source: [help/en/timetables/calendar-mode.md](#) · id: [timetables.calendar-mode](#) · Audience: admin · Plan: premium · Updated: 2026-06-13

PREMIUM

PREMIUM

Calendar mode is one of the three timetable types supported by Omniscol, alongside weekly mode and cyclic mode. It is used when the timetable does not follow a regular recurrence: each lesson has a precise date, with no repeating weekly pattern.

Creating a calendar timetable is included in the Premium plan, along with [availability in calendar mode](#) and the publication of [multiple active timetables in parallel](#): on a Premium account, these functions are available without any specific activation.

What calendar mode specifically enables

- create a timetable whose lessons are individually dated;
- define a start and end date range;
- use date constraints when they are configured;
- generate only a target window shorter than the timetable's total range;
- ask the solver to compact lessons at the start of the period, at the end of the period, or with no position preference;
- display, publish and view lessons like other timetables but over a date interval;
- use off-grid mode on calendar classes, with explicit start and end times on each lesson.

Complementary functions

Other functions, described on their dedicated pages, naturally complement a calendar timetable:

- [Multiple active timetables in parallel](#) to publish several timetables over the same period;
- [Availability in calendar mode](#) to collect dated availability;
- [Groups of groups](#) to work with transverse groups.

When to prefer calendar over weekly

- **Non-recurring programmes:** intensive modules, seminars, conferences on varying dates.
- **Continuing education:** sessions that do not run on a regular academic year.
- **External teachers:** each one has their own calendar, not fixed weekly service hours.
- **End-of-programme courses:** projects, thesis defences, juries.

See [Calendar mode](#) for typical higher-education cases.

☰ How-to

Creating a calendar timetable

1. **Create a new timetable** in Timetable management. On the type selection screen, choose **calendar mode**. Enter the start and end dates, then confirm.
2. **Configure the calendar:** label, description, working days and time structure. You can also declare reusable **date windows**, applied on the time constraints of classrooms, classes, subjects and groups.
3. **Enter the lessons** on precise dates: date, times, group or class, teacher(s) and classroom(s). If a class is off-grid, fill in explicit start and end times.
4. **Start the generation** if you use the solver on this timetable. The available constraints depend on the data filled in: teachers, classrooms, dates, availability and calendar constraints. For a test or a progressive build, use the advanced options to limit the generated date window; enable compacting if you do not want the algorithm to spread lessons across the whole available period.
5. **Publish and combine:** you can publish this calendar alongside a regular weekly timetable. The people concerned see the merged lessons when viewing. See [Multiple active timetables in parallel](#).

🔍 See also

[Calendar mode](#)

[Choosing the right timetable type](#)

[Availability in calendar mode](#)

[Date windows](#)

[Multiple active timetables in parallel](#)

[Calendar mode for non-recurring programmes](#) {append=(higher education)}

3.4 Availability and constraints in calendar mode

Source: help/en/timetables/calendar-wishes.md · id: [timetables.calendar-wishes](#) · Audience: [admin/teacher](#) · Plan: [premium](#) · Updated: 2026-06-20

👑 PREMIUM

PREMIUM

In **calendar mode**, planning happens on **specific dates** rather than on a recurring typical week. This changes how **time constraints** are expressed, on two distinct levels:

- the **internal constraints** of classrooms, classes, subjects and groups;
- **teacher availability**.

In calendar mode, we speak of **availability**: the term *wishes* stays reserved for a teacher's input on a typical week, upstream of planning, and subject to the planner's approval.

Time constraints of classrooms, classes, subjects and groups

As soon as a timetable is of calendar type, the **internal** time constraints of these entities are automatically expressed **per date**, rather than on a typical week. You can additionally set a **typical-week recurrence** when a constraint repeats identically from one week to the next.

For reusable periods — exam sessions, holidays specific to certain classes, allowed slots — apply a [date window](#) as an overlay on these constraints.

Teacher availability

On a typical week, a teacher declares a constant rule — *“I am not available on Wednesday afternoons”*. In calendar mode, availability is declared **date by date**, which captures finer situations:

- *“I am unavailable only on the morning of Wednesday, March 12”*,
- *“I can teach a lesson only between January 20 and January 30”*,
- *“I am available 3 half-days per week, to be chosen depending on the week”*.

This is the mode to prefer when availability varies from week to week — typical of higher education, adjuncts and continuing education.

Setting the input mode

On the teacher side, set the [Teachers' availability](#) setting (school general settings) to **calendar** or **calendar + weekly** to match planning on specific dates. The combined mode additionally lets the teacher simply state that they are **never available on a given day of the week**, whatever the date (one-off exceptions remain possible) — the teacher-side equivalent of the typical-week recurrence of internal constraints.

Availability levels

The input screen uses a **calendar view**. The planner or the teacher clicks the relevant dates and qualifies the slot:

- **Unavailable**: blocking unavailability — generation will not place any lesson there;
- **Undesired**: strong constraint, to be avoided as much as possible;
- **Preferred**: positive preference;
- **Eraser**: clears the availability already set on the covered area.

Input can be done by day, half-day or time slot, depending on the granularity configured for the school.

Who this is for

- **Adjuncts and external instructors** — do not have a fixed weekly service. See [External instructors \(adjuncts, visiting faculty\)](#).
- **Teachers on medical part-time or a specific accommodation** — variable weeks.

- **Continuing education courses** — each session is a one-off.
- **Teachers in a partial mobility year** — available certain weeks only.

☰ How-to

Collecting an adjunct's availability on the calendar

1. **For an adjunct who does not have a fixed weekly service** (an external instructor for example), typical-week input is not enough. **Calendar** mode captures availability date by date.
2. **Enable the appropriate mode:** set the **Teachers' availability** setting (school general settings) to **calendar** or **calendar + weekly**, depending on the account's needs.
3. **Open the input** to the adjunct. Either they sign in with their teacher account, or you generate a **direct share link** to their availability screen (see **Teacher availability**). The direct link avoids creating a dedicated account when this flow is used. Any administrator can also enter the availability themselves.
4. **On the adjunct's side:** they see a **yearly calendar** rather than a weekly grid. They click the dates (or half-days, or time slots depending on the school's granularity) and qualify the area with the proposed availability levels.
5. **Check the consolidation** on the administration side. The recorded availability is read back by the timetable screens and by the diagnostics that consult constraints.
6. **Run automatic generation** on the adjunct's lessons: the solver combines the availability priorities with the lesson's other constraints. Use cases covered: adjuncts, medical part-time, continuing education, teachers on partial mobility.

🔍 See also

[Teacher availability](#)

[Calendar mode](#)

[Date windows](#)

[Choosing the right timetable type](#)

3.5 Date windows

Source: <help/en/timetables/date-windows.md> · id: `timetables.date-windows` · Audience: `admin` · Plan: `premium` · Updated: 2026-06-20

👑 PREMIUM

PREMIUM

A **date window** is a **reusable date inclusion** or **exclusion** period that restricts the periods during which certain lessons can be scheduled. You define it once, then apply it as an **overlay** on the time constraints of several entities — classes, groups, subjects or classrooms.

This feature only applies to **calendar-type timetables**.

Two types: allowed or excluded

- **Allowed lessons** — lessons can be scheduled **only** during these dates.
- **Excluded lessons** — lessons **cannot** be scheduled during these dates.

How they combine

Several windows can apply to the same entity:

- allowed windows are cumulative;
- excluded windows are cumulative;
- where they overlap, **an exclusion takes precedence over an authorisation**.

Typical use cases

- **Classes on the same work-study rhythm** — several classes take certain courses only on specific weeks (or only at the beginning or end of weeks). An *allowed* window describing these weeks, applied to all those classes, avoids re-entering the rhythm class by class.
- **Exam periods** — an *excluded* window covering the exam session temporarily blocks the classrooms concerned, or the classes, so that no ordinary lesson is scheduled there.
- **Holidays specific to certain classes** — on a campus abroad, a set of classes follows a holiday calendar different from the rest of the institution. An *excluded* window on these dates, applied to those classes only, neutralises the period without affecting the others.

Defining a date window

On the edit screen of a calendar timetable, in the Date windows section:

1. **+ Add** opens the date picker.
2. Give it a **label** and choose the type — allowed or excluded.
3. Select the dates on the calendar grid: click and drag, whole week, single-day column, even / odd week shortcuts, or selection inversion.
4. Save: the window joins the timetable's reusable list.

Applying a window

Once defined, the window is available in the time constraints of **classes, groups, subjects and classrooms**: select one or more windows there. The same window can serve as many entities as needed — that is the whole point of the reusable overlay.

At generation

The solver treats the windows as constraints: it only places a lesson within the allowed dates, and never within the excluded dates.

See also

[Calendar mode](#)[Availability in calendar mode](#)[Teacher availability](#)[Automatic generation](#)[Manual placement](#)

3.6 Step 1 — General settings

Source: help/en/timetables/general-settings.md · id: [timetables.general-settings](#) · Audience: *admin* · Updated: 2026-05-18

On the **General** tab, you set the global parameters that structure the timetable.

The pre-filled values come from the [school settings](#), which you can also adjust beforehand.

Fields to fill in

1. **Label** — the name of the timetable ("Timetable 2025-2026", "S1 2025"...). Visible everywhere. Editable.
2. **Description** — free text, useful when you have several timetables.
3. **Display mode** — by default, one row per time slot with the times on the left. You can choose a numbered-period style, common in English-speaking schools, or a calendar display on round hours.
4. **Base lesson duration** — the reference duration of a time slot, possibly including the gap between lessons. If your lessons last 55 min with a 5-min gap, enter **60 min**. If your regular lessons last 1.5 hours, enter **90 min**.
5. **Lesson unit division** — into how many sub-parts can the base duration be cut? To be able to create lessons of 1 hour, 1.5 hours or 2 hours with a base duration of 60 min, choose **1/2**. To go down to the quarter-hour, choose **1/4**.

- **Tip:** do not go finer than necessary. The finer the division, the longer the solver computation.
- 6. **Working days** — tick the days with lessons. By default Monday to Friday (unless the country settings differ — Arab countries, Israel...).
- 7. **For cyclic timetables** — the number of days in the cycle.
- 8. **For calendar timetables** — start and end dates (editable later) and optionally date windows.

Saving

The button at the bottom saves the configuration.

What next

Once these parameters are validated, move on to the next step: [Sites, rooms, resources](#).

Tip. If you are migrating from another system or simply want to try Omniscol quickly, for large mass creations of lessons (hundreds at once), the [Mass import via spreadsheet](#) is faster than screen-by-screen entry. However, the more structural data you enter beforehand, the better the import.

☰ How-to

Configure the general settings of a timetable

1. **Step 1 of timetable creation:** set the label, base lesson duration, lesson unit division and working days. These choices shape the timetable for the long term — best to get them right from the start.
2. **Label and description:** a meaningful name (`Timetable 2026-2027, S1 2026 - M2 business cohort`) and a free description. Visible everywhere in the interface. Editable, but avoid renaming in production.
3. **Display mode:** one row per time slot (default) or numbered periods (common in English-speaking schools) or a calendar on round hours. Follow the culture of your institution.
4. **Base lesson duration:** the reference duration of a time slot. 55-min lessons with a 5-min gap → **60 min**. Regular 1.5-hour lessons → **90 min**. If you keep a base duration of 60 min, the lesson unit division still allows 1.5-hour lessons.
5. **Lesson unit division:** into how many sub-parts can the base duration be divided? **1/2** for 30-min steps with a base duration of 60 min, **1/4** for 15-min steps. The finer it is, the more possibilities the solver has to test.
6. **Working days:** tick Monday to Friday (default). Arab countries, Israel: Sunday to Thursday. For **cyclic** timetables: the number of days in the cycle. For **calendar** timetables: start and end dates (editable afterwards).
7. **Save** via . Next step: [Sites, rooms, resources](#). For large mass creations, see [Mass import via spreadsheet](#).

🔗 See also

[Time grid, time slots and durations](#)

[Previous step — Prerequisites](#)

[Next step — Sites](#)

[General school settings](#)

3.7 Step 2 — Sites, time grids, classrooms, resources

Source: [help/en/timetables/sites-rooms.md](#) · id: [timetables.sites-rooms](#) · Audience: admin · Updated: 2026-06-13

The **Sites** tab is where you model the physical dimension of your institution.

Creating the sites

If all your lessons take place in the same location: one site is enough. Click [+ Create a site](#) and give it a name.

If you have multiple sites: create one site per distinct physical location. Enter the **travel times** between sites (matrix of durations). This is essential — without these times, the solver may place a teacher from 9:00 to 10:00 on site A and at 10:05 on site B, 5 km away, which is physically impossible.

Do not confuse site and campus: a **campus** is an organisational grouping configured in the general settings; a site carries the time grid, the classrooms and the travel times.

See [Site](#) for the use cases and limits (two virtual sites for the same location, choosing between multiple sites and multiple accounts, etc.).

Configuring the time grid

Each site carries its own **time grid**:

- start / end times of the time slots,
- breaks,
- lunch break,
- closures (Wednesday afternoon, Saturday...).

Standard lessons are attached to the time slots of this grid. If you later change the time of a slot, the lessons placed on it follow the new grid. Custom times and off-grid classes are special cases described in [Time grid, time slots and durations](#).

Omniscol guesses the grid automatically from the information you enter, but review it and customise it — this matters for the display and for the algorithm.

The time slots should match the base lesson duration entered on the previous screen. If you specified 60 minutes, for example, for a standard lesson, then each slot will be considered to last 1 hour, regardless of the times actually entered — sometimes 50 or 55 minutes, this can vary slightly because of breaks. If the gap between theory and practice is too large, on a given slot, its times may turn red to warn you of a bad configuration.

Do not declare breaks as time slots! Any slot can host lessons (unless the period is marked as unavailable). A frequent mistake is to believe that the times have a direct impact: this is not the case — Omniscol places lessons on time slots, then displays them with the times entered for each slot concerned.

Entering the classrooms

For each site, create the **classrooms**. Fields:

- **Name** (required),
- **Capacity** (required — without this field, the solver cannot check the sizing),
- **Specialisations** (chemistry, IT, sports... free-text),
- **Maximum number of classes for a large room** (exam room, theatre, gymnasium, swimming pool, outdoors). This field only appears once a specialisation has been entered,
- **One or more campuses** if the classroom should be preferred for a department, a faculty or a school in the automatic assignment, and for certain filters,
- **Building** (optional),
- **Availability** (useful for classrooms shared with another institution, or outdoor sports facilities),
- **Tags / comment** (flip chart, power outlets, computers... free-text).

Associating a classroom with one or more campuses remains a **distribution preference**. If no classroom of the campus is compatible or available, Omniscol may assign another one.

You can create classrooms **in bulk** by copying and pasting from a spreadsheet.

PREMIUM

Multi-room

Omniscol lets you assign several classrooms to the same lesson, with total capacity = the sum of the rooms. See [Multi-room](#) for the typical use cases (exams split across lecture halls, broadcast lectures, split lab sessions, etc.).

Specialisations

Specialisations (free-text fields) are used to indicate that a subject requires a particular type of classroom. The solver enforces them strictly.

Large room

A specialised classroom can become a **large room**: enter its **Maximum number of classes** so that it hosts several different lessons at the same time (distinct teachers and groups), within the limit of that number and of its capacity. This is the case for exam rooms, theatres, gymnasiums, swimming pools or outdoor spaces. The field only appears after the classroom has been given a specialisation.

Entering the resources




Resources are the mobile equipment not attached to a particular classroom (portable projectors, tablet carts, kits...). For each resource:

- **Name**,
- **Available quantity**.

A cart of 30 tablets counts as 1 (not 30) — you enter the number of carts.

There is no need to model resources you "always have enough" of — only do it for real shared limits.

PREMIUM

Sites, **classrooms** and **resources** can be synchronised from an external system — each list then offers its own synchronisation button ( Synchronize ,  Synchronize ,  Synchronize). See [Synchronization with external systems](#).

What next

Next step: [Assigning teachers](#).

☰ How-to

Creating a site and its time grid

1. **A site** models a distinct physical location, with its own time grid (time slots, breaks, lunch). At least one site is required. This procedure describes creating a site and its minimal dependencies (classrooms, resources).
2. **Click Create a site** and give it a name (`Bâtiment principal`, `Antenne Paris ...`). If you are starting with a single site, one site is enough. For multiple sites, add as many as there are physical locations to distinguish.
3. **Configure the time grid**: start / end times of the slots, breaks between lessons, lunch break, closures (Wednesday afternoon, Saturday...).
4. **Enter the classrooms**: name, capacity (a critical field for the solver), specialisations (chemistry, IT, sports...), opening hours, campus if your organisation uses them. The campus on a classroom serves to prefer it for certain classes, without blocking the other compatible classrooms. A classroom belongs to a single site — for a classroom used by several virtual sites, see [Sites, classrooms, resources — concepts](#).
5. **Enter the resources** (mobile equipment): name + available quantity. Only model resources with a real shared limit (portable projectors, a tablet cart, a microphone kit), not those you "always have enough" of.
6. **If you have multiple sites, enter the travel times** via the distances popup (triangular half-matrix). Without these times, the solver teleports — see [Multiple sites: policy and travel times](#) for the nuances.

🔗 See also

[Sites, classrooms, resources — concepts](#)

[Campus](#)

[Time grid, time slots and durations](#)

[Multi-room](#)

[Classroom specialisation](#)

3.8 Step 3 — Assigning teachers

Source: [help/en/timetables/assigning-teachers.md](#) · id: `timetables.assigning-teachers` · Audience: `admin` · Updated: 2026-06-13

On the **Teachers** tab, you indicate which teachers are **involved** in this timetable — as opposed to those simply declared at school level.

Assigning

Click [👤+ Assign teachers](#). The list of available teachers is displayed. Select the relevant ones.

You can **assign in batches**: start with the teachers already definitively hired, then add new ones as recruitment progresses.

PREMIUM

Teachers can be synchronized from an external system: [🔄 Synchronize](#) opens the matching screen. See [Synchronization with external systems](#).

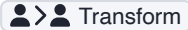
Virtual teachers

The [+ Virtual 👤](#) button creates a **virtual teacher** to temporarily represent a position to be filled. Convenient for preparing the start of the school year without waiting for recruitment to be finalized.

A 👤 icon appears next to the name. When recruitment is finalized, the [👤>👤 Transform](#) button at the end of the row turns the virtual teacher into a real one: choose the real teacher from the list, and all its references switch over to

them — the subject assignments in the classes as well as all the lessons. The virtual teacher then disappears for good. **The action is irreversible.**

Permanently replacing a teacher

The same  Transform button also applies to a **real** teacher: it transfers all of their courses to another teacher from the list — already assigned to this timetable or simply declared at school level — then removes the original teacher from the timetable.

This is the **permanent replacement** gesture, useful when a planned teacher drops out, is transferred or has to give up their courses: all lessons and assignments move to the replacement in a single operation. **The action is irreversible.**

Availability

If you have enabled availability entry via the [Teachers' availability](#) setting (see [Teacher availability](#)), this is where you **check and validate** the availability entered by teachers.

You can also:

- edit the **service hours** (number of planned weekly hours, carried over into new timetables),
- assign a **preferred classroom** (used as a priority by the solver for the teacher's lessons, unless the class has its own dedicated classroom).

School level vs timetable level


Important: the teacher record **at school level** carries the identifying information (name, e-mail, login) and remains stable. The teacher record **at timetable level** carries the information *tied to this timetable* (availability, preferred classroom, service hours).

Consequence: if you correct a teacher's name in Teachers after a timetable has been created, the timetable keeps the historical name. This is a deliberate choice for traceability, but it assumes the name is spelled correctly when the timetable is created.

External teachers (adjuncts)

For external teachers, see the [glossary](#). They are created like ordinary teachers.

PREMIUM

The **External teacher** marker on the user record adds a small  icon that distinguishes external teachers from permanent teachers on some screens.

For visiting professors who contribute occasionally (one or two lessons), you can, depending on the case:

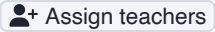
- add them as a co-teacher on a lesson (see [Co-teaching](#)),
- create a separate course for their contribution (calendar mode especially).

What next

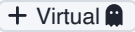

Next step: [Creating the classes and their groups](#).

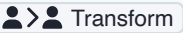
☰ How-to

Assigning teachers to a timetable


1. **Step 3 of timetable creation**: indicate which teachers are **involved** in this timetable (as opposed to all those declared at school level).
2. **Assign the teachers already definitively hired** via . The list of teachers declared at school level is displayed. Select those involved in this timetable.

You can **assign in batches**: start with the permanent teachers, add new ones as recruitment progresses.

3. **For every position to be filled**, create a **virtual teacher** via . A  icon marks it out. Convenient for preparing the start of the school year without being blocked by recruitment.

When recruitment is finalized, the  button at the end of the row switches all its references over to the real teacher chosen from the list. Lessons already positioned are kept. The same button is used for the **permanent replacement** of a real teacher by another (transfer, withdrawal...).

Specify the subjects taught by the virtual teacher on this timetable: it will be proposed first for those subjects during assignments.

4. **Availability**: for a weekly timetable, you **check and validate** the entries. For any timetable type, you can **view and fill in** all availability. See [Teacher availability](#).
5. **Preferred classroom** (optional): if the teacher has a dedicated classroom, assign it here via . The solver prioritises it for their lessons, unless the class already has its own dedicated classroom (class priority > teacher priority).
6. **Next step**: [creating the classes and their groups](#).

🔗 See also


[Teacher availability](#)[Managing teachers at school level](#)[Teacher](#)[External teacher](#)


3.9 Step 4 — Creating the classes and their groups

Source: <help/en/timetables/creating-classes.md> · id: [timetables.creating-classes](#) · Audience: admin · Updated: 2026-05-10

This is the most foundational step. On the **Classes** tab, you create the classes, their groups, and you enter the hours per subject.

1. Creating the classes

Click . For each class:

- **Name** (required),
- **Level** (from the existing levels; click  to add one),
- **Campus** if your institution uses this structure,
- Default **site** (the solver will only search among this site's classrooms, unless a lesson is explicitly assigned to another site),
- **Dedicated classroom** (optional — lessons take place there by default),
- **Theoretical headcount** (optional but recommended for sizing classrooms).

The campus and the site answer two different questions: the campus indicates which entity of the institution the class belongs to; the site indicates where it is physically located and which classrooms are used by default.

△ If you forget to assign the site, lessons cannot be placed (no classrooms and no time grid). An alert reminds you of this.

2. Creating the groups for each class

Click a class's **+ Add group** button. Create as many groups as there are courses where students are not taught as a whole class.

For each group:

- **Name** and **code** (use both to make identification easy),
- **Theoretical headcount**,
- If several groups must or can be on the **same time slot** (Lab-A and Lab-B, or Spanish and German), create a **class division** with **+ Add class division**.

Tip: if a class's groups are (almost) identical to those of another, use the **Import from another class** button to import the configuration and adjust it.

PREMIUM

Classes and **groups** can be synchronized from an external system: **Synchronize** and **Synchronize** open the matching screen. See [Synchronization with external systems](#).

3. Configuring the subjects per class (the courses)

Go to the **Subjects** tab (or a class's **Courses** button).

Each subject you assign to a class in this way — with its hours and, optionally, its type (tutorial, lab, lecture...) — constitutes a **course**: the basic unit of your teaching offer ("mathematics in Grade 9 A, 4 h per week, as a tutorial"). The interface says "subjects" at this point, but what you are building is the notion of a course; together, a class's courses form its **program** (its curriculum, its brochure). The **lessons** you will later place on the grid are their occurrences.

For each course, specify:

- **Number of hours** per week,
- Optional **course type** (tutorial, lab, lecture...),
- Assigned **teacher(s)**,
- **Pedagogical weight** (the solver tries to balance the subject across the days),
- **Special classroom required** (from the site's [specialisations](#)),
- **Placement preferences** (avoid mathematics at the end of the day, sports during the hottest hours...).

Tip: use **Import from another class** to reuse another class's subject configuration. The dialog even lets you import a class from another timetable, and the scope can be extended: with or without replacing the existing list, and with or without the associated teachers, the groups, the incompatibilities, the class's time constraints. You can go as far as duplicating the distribution into lessons, again choosing how far the import extends. As for the positions of the lessons, on a calendar-type timetable, it is possible to shift the days (the sequence is then recomputed according to the working days of the source and the target). This is very handy for institutions that run recurring full sessions, for example six or eight consecutive weeks, several times a year.

4. Incompatibilities between subjects

An **incompatibility** forbids a subject from **following** another. The rule is **directional**: "subject A must not be followed by subject B" is not the same thing as the reverse. The typical case: "no mathematics right after sports". Declare them on the class's **Incompatibilities** tab.

When creating the rule, choose the application **window**:

- **Consecutive** — not in the immediately following lesson, on the same day. E.g.: avoid chaining a demanding subject onto an exhausting one, "no mathematics right after sports".

- **Half-day** — not later in the same half-day. E.g.: alternate languages, "no Spanish after English" — and, for a mutual exclusion, add the reverse rule "no English after Spanish".
- **Day** — not later in the same day. E.g.: spread out the artistic subjects, "no drawing after music on the same day". For this kind of balancing, the subject's **pedagogical weight** is often a simpler option.
- **Week** — not later in the same week. E.g.: respect the order of lecture then practical work, "no chemistry practical work before the chemistry lecture".
- **Always** (*calendar timetable*) — never afterwards, over the whole period: the **sequencing** tool for prerequisites. E.g.: "no C++ before C", "no finance before accounting".

A few pointers to pick the right setting:

- **A subject that must not be placed at a specific moment** ("no maths in the first period", "no sports in the middle of the day") is **not** an incompatibility: use the subject's **time constraints** (🕒), which govern the absolute placement of a single subject.
- **Preventing a subject from coming back twice on the same day** is handled more simply with the global **self-incompatibility** option than with pairwise rules.
- An incompatibility is a **soft** constraint (a penalty the solver tries to eliminate), not a hard block: it can remain in the generated timetable, where the **diagnostic** reports it. Avoid piling them up — each one reduces the solver's freedom.

What next

Next step: [Alignments and groups of groups](#), if you have courses shared across classes ([transverse courses](#)).

Otherwise, go straight to [Distribute the hours](#).

☰ How-to

Creating a class with its groups and class divisions

1. **A class** brings together the students of the same program. It breaks down into **groups** (half-classes, electives, ability levels), with **class divisions** to declare the mutually exclusive groups within the class.
2. **Click Create class**. Enter the name (`Year 7A`, `L1-Computer-Science ...`), the level (from those configured), the campus if you use it, the default site, optionally a dedicated classroom + theoretical headcount. Without a site, lessons cannot be placed — an explicit alert says so.
3. **Create the class's groups**. One group per distinct teaching use: `Lab-A`, `Lab-B` (lab half-classes), `German`, `Spanish` (second-language electives), `Advanced English`, `Standard English` (ability levels). Strong recommendation: **one group per distinct course**, even if the students are the same.
4. **Declare the class divisions** — mutually exclusive groups. Select the groups (e.g. `German + Spanish`), click `+ Add class division`. Without a class division, the solver treats two groups of the same class as potentially conflicting.
5. **Configure the subjects per class**: weekly hours, assigned teachers, default course type. Subjects are selected from the school's reference list (official or custom).
6. **Next step**: [Alignments and groups of groups](#) if you have courses shared between several classes, otherwise go straight to [Distribute the hours](#).

🔍 See also

[Class, group, subgroup](#)

[Class divisions](#)

[Campus](#)

[Class](#)

[Group](#)

3.10 Step 5 — Alignments and groups of groups

Source: <help/en/timetables/alignments-and-groups-of-groups.md> · id: [timetables.alignments-and-groups-of-groups](#) · Audience: admin · Updated: 2026-05-10

This step is **optional** — skip it if you have no shared cross-class courses. It becomes necessary as soon as students from different classes must attend **the same course together** (Latinists from several classes, cross-class electives, speciality courses, etc.).

Alignments

Usable in **all** timetable types, but most relevant for **recurring** timetables (weekly / cyclic) with parallel classes of identical structure. See the reference page: [Group alignments](#).

Summary:

- First create, in each class concerned, a **group** with the same name ("Latinistes" in 8A, in 8B, in 8C).
- Then, on the **Group alignment** tab, click [+ Add an alignment](#) and link the groups from the three classes.
- Check that the **hourly volumes are identical** across the aligned groups (otherwise Omniscol will raise an inconsistency diagnostic).

Groups of groups

See the reference page: [Groups of groups](#).

Summary:

- **Group of groups** tab.
- Create the group of groups by selecting the member groups (from the same class or from different classes).
- Then assign the courses to the group of groups — they will appear in all the parent classes.
- Editable at any time (adding / removing member groups).

Assigning several groups directly to a lesson

There is also a shortcut to **assign several groups directly to a lesson**, without creating a named group of groups.

This mode is convenient for a **one-off** or exploratory need:

- you tick several groups in the lesson's group selector;
- the lesson is then shared between these groups without creating a dedicated named entity;
- you can later revert to a single group, or formalize the case as a group of groups if the need becomes lasting.

On the other hand, it is **less readable** and **less reusable** than a named group of groups: the explicit business intent is lost and it is harder to spot, re-edit or reapply the same grouping across several lessons.

When to use which

Situation	Tool
Recurring shared course in weekly / cyclic mode	Alignment
One-off or evolving shared course	Group of groups
Membership likely to change	Group of groups
One-off need without a named structure	Direct assignment of several groups
Fixed, definitive configuration	Alignment or group of groups

Classic pitfalls

- **Different volumes across the aligned groups.** If one group has 3 hours of Latin and another 2, the alignment raises a diagnostic. Either harmonize, or unalign.

- **A course shared by only 2 classes** out of 3 aligned ones. For one of the classes to have its independent course on that time slot, duplicate its group and unalign the clone.
- **Students with a double enrollment** (for example a student who takes Latin in 8A but is enrolled in 8B). Solution: create a "Latinistes" group in 8A *and* in 8B with the same student in both, then align.

What next

Next step: [Distribute the hours and create the lessons.](#)

☰ How-to

Creating a cross-class alignment

1. **An alignment** declares that several groups from **different classes** must attend the same course, at the same time, in the same classroom, with the same teacher. Typical case: the Latinists of 8A, 8B, 8C.
2. **Prerequisite:** first create, **in each class concerned**, a group with the **same logical name** (`Latinistes` in 8A, in 8B, in 8C). See [Creating a class with its groups](#).
3. **Group alignment tab** → click `+ Add an alignment`. Select the groups of the classes concerned (`Latinistes 8A`, `Latinistes 8B`, `Latinistes 8C`). Confirm.
4. **Check the hourly volumes:** they must be identical across the aligned groups (3 hours of Latin everywhere, for example). Otherwise, Omniscol raises an inconsistency diagnostic.
5. **Calendar mode:** prefer [groups of groups](#), which are more flexible (editable afterwards, member groups added / removed without duplicating). `+ Add a group of groups` button on the dedicated tab.

If the need is **one-off**, you can also assign several groups directly to the lesson from the group selector, without creating a named group of groups.

6. **Next step:** [Distribute the hours and create the lessons.](#)

🔍 See also

[Group alignments](#)

[Groups of groups](#)

[Transverse course](#)

[Group alignment](#)

[Group of groups](#)

3.11 Step 6 — Distribute the hours and create the lessons

Source: <help/en/timetables/hour-distribution.md> · id: `timetables.hour-distribution` · Audience: `admin` · Updated: 2026-06-25

You have already declared the hourly volumes per subject for each class (step 4). This is where you break those volumes down into **individual lessons**.

The screen is organized in two parts:

- **On the left:** the list of classes and their **courses** (the class's subjects, possibly broken down by type), with a dynamic counter of hours already created per course.
- **On the right:** the lessons created for the selected class.

Three display modes

The right-hand panel works in **three modes**, switched with the selector at the top left of that panel. Each has its purpose, and you move from one to another **without losing your work in progress**.

Sticky notes (default)

Lessons appear as **cards** sorted by subject or by date (for calendar-type timetables). This is the mode detailed throughout the rest of this page: double-click to create, drag the border for the duration, card icons for the attributes.

- **Strengths**: very visual; ideal for **creating** lessons, breaking down the volumes, adjusting durations by dragging and handling complex lessons (alternate, concatenated...) directly on the card.
- **Drawbacks**: not very practical for acting on **many** lessons at once, for getting a finely filtered overall view, or for seeing placements and placing lessons quickly.

Timetables (reorganization)

Lessons are displayed in a layout close to the timetable, ready to be **reorganized in bulk**.

- **Selection**: a click selects a lesson and opens an interactive form.
- **Multi-selection**: **Shift+click** extends the selection to a **range**. The selected lessons are then edited **together** on a shared form (bulk duration, shared attributes...).
- **Dynamic filters** and the **choice of days to display** narrow the view down to what you are working on.
- **"Awaiting" filters**: isolate the lessons **awaiting classroom** or **awaiting teacher** to finish the missing assignments; warning badges point out what remains to be completed.
- A **days / entities toggle** flips the axis of the schedule depending on what you are comparing.
- **Strengths**: very efficient manual placement, combined views (class + teacher, for example), reorganizing many lessons, targeted filtering to deal with the gaps, fast **bulk** editing directly on a calendar.
- **Drawbacks**: less direct than sticky notes for initial creation, no creation of complex lessons, less fine-grained filtering for viewing and bulk editing.

Listing

A **table view** of the lessons: one **row per lesson**, one **column per piece of information** (status, class, subject, duration, position, alternate week, teacher, group, room, headcount, resource, memo).

- **Sorting**: **every column** can be sorted — click its header to sort on it, click again to reverse the order (ascending / descending).
- **Per-column filtering**: under each header, a **multi-choice drop-down menu** filters the column (sometimes **two** menus for a compound column such as day + time; a range field ≥ 10 , < 5 , 10–20 for the headcount). Several values in **the same menu** combine as **OR**, unless they are values of different kinds (for example, on rooms: between the tags, videoconferencing, size and name), in which case it is an **AND**; filters set on **several columns** stack as **AND**. A button resets all the filters.
- **Pagination**: for large volumes, the list is paginated; the page size **adjusts automatically** (from a few hundred to a few thousand rows) to stay responsive.
- **Simplified mass editing**: tick the lessons you want (selection column), then use the column's **header action button** (duration, teacher, group, room...) to apply it **in one go** to the whole selection.
- **Automatic room allocation**: choose a **set of lessons** and a **set of rooms**; the algorithm proposes an **optimal assignment** (headcounts, room constraints, availability), which you **adjust row by row** before confirming. See [Automatic classroom assignment](#).
- **Strengths**: sortable overview, **fine-grained per-column filtering**, **mass** editing and room assignment.
- **Drawbacks**: less visual time context than the other two modes, no creation of complex lessons.

Create a lesson

A *lesson* is an individual occurrence of a **course** (its subject, a duration, sometimes a group, usually a teacher and a room), later placed on the timetable.

Select a class, then **double-click a course**: a lesson of that subject is created with the default values (duration = the class's time unit, the course's teacher(s), the default room if any). It starts out **without a position** and joins the **lessons not yet placed**, ready to be positioned by the algorithm or by hand.

Double-clicking works in all three modes; the list of courses is simply in a different place: in the center in sticky notes mode, **under each class in the left-hand panel** in timetables and listing modes.

To then **fine-tune** the lesson (duration, group, teacher, room, resource), the interface depends on how it is displayed:

- **While it is a card** — in sticky notes mode, or not yet placed in the timetables view — drag the **bottom border** for the duration and click the card's **icons** for the attributes.
- **Once placed on the calendar, or in listing mode**, settings go through a **drop-down menu**: the *Edit lesson* form (click the lesson on the calendar), or the listing's row and column menus — the latter also allowing **mass** editing.

Break down an hourly volume


Example: the 4 weekly hours of a mathematics course can become:

- 4 lessons of 1 h, or
- 2 lessons of 2 h, or
- 1 lesson of 2 h + 2 lessons of 1 h.

The choice is yours, depending on your teaching rhythm. The course counter tracks your progress continuously (see below).

Automatic allocation

The **Actions** menu offers an **Automatic allocation** that fills each course's **missing volume** in one go: it adds lessons of the chosen duration (1 or 2 grid slots, 3 and 4 on calendar timetables — the value in hours therefore depends on the timetable's slot duration) and ends with a shorter lesson if the remainder is not whole. It respects the lessons already created, sets the default teachers and room, and leaves everything **unplaced**.

It only acts on the **visible courses**: combine it with the **filter** ( Filter) by subject or by type to allocate, for example, only the "Theory courses". In **listing** mode, it processes **all classes** at once; elsewhere, the **selected class**.

The per-course hours counter


Each course, on the left, carries a counter that sets the **hours created** against the **target volume** declared at step 4 (for example 3 h / 4 h). It is the **look of the card** that signals your progress:

- **Below the volume**: normal look — there are still hours to create.
- **Exactly at the volume**: the card **fades out** (reduced opacity), a sign that it is complete.
- **Above the volume**: **red border** — you have created too many hours.

The counter's badge, for its part, simply takes the **subject's color** (it can therefore be green, blue... with no link to progress); on hover, the card is underlined with the interface's accent color. Hovering also shows a tooltip detailing the **breakdown** of the lessons created (durations, types, alternate weeks, drafts / canceled).

Subgroups and the balance indicator

When a course is split into a **division** (several subgroups having their lessons **at the same time**, typically a half-class split), the counter does **not multiply** by the number of groups: a division counts **once** — otherwise the total would show $n \times \text{volume}$, which would make no sense.


What remains is to check that all the subgroups receive the same volume: that is the role of the  icon that then appears to the right of the course.

- **Green balance** — *Groups in class division in balance*: all the subgroups have the same number of hours.
- **Orange balance** — *Groups in class division out of balance*: one subgroup has more or fewer hours than the others.

Hovering over the icon lists the **detail per group** (hours of each subgroup), to immediately spot the one that needs catching up.

Duplicate a lesson

Two levels of duplication coexist.

Quick clone. The  Duplicate icon — present on the sticky-note card as well as on every listing row — creates **one copy** of the lesson in the **temporary zone** (unplaced), attributes included. Ideal for stringing together identical lessons.

Advanced duplication. In the lesson's form (the *Edit lesson* popup), the chevroned copy button opens a much richer menu, used mostly in the **timetables / calendar** view. You first choose **how many** copies (1 to 999) — or, on a calendar-type timetable, **up to a date** — then **how** to place them:

- **Add lesson to the temporary zone** — copies left unplaced.
- **Same time slot** — repeats the lesson on the same slot.
- **Next free slot(s)** — automatically looks for the slots compatible with the duration.
- **Next day(s) / previous day(s)** — shifts to neighboring days.
- **Next week(s) / previous week(s)** — (*calendar only*) carries over from one week to the next.
- **Choose the slots** — you point to the free slots yourself on the schedule.

On the per-day and per-week options, a **"Free"** button restricts the search to unoccupied slots only. On calendar timetables, this search can even extend beyond the current timetable when a cross-timetable comparison is active.

Memos



You can attach a **memo** (free-form comment) to a lesson via the  Comment icon. Several memos are possible, with different **visibility levels**:

- administrators only,
- administrators + teachers,
- everyone (visible to students and on the display panel).


Memos appear in the tooltips when hovering over lessons on the timetables.


Complex lessons

For sophisticated configurations, see the dedicated page: [Complex lessons](#). In short:

- **Alternate** (week A/B) —  Add week at the top right of the lesson, adds an alternate week.
- **Concatenated** (consecutive) — drag and drop one lesson under another.
- **Associated** (alternating half-groups) — a button that appears between two concatenated lessons.
- **Co-taught** (several teachers on the same lesson) — multi-teacher selection on  Assign teachers.

Mass actions

The  Actions menu at the top right gives access to the mass operations:

- **Cancel the placement** of all lessons (except locked ones),
- **Remove the locks** on placements,
- **Clear the room assignments**,
- **Assign the preconfigured rooms** (the class's or the teacher's default room, excluding special rooms),
- **Automatic allocation** of lessons over 1 or 2 periods,
- **Spreadsheet import / export** (see [Mass import](#)),
- **Delete all lessons** of a class .

Dynamic error detection

On every change, Omniscol checks consistency and displays the alerts at the top of the screen and on the affected classes / lessons. See [Diagnostic](#).

Save

⚠ Deleting a lesson is **immediate and permanent** (unless you have not saved yet). Save regularly with .


There is no global Undo / Redo function on this screen. Before saving, review the deletions and the mass actions.

What comes next

Next step: [Automatic generation](#).

☰ How-to

Distribute a course's hours

1. **You declared the hourly volumes of each course** on its classes at step 4. This step **breaks down** those volumes into individual lessons (one lesson = one slot, later placed by the algorithm or by hand).
2. **Select a class on the left**. The right-hand panel shows the lessons already created for that class, grouped by course. Each course's counter sets the hours created against the target volume (3 h / 4 h): the card fades out when the volume is reached, and turns to a red border when it is exceeded.
3. **Double-click a course** to add a lesson with the default settings (duration, type). Drag the lesson's **bottom border** to adjust the duration. Click the **icons** in the card to change group, teacher, room, resources.
4. **Break down the volume however you like**: a 4 h course = 4 lessons of 1 h, or 2 lessons of 2 h, or a mix. The course card fades out as soon as you are exactly at the target volume. In a division (simultaneous subgroups), a green / orange  icon signals whether all the subgroups receive the same volume.
5. **For identical lessons in a series**, create the first one then **duplicate**: quick clone to the temporary zone via , or advanced duplication from the form (same slot, neighboring days / weeks, free slots...), especially in calendar view.
6. **Complex lessons** (alternate A/B, concatenated into a double period, associated with rotation, co-taught by several teachers): see the dedicated icons on the lesson's card and the page [Complex lessons](#).
7. **Next step**: [Automatic generation](#).

🔗 See also

[Complex lessons](#)[Mass import of courses from a spreadsheet](#)[Course](#)[Lesson / Session](#)

3.12 Automatic classroom assignment

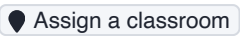
Source: <help/en/timetables/auto-room-assignment.md> · id: [timetables.auto-room-assignment](#) · Audience: [admin](#) · Updated: 2026-06-27

Automatic classroom assignment assigns a classroom to a set of selected lessons **in one go**. You choose a **set of candidate classrooms**, the algorithm proposes an optimal assignment, and you **validate after review** — row by row if needed.

It is a tool for **classroom allocation only**. It does not move lessons in time, does not change their duration, does not assign resources and does not rerun the full timetable generation. To have classrooms placed **during** generation, see [Automatic generation](#) instead.

Where to find it

This tool lives in the **listing view** of the [Distribute the hours and create the lessons](#) screen (the table view, one row per lesson).

1. Display the lessons in the **listing view**.
2. **Tick** at least two lessons (selection column). If needed, use **sorting** and **column filters** to isolate the lessons to process — for example those **awaiting a classroom**.
3. Open the grouped classroom action in the header, .
4. In the classroom selector, enable Automatic classroom assignment mode.

Remote or **self-study** lessons (according to their [modality](#)) are ignored: they need no physical classroom. **Hybrid** lessons remain taken into account. Leave lessons whose classroom is **locked out of the selection**: the tool does not override a lock.

Choosing the set of candidate classrooms

In automatic mode, the selector displays classrooms grouped by **site** then by **building**, with checkboxes at three levels:

- ticking a **site** takes all its eligible classrooms;
- ticking a **building** takes all the classrooms of that building;
- ticking a **classroom** adds it individually.

Classrooms already assigned to the selected lessons are **pre-ticked**. A classroom whose **specialisation** matches none of the selected lessons, or that is **too small** for all the compatible lessons, is **disabled**. The capacity turns **red** when the classroom cannot host any of the compatible lessons, **orange** when it can host only some of them.

Confirm the set to start the computation.

What the algorithm does

From the chosen set, the algorithm assigns **one classroom per lesson**, respecting **strict rules** and optimizing the rest:

- the **specialisation** required by the lesson's subject;
- the classroom's **capacity** against the lesson's headcount;
- the classroom's **existing occupancy** on the time slot — lessons from other timetables or other linked accounts, events, classroom time constraints, shared [large rooms](#);
- a **preference for the lesson's reference site**;
- **alternate weeks**: two lessons that never fall in the same week can reuse the same classroom.

The algorithm can **reshuffle classrooms among the selected lessons** to house them better, and prefers the **smallest suitable classroom**. When the current classroom already suits and another would bring only a marginal gain, it **keeps** it rather than producing a needless change. If **no** classroom in the set suits, the lesson is left **without a solution**.

Review, adjust, apply

The computation opens a **preview window**: as long as it is open, the actual timetable **is not modified**. Each row carries a **status** — assigned, unchanged, no solution, or manually adjusted — along with the **previous** classroom and the **proposed** one.

- **Adjusting a row**: click its proposed classroom to open the regular selector. The manual choice is **not limited** to the initial candidate set; the row switches to the "adjusted" status and any conflicts are immediately

recomputed on all rows.

- **Resetting**: a button restores the algorithm's proposal on the adjusted rows.
- **Unticking** a row **excludes** it from being applied.
- **Validate** applies the ticked rows to the timetable; **Cancel** closes without changing anything.

After applying, the modified lessons **flash** and a summary shows the **number of lessons modified, unchanged and without a solution**.

Notes

- A lesson that carried **several classrooms** can be brought down to **a single one** by the automatic assignment.
- A lesson **with no positioned time slot** can still receive a classroom (by set, specialisation, capacity and site), but no time conflict can then be checked for it.
- To simply **set the default classrooms** (the class's or the teacher's, excluding special classrooms) or **clear** assignments, the **Actions** menu offers dedicated mass actions — see [Distribute the hours and create the lessons](#).

See also

[Distribute the hours and create the lessons](#)

[Sites, time grids, classrooms, resources](#)

[Classroom specialisations](#)

[Large room](#)

3.13 Step 6b — Mass import of courses from a spreadsheet

Source: [help/en/timetables/mass-import.md](#) · id: [timetables.mass-import](#) · Audience: admin · Updated: 2026-06-02

If you already have the list of your courses in a spreadsheet — an export from your SIS, a schedule prepared beforehand in Excel, or even an export from another program such as Hyperplanning or Aurion — you can **copy and paste** it directly into Omniscol.

Accessing the import

In the **Actions** menu (on the hours distribution screen), choose **"Table - import/export"**. If no class has been created yet, a direct link is also available.

The import takes place in **four steps**.

Step 1 — Paste the table

An editable area opens with a **column template pre-defined** by Omniscol. Columns are typed by position (3rd column = subject, etc.); you can **rearrange the template columns** so that they match the order of your source spreadsheet — which makes the copy-paste from Excel or Google Sheets clean, without rewriting your file.

Recognized fields:

- **Class + subject** (at a minimum, for a course to be valid),
- **Duration**, or **start time + end time** (and **day** for pre-placed lessons),
- **Group** (free-form separators for multiple groups),
- **Course type** (tutorial, lab, lecture...),
- **Teacher(s)** (free-form separators),
- **Site + Room(s)** (several rooms per lesson are possible, see [multi-room](#)),
- **Resource(s)**,
- **Alternating weeks** (A/B , $1/2$... format),
- **Comment**.

You can also import the **list of courses already created** (which is what the export produces) if you want to rework them in Excel and re-import.

Step 2 — Checking the schedule and sites

Omniscol displays the detected hours (still editable). If several sites are guessed, you can **drag and drop** misassigned classes, rooms and resources to the correct site. If a superfluous site was created, empty it and it will disappear.

Step 3 — Disambiguation

After the imported values are analyzed, the table reappears with **drop-down menus** on the ambiguous fields. Check and correct the proposed matches.

This is where you settle, for example:

- "Maths" = which exact subject (with or without a type)?
- "M. Dupont" = which of the three M. Duponts?
- "A102" = room A102 on the Lyon site, the Avignon site, or another?

Step 4 — Creating unknown entities

If the import detected entities that do not exist in the database (classes, groups, groups of groups, teacher, custom subject, course type, site, room, resource...), a final step asks you:

- to **confirm their creation** in the timetable,
- for **custom subjects** and some **teachers**, to request, if needed, creation on the **Administration** side to make them available across the whole school.

Limitations

The import automates preparation, but does not replace a human check:

- **Complex courses** (multiple alternations, concatenations, associations) are sometimes imperfectly rebuilt — plan for a manual finishing pass.
- With very specific internal naming schemes, some columns will remain ambiguous — you will settle them at step 3.
- For **dedicated migrations** from a specific program (Aurion, ASC...), [specialized adapters](#) are available.

☰ How-to

Preparing and importing the courses

1. **Before clicking Import**, take 10 minutes to prepare the source file. A clean file = an import with no back-and-forth. In Excel or Google Sheets, remove merged rows, multi-line headers and subtotals. **One row = one lesson**. Keep consistent labels for classes, subjects, teachers, groups, sites, rooms and resources.
2. **Open the import screen** from the hours distribution screen, menu *Actions* → *Table - import/export*. Rearrange the template columns so that they match the order of your spreadsheet, then **paste the table** into the editable area. This is step 1.
3. **Step 2 — checking the schedule and sites**. Omniscol displays the detected hours and the guessed sites. If needed, drag and drop classes, rooms and resources to the correct site, or empty a superfluous site to make it disappear.
4. **Step 3 — disambiguation**. The table reappears with drop-down menus on the ambiguous fields: exact subject, teachers with the same name, rooms with the same name, etc. You make the call here.
5. **Step 4 — creating unknown entities**. Confirm the creations proposed based on your file: classes, groups, groups of groups, teachers, subjects and types, sites, rooms, resources. For custom subjects and some teachers, you can also request creation on the Administration side. The courses are then injected into the timetable.
6. **At the end**, you get a partially configured timetable. Plan for a manual finishing pass for the complex cases (multiple alternations, concatenations, associations) that the import does not always fully rebuild. For migrations from another program, see [Migration from another program](#).

🔗 See also

[Preparing your data for a mass import](#)

[Migration from another program](#)

[Previous step — Distributing the hours](#)

Next step — [Automatic generation](#)

3.14 Step 7 — Automatic generation

Source: [help/en/timetables/generation.md](#) · id: [timetables.generation](#) · Audience: *admin* · Updated: 2026-06-24

Once the lessons are created, you position them on the grid. Building a timetable is in practice a **cycle**, not a one-off action:

1. you **place a few lessons by hand** — to anchor immovables or lock strong constraints (dated interventions, exams, manual anchors);
2. you **generate** (everything or a subset) so the algorithm places the rest under constraints;
3. you **adjust at the margins** — fix a special case, move a lesson, change your mind;
4. and it **can loop back**: you regenerate as you go (added lessons, or — in calendar mode — by date ranges, by batches of classes or campuses).

You can also do everything **100% by hand**, without the solver. Manual placement (placing, moving, locking, one by one or in batches) has its own dedicated page: [Manual placement](#). For the overall flow, see [Module overview](#).

This page covers **automatic generation**.

Availability

Available on [weekly](#), [cyclic](#) and [calendar](#) timetables. In calendar mode, lessons are placed on real dates and the options can limit the date range to generate.

Preparation


Before clicking  :

- expand the **Generation** tab (labeled Checking on a calendar timetable) to check the global statistics,
- fix the **critical alerts** (red) — they block generation,
- review the **warnings** (orange) — they do not block but signal a risk (overly restrictive availability, borderline capacity, etc.).

The screen also shows:

- the number of lessons created vs lessons positioned,
- the hour volumes entered vs the teachers' service hours,
- the groups assigned to at least one course (to spot forgotten or unused groups).

Launching

Click . Omniscol spins up a dedicated, parallelized computing environment. The duration depends on the size of the timetable, the number of constraints and the chosen options. There is no queue to manage on the school side; initialization often takes around ten seconds before the computation actually starts.

The computation is **capped at twenty minutes**, but in practice this is far beyond what is needed: a timetable of about 200 courses often responds in about twenty seconds — the algorithm is nearly instantaneous, most of the time goes into starting the machine — and ~600 courses in about one minute. For a very large institution approaching this cap, the Omniscol team can increase the power of the computing machine, on request.

Generation options

By default, generation places all the created lessons that are not yet positioned. It can also move already positioned lessons if that improves the timetable. Locked lessons keep their position.

The options menu notably lets you:

- manage rooms with three choices: optimized assignment by default, verification only, or ignoring rooms entirely;
- select only certain classes or certain subjects;
- restrict a calendar timetable to a target date window;
- ignore already assigned rooms or the current placements;
- assign only the rooms without recomputing the whole placement;
- run a fast generation without optimization;
- allow flexible placement on the sub-slots of the time grid when the timetable uses sub-slots.

The advanced optimization settings also let you reduce gaps, avoid two lessons of the same subject on the same day, spread lessons out, set hour minima or maxima, or choose the number of attendance days for teachers. Practices vary across countries: in France, some institutions aim to reduce the number of attendance days for teachers; in other contexts, you may instead prefer to spread those days out, for example to organize other on-site duties.

Calendar-mode-specific options

In calendar mode, the advanced options can limit generation to a **target date window** shorter than the total range of the timetable. This is useful to generate only part of the year: a one-month test, progressive construction of a semester, or reworking a period already prepared.

The other important option is **day compacting** for classes. By default, the solver tends to seek a balance across the available days; if you generate only part of the year's lessons, this behavior can artificially spread out the courses. Compacting lets you request a grouping:

- at the start of the window;

- at the end of the window;
- with no start or end preference, but keeping the days as compact as possible.

How generation works

Understanding the solver's logic helps you anticipate what it will place, what it will sacrifice, and how to steer it. It pursues **two goals, in this order**:

1. **Fit everything in** — the absolute priority: position *all* the lessons while respecting the hard constraints.
2. **Optimize quality** — once everything fits, reduce gaps, smooth out the days, honor the preferences.

It starts from a clean state (**locked** lessons keep their place, the others are replayed) and places **from the hardest lesson to the easiest**. Difficulty mostly comes from the **tightness of the constraints**: a course whose teacher is only available one afternoon per week, or a **four-hour block** to fit in one piece, goes to the front. At equal difficulty, it alternates subjects and handles **the rarest before the most common**, to distribute each one's chances fairly.

When not everything fits: the sacrifice

If the timetable is too constrained to fit everything, the solver **does not give up**: it picks the smallest set of lessons to **set aside** so that the rest fits, then optimizes this partial timetable. The sacrificed lessons stay in the sticky notes bar, waiting to be placed. Since the ordering prioritized the hardest and the rarest, it is mostly **easy and numerous** lessons that yield: the 5th hour of a high-volume subject before the single hour of a rare subject.

You stay in control of these trade-offs: **lock** the lessons that must never move — placed first, they are not sacrificed — and **relax the over-constraint** (open time slots, availability) where possible, so that everything fits.

Hard constraints and soft constraints

Not all constraints are equal:

- **Hard** — never violated: **occupancy** (a teacher, a class, a group, a room or a resource cannot be used twice at the same time), the ranges marked **Unavailable** (in black) on a teacher, a class, a room, a subject or a group, the **lack of a free room or resource** on the time slot, the **closed days**, the **locked positions** and the **alignments**.
- **Soft** — everything else, which the solver can **break** at the cost of a penalty in order to fit everything: the ranges marked **Undesired** (in red — to avoid, not forbidden) and the optimization preferences (compactness, spreading, grouping by day...). A *high*-priority constraint weighs more than a *low*-priority one, but none is blocking. The solver also tries targeted relaxations (placement on the sub-slots of the grid...) when that is the price to pay to sacrifice no lesson.

Checking without optimizing

The **fast generation** option (without optimization) runs a lightweight pass that mostly answers one question: *does everything fit?* Useful to test the feasibility of a configuration before investing the time in a full optimization.

Result

- **Green banner**: generation succeeded. The lessons in the requested scope could be positioned and the result can be inspected.
- **Red banner**: no complete solution was found. The best computed timetable remains viewable and the unpositioned lessons appear in the sticky notes bar for diagnosis.

See [Diagnosing a failed generation](#).

Consolidate or copy

After a successful generation, Omniscol offers two options:


- **Consolidate** into the current timetable (the placement replaces the existing one).
- **Create a copy** (the current timetable stays intact, the optimized result becomes a new timetable).

If you plan to regenerate later (because you are building incrementally, for example), **do not consolidate**: otherwise the already positioned lessons would be starting points for the new generation.

Locking a lesson before running another generation

On an already positioned lesson, use the position **padlock** to lock it. The next generation will not move it but will adjust the other lessons around it. Handy for anchoring "immovable" lessons (fixed external interventions, dated exams, manual adjustments...). Full details in [Locking a lesson](#): mass locking, effects on the diagnostic (red → orange), behavior of the algorithm.

Exporting a timetable as JSON

The  button exports **the current timetable only** in JSON format. This is not the full account export documented in [Import and export](#).

Typical cases:

- keeping a record of a draft or a scenario before deletion;
- archiving an intermediate state before a major reorganization;
- sending this specific timetable to the Omniscol team as part of support.


As with the full JSON export of the school account, re-importing is not a standard use freely available on the school side.

What next

Final step: [Publishing \(activating\) a timetable](#).

How-to

Run an automatic generation

1. **Automatic generation** positions the lessons on the grid while respecting dozens of constraints (availability, capacities, alignments, specializations, alternations). Available on weekly, cyclic and calendar timetables.
2. **Check the Generation tab**. The global statistics appear there: lessons created vs positioned, hour volumes entered vs teachers' service hours, critical alerts (red — blocking), warnings (orange — warn without blocking).
3. **Fix the red alerts** before launching. As long as any remain, generation cannot start — unless you force it via the "More options" menu, which is not recommended. Orange ones call for a decision but do not prevent generation.
4. **Set the generation options** if needed: rooms to assign, verify only or ignore; targeted classes or subjects; fast generation without optimization; ignore already assigned rooms or current positions. In calendar mode, the advanced options also let you restrict the generated date window and compact the lessons at the start of the period, at the end of the period or with no preference.
5. Click  **Generate timetable**. Processing runs in the background; the screen shows the state of the generation and lets you inspect the result when it is available.
6. **Save** when the score suits you. The timetable is positioned. Next step: review any conflicts, adjust the special cases by hand (see [Manual placement](#)), then publish.
7. **What next**: [Publishing \(activating\) a timetable](#).

See also

[Manual placement](#)

[Diagnosing a failed generation](#)

[Solver](#)

[Detecting and resolving conflicts](#)

3.15 Step 8 — Publishing (activating) a timetable


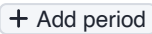

Source: help/en/timetables/publication.md · id: [timetables.publication](#) · Audience: admin · Plan: standard · Updated: 2026-06-13

△ A step that is **often forgotten**: a generated timetable is **not published automatically**. Until it is published, it is a draft — visible only in the Timetable management module, by administrators.

Only after publication does the timetable appear in the [Timetable](#) module consulted by all authorised users.

How to publish

On the home page of the Timetable management module:

1. Select the relevant **school year** in the **timeline at the top**. By default, the **current school year** is displayed. If you are preparing the next school year, click the **right arrow of the timeline** to switch to year N+1 — otherwise you will be trying to publish on the weeks of the current year, where the weeks are already assigned to the current timetable. See [Timeline and time navigation](#) for the general concept.
2. Click  **Timetable allocation** (Timetable allocation).
3. A grid appears: timetables as rows, weeks as columns.
4. For each timetable, **select the weeks** where it applies (click or drag on the strip of weeks). The  **+ Add period** button adds an extra range for a timetable.
5. Save with  **Save**.

Standard account: one timetable per week

With a standard account, **only one publication** per week is allowed. If you want to publish two different timetables, they must cover disjoint ranges (semester 1 vs semester 2, for example).

OPTION: MULTIPLE ACTIVE TIMETABLES IN PARALLEL

Simultaneous publications

When your account has this capability (included in Premium), you can publish **several timetables in parallel** on the same weeks. The engine merges the views dynamically: if two published timetables share a teacher or a classroom, their consolidated timetable shows the lessons of both timetables.

Typical use case: a weekly timetable for recurring core-curriculum courses + a calendar timetable for one-off masterclasses.

Calendar timetable: binary publication

A timetable in [calendar](#) mode is not published by week ranges (its lessons are already individually dated). Publication is binary — the timetable is either published, or it is not.

Weekly / cyclic timetable: publication by ranges

For weekly and cyclic timetables, you add **one or more publication ranges**, then save. Pending changes appear hatched for additions and translucent for removals.

Alternating weeks: which lessons on which dates

When a timetable with alternating weeks goes live, each real week keeps only the lesson whose rank in the cycle matches the **week count**. Holidays can shift this count: the **timeline strip** of the [school year](#) screen lets you force the points where it restarts at week A, so that the right alternating lessons are kept on each date.

Assigning students after publication

Once the timetable is published for the current school year, you can **assign students to their classes and groups**:


1. Students module.
2. Select the students concerned.

3. Assign to a class .

Students will only see their personalised timetable once this assignment is done — and after switching the current year if you are preparing the next one.

How-to

Publishing a timetable in 5 steps

1. **Publishing makes a timetable visible** in the Timetable module of all authorised users. Until it is published, the timetable remains a draft visible only to administrators.
2. **Check the school year displayed** in the timeline at the top. By default, it is the current year. To publish on the next year, click the **right arrow of the timeline** to switch to N+1 before continuing.
3. **Click**  **Timetable allocation** . A grid appears with the timetables as rows and the weeks of the school year as columns.
4. **Select the weeks** where the timetable must apply. Click or drag on the strip of weeks. The selected weeks appear coloured; the others remain white.
5. **Save the publication**. The timetable goes live immediately: it becomes visible in the Timetable module of all authorised users.
6. **Next step: assign the students** to their classes and groups. This assignment can only be done after a first publication. Students will then see their personalised timetable.

See also

[Publication / Activation of a timetable](#)

[Multiple active timetables in parallel](#)

[Preparing the next school year](#)

3.16 Multiple active timetables in parallel

Source: <help/en/timetables/multiple-active-timetables.md> · id: `timetables.multiple-active-timetables` · Audience: `admin` · Options: `multiple-active-timetables` · Updated: 2026-06-13

OPTION: MULTIPLE ACTIVE TIMETABLES IN PARALLEL

At the standard tier, a **single timetable** can be published at a time over a given period. This is enough for the typical operation of a middle or high school. In some contexts — typically higher education, modular programs, schools with multiple tracks — the need arises to **publish several timetables in parallel** that complement each other. This capability is included by default with Premium accounts.

In some mixed school accounts, for example middle-high school or primary-middle-high school with several genuinely separate timetables, Omniscol can enable this capability specifically. Scoping and billing are then adapted to the actual need.

Why several timetables in parallel

A few typical scenarios:

- **Common core + specializations** — a weekly timetable covers the common core (subjects shared by all cohorts), timetables per specialization or per track are added on top with their own constraints.
- **Regular + one-off calendar** — a regular weekly timetable carries most of the courses, a calendar timetable carries the dated lessons (lectures, exams, juries).
- **Multiple sites** — one timetable per site with local management, merged at viewing time for students who move from one site to another.

- **Programs of different lengths** — for example an 18-month EMBA track, a Grande École cycle running on calendar semesters and a weekly preparatory program, with shared teachers and rooms.

Without this option, these cases force you to merge all the complexity into a single timetable, which quickly becomes unmanageable.

How it works

You create several timetables in the Timetable management module, as usual. At **publication** time onto the weeks, the allocation screen accepts several timetables on the same weeks rather than just one.

On the consumption side (portals, display panels, iCal, API), the timetables published on a week are **merged dynamically**: a student sees all the lessons that concern them, regardless of which timetable they come from.

Within the same account, the timetables active in parallel share the occupancy of the resources concerned: rooms, teachers and, in the cases where it is relevant, classes. This prevents the same teacher or the same room from being booked twice by two active timetables. Classes shared across several timetables exist too, but this case remains more marginal and depends heavily on the pedagogical organization.

Cross-timetable conflict detection

Two mechanisms complement each other, depending on where you are working.

While building the timetable

When you build or reorganize a timetable in Timetable management, the **Synchronization** button (in the toolbar of both the reorganization and the calendar timetable editing views, on **shared dates**) checks your work against the **other published timetables**. It automatically takes into account those whose **dates overlap** those of the current timetable and which share at least one **teacher, room or class** — locally (same account) as well as with **linked accounts**. It therefore activates on its own as soon as there are shared resources (orange = active). This automatic coupling links **dated (calendar)** timetables.

A welcome exception: it **ignores a plain copy** of the current timetable (same lessons). You can therefore work on a **duplicated draft** without being put in conflict with yourself.

The occupancy of these timetables then feeds into the conflict engine:

- the **candidate time slots** for placement (colored dots) turn orange or red if the teacher or the room is already taken elsewhere;
- the **diagnostic** reports double bookings across timetables.

The reported conflict then clearly states that it comes from an external timetable, with its precise reference.

The button's drop-down menu lists the timetables and accounts taken into account: untick those you do not want to monitor, for example to deliberately accept an overlap.

At viewing time (operational timetable)

Once published, the timetables **merge** into a single **operational** timetable — the viewing screen in Omniscol (the Timetable module), and everything that consumes the timetable (portals, display panels, iCal, API). Conflicts between timetables can be read there naturally: a teacher (or a room) taken by two simultaneous lessons from two published timetables sees them overlaid on the merged grid. That is the point of the merge: it forms the timetable as actually experienced, **all types combined** (including a weekly timetable and a calendar timetable). In the reorganization view, this creates a conflict.

In both cases, resolution remains a scheduling action: move, adapt or accept the conflict as appropriate.

☰ How-to

Publishing a regular timetable + a calendar timetable in parallel

1. **The classic hybrid case:** a regular **weekly** timetable for most of the courses, plus a **calendar** timetable for the dated lessons (guest lectures, exams, juries).
2. **Prerequisite:** two timetables created in Timetable management — a weekly one with the regular courses, a calendar one with the dated lessons. Configure them independently. See also [Calendar mode](#).
3. **Run the generation** on each timetable separately if you use the solver. First check the conflicts specific to each timetable.
4. **Publish the weekly timetable** on the desired weeks (typically: the whole year). See [Publication](#). Then **publish the second timetable** (calendar) on **the same weeks**. The allocation screen accepts several timetables in parallel (vs a single one at the standard tier).
5. **Check the merge:** open the student or teacher portal — the lessons from both timetables appear on the same grid, merged dynamically. iCal, the API and display panels see the same thing.
6. **Cross-timetable conflicts:** between a weekly timetable and a calendar timetable, the overlap shows up mostly on the **merged operational timetable** (viewing screen, portals) — if a lecture (calendar) lands on the time slot of a regular course (weekly) for the same teacher or the same room, the two lessons overlay each other there. Resolve it like an ordinary conflict: move, adapt or accept.

📌 See also

[Publishing a timetable](#)

[Calendar mode](#)

[Linked accounts and shared resources](#)

[Omniscol plans and options](#)

3.17 Manual placement of lessons

Source: [help/en/timetables/manual-placement.md](#) · id: [timetables.manual-placement](#) · Audience: [admin](#) · Updated: 2026-06-23

Placing a lesson **by hand** means choosing its time slot yourself, instead of letting the algorithm decide. This almost always combines with [Automatic generation](#): building a timetable is a **cycle** (place a few lessons → generate → adjust), described in [Module overview](#).

When to place by hand

- **Anchoring immovables before generating** — a dated guest session, an exam, an imposed fixture: you place it (and often **lock** it, see [Locking a lesson](#)) so that the generation builds *around* it.
- **Making marginal fixes after a generation** — moving a lesson, settling a special case, changing your mind about a time slot.
- **Building entirely by hand** — on a small timetable, or when you prefer to keep full control from start to finish (see below).

The gesture: pin, then colored time slot



Placing a lesson is **never a drag and drop**. Click the lesson's placement button (the pin, [Place on timetable](#)) — on its sticky-note card, its listing row, or in its details ([Place on timetable](#)) if it is already placed. Omniscol then displays **all the possible time slots**, colored by conflict level:

- **green** — no conflict;
- **yellow** then **orange** — minor to moderate conflicts (the reason is displayed);
- **red** — blocking conflict.

The details of each level and the full list of detections are in [Conflicts and diagnostic](#).

The time slots appear on the class **and** on the teacher(s) concerned, along with the other lessons already placed to help you choose. **Click the chosen time slot** to place the lesson there. To remove it, double-click a lesson that is already placed (it goes back to the unplaced lessons area, without being deleted).

Batch placement — schedule view

Click  **Grid**: the schedule of the classes / teachers appears, handy for chaining placements. The gesture does not change — a lesson's pin , then a click on a colored time slot. This is useful in calendar mode to pin down imposed dates or to locally rework the result of a generation.

Locking to freeze a manual placement

A lesson placed by hand remains, by default, movable by the next generation. To make sure it **stays put**, lock its position (padlock): the generation then builds around it. Details (mass locking, effect on the diagnostic): [Locking a lesson](#).

Doing everything by hand (without generation)

Nothing forces you to run the solver. On a small weekly timetable, or when placement is heavily constrained by human choices, you can place **each lesson** one by one (or in batches in the schedule view) and never generate. The real-time diagnostic stays active: conflicts (teacher, room, class, resource) are reported as you go, just as during generation. You then publish as usual.

For lessons that fall outside the grid (explicit times, extended exams), see [Off-grid lessons](#).

What next

Once the timetable is placed — by hand, by generation, or both — [Publishing \(activating\) a timetable](#).

See also

[Automatic generation](#)

[Locking a lesson](#)

[Off-grid lessons](#)

[Distribute the hours and create the lessons](#)

3.18 Editing a lesson

Source: <help/en/timetables/lesson-edit.md> · id: [timetables.lesson-edit](#) · Audience: admin · Updated: 2026-06-12

Once the lesson is created (see [Distribute the hours and create the lessons](#)), you adjust its **fields** one by one: subject, group, teacher, room, resources, duration, memo, and — depending on your plan — status and modality. This page sums up **all the fields** of a lesson and how to change them.

Where a lesson is edited








The same fields can be set in **three ways**, depending on the display mode of the hours distribution (see [Three display modes](#)):

- **Sticky notes**: each lesson card carries a row of **icons** (one per field); a click opens the matching selector.
- **Reorganization**: clicking a lesson opens a **lesson form** gathering those same fields; a multi-selection edits them in bulk.
- **Listing**: one **column per field**, with mass editing from the column header on the ticked lessons.


Whichever path you take, it is the **same lesson** you are editing: the values are consistent from one mode to the next, and nothing is published until you have saved.

Standard fields

Available on all plans:

- **Subject**  — the subject attached to the lesson, among those declared on the class.
- **Group**  — the group or subgroup involved when the class is divided.
- **Teacher**  — the teacher or teachers. Selecting **several** teachers on the same lesson amounts to **co-teaching** (see [Complex lessons](#)).
- **Room**  — choose the lesson's main room; the advanced variants are detailed further down, depending on the features active on the account.
- **Resources**  — the equipment involved (projector, tablets, etc.), with availability checking.
- **Duration** — drag the card's **bottom border** (sticky notes mode) to lengthen or shorten the lesson.
- **Position and lock**  — pin a lesson so that the generation does not move it (see [Locking a lesson](#)).
- **Memo**  — a free-form comment, with visibility levels (see [Memos](#)).

Room and assignment

Assigning a room  is open to all plans: choose a room from the list, or let the **automatic mode** suggest a compatible room (capacity, tags, availability).

PREMIUM

Advanced lesson features

Lesson status

The **status**  indicates where a lesson stands in the planning cycle. Four values:

- **Planned** — the normal status of a placed lesson.
- **Draft** — lesson **hidden** and **ignored by the automatic generation**: useful for preparing a lesson without imposing it on the timetable yet.
- **Canceled** — lesson **displayed as canceled** but kept for the record; it is **ignored by the generation**.
- **Done** — lesson **carried out**, counted in the dashboards and in billing. This status is offered on calendar-type timetables, where tracking actual lessons makes sense.

The status is carried by the **whole lesson** (not by an alternate-week variant). You change it lesson by lesson, or **in bulk** from listing mode.


Modality

The **modality** specifies how the lesson takes place:

- **In person**;
- **Remote**;
- **Hybrid**;
- **Self study**.

The modality is chosen in the room selector, next to the assignment. It has a concrete effect: a **remote** or **self-study** lesson requires **no room**, and the generation does not ask for one. The lesson's icon reflects the chosen modality.

Multi-room

A single lesson can receive **several rooms**: hold **Shift**  while clicking to extend the selection. Handy when a cohort spreads over two neighboring rooms, or when a lesson occupies a doubled space. The selected rooms are all booked and count towards conflicts.

Custom durations and times

Beyond the duration read from the grid, a lesson can carry **custom durations** (actual duration for the dashboards, counted duration for billing) and an **off-grid time** (explicit start and end times). These settings, along with the off-grid classes, are detailed on their dedicated page: [Off-grid lessons](#).


Edit several lessons at once

To apply the same value to a batch of lessons:

- in **reorganization**, **Shift+click** extends the selection to a range; the shared form then edits all the lessons together;
- in **listing**, tick the lessons you want then use the column's **header action button** (duration, teacher, group, room, status...).


It is the fast lane for, say, switching a set of lessons to **Draft** or assigning them a shared room.

Delete a lesson

The  **Delete** icon removes the lesson. The deletion becomes **permanent** upon saving: before you save, review the deletions, as there is no global undo on this screen.

How-to

Change a lesson's status

1. Open the class's hours distribution ([Step 6](#)).
2. Click the lesson's **status** icon  **Status** (or open its form in reorganization mode).
3. Choose the status you want — for example **Draft** to hide it from the generation.
4. To process a batch, switch to **listing**, tick the lessons and apply the status from the column header.
5. **Save** to publish the changes.

See also

[Distribute the hours and create the lessons](#)

[Off-grid lessons](#)

[Locking a lesson](#)

[Complex lessons](#)

[Sites, time grids, classrooms, resources](#)

[Detect and resolve conflicts](#)

3.19 Off-grid lessons and classes

Source: [help/en/timetables/off-grid-lessons.md](#) · id: [timetables.off-grid-lessons](#) · Audience: admin · Plan: premium · Updated: 2026-06-13

PREMIUM

PREMIUM

An **off-grid lesson** is a lesson placed with explicit start and end times, instead of following exactly the boundaries of a grid slot. This possibility exists in weekly, cyclic and calendar timetables.

The **off-grid class** is a separate setting: it is a property of the class, available for calendar-type timetables.

Lesson with custom times

Use custom times when a one-off lesson does not fall exactly on the grid: an exam starting at 08:30, an evening session, a lesson encroaching on a break, a booking that starts before the actual lesson.

The lesson remains visible in the timetable and still takes part in conflict detection. Every entity used by the lesson — teacher, room, class, group or resource — is considered busy as soon as the custom times encroach on a time slot.

For automatic generation, the lesson is treated as unmovable: its position is locked automatically.

When to use an event instead of a lesson

Use a **one-off event** instead if the need is not a timetable lesson: a meeting, a simple booking, room maintenance, a special day with no teaching volume to account for.

Keep a lesson when the volume must remain attached to a course, a subject, a class, a teacher, a group or a teaching dashboard.

Off-grid class

The off-grid class serves programs whose lessons must all be stored with precise times, independently of the site's grid. It is configured when creating or editing the class, only in calendar-type timetables.

Typical case: a continuing-education program uses the same rooms and the same teachers as initial training, but runs on very different times. The class can then be configured off-grid, with a default time step, for example a quarter of an hour.

In this mode, the class's lessons carry their start and end times directly. They **reserve** the teachers, rooms, classes, groups and resources involved: an overlap with another lesson is reported as a **conflict** — not blocking for manual placement, but automatic generation will not place anything on a conflicting slot.

Custom durations

A lesson can also carry custom durations:

- **Calculated duration:** duration deduced from the start and end times;
- **Actual duration:** duration used first in dashboards;
- **Accounted duration:** duration used for billing or the teacher's pay.

Example: for an exam, you can reserve the room before and after the test, while counting only the actual duration of the exam in the Dashboard. For a supervisor, the accounted duration can be higher if the institution also compensates for grading.

Points to watch

- A lesson with custom times can overlap a break or a normally free slot; check that this is really the intention.
- Since the lesson is locked for generation, it can strongly constrain automatic placements around it.
- An off-grid class is a structural choice: it must be selected before creating the class's lessons in bulk.
- **Complex lessons** (alternations, concatenations, associations) are **disabled** as soon as explicit times or an off-grid class are involved: the feature is not available in that case.

How-to

Creating a lesson with custom times

1. Place or open the lesson in the timetable.
2. Open the lesson's custom times for editing.
3. Enter the start time and the end time.
4. Check conflicts on the teachers, rooms, classes, groups and resources involved.
5. Run generation again if needed: the lesson will stay locked and the algorithm will place the rest around it.

See also

Time grid, time slots and durations

General settings

Sites, time grids, classrooms, resources



Automatic generation


One-off events

3.20 Locking the position of a lesson

Source: <help/en/timetables/lesson-lock.md> · id: [timetables.lesson-lock](#) · Audience: admin · Updated: 2026-06-13

Locking a lesson means locking its **position** in the timetable. The lesson is still taken into account by the solver, but the automatic generation can no longer move it.

The visual cue to look for is a **closed padlock** . Conversely, an **open padlock**  indicates that the position can still be moved by a generation or a reorganization operation.

Not to be confused: the **placement** icon  is used to place a lesson on the timetable. It opens or activates placement; it is not the lock marker.

When to lock a lesson

Typical cases:

- **External instructor** whose schedule is imposed.
- **Dated exam** whose time slot or room is already settled.
- **School trip or visit** scheduled on a very specific time slot.
- **External room with imposed hours** (gym, swimming pool...): possible through locking, but it is often better to use the room's **time constraints**, which guide the solver without freezing each lesson one by one.
- **Lesson of a free group** placed manually next to a main lesson.
- **Position obtained by a previous generation** that you want to keep before running a broader generation.




How to lock

1. Place the lesson on the timetable if it is not there yet.
2. Open the lesson's position details.
3. Enable the **closed padlock** to lock the position.
4. Check that the lesson shows a padlock on its position.

To release the position, use the **open padlock**. The lesson can then be moved by a generation or by some reorganization operations.

Mass actions

In the hours distribution, the **Actions** menu also offers global operations:

-  **Lock all scheduled lessons** locks the lessons already placed;
-  **Unlock all scheduled lessons** removes the lock without removing the position;
-  **Cancel all scheduled lessons** removes the lock and removes the position.

This last option is useful before a clean new generation, when you want to start over from unplaced lessons.

Effects on generation and the diagnostic

A locked lesson remains a strong constraint for the solver. The other lessons must fit around it.

An important consequence: if you lock a lesson on a highly constrained time slot, Omniscol respects that decision. Conflicts or warnings remain visible, but the lock indicates that the position is intentional. In the diagnostic, a problem carried by a locked lesson is therefore downgraded from a blocking alert (red) to a warning (orange): Omniscol reports the issue without imposing it as a blocker to fix.

Not to be confused

- **Locking a lesson's position:** the lesson keeps its time slot during the generation.
- **Collaborative lock:** a mechanism related to simultaneous editing by several administrators. See [Real-time collaboration](#).
- **Position locked with no room:** a special case where a lesson must remain without an automatically assigned room, via [Lock](#).

Good practice

- Before running a new generation, review the locked lessons: too many fixed positions reduce the room for optimization.
- After a testing phase, unlock whatever should not be kept.
- For free groups, place the lesson manually and then lock its position. See [Free groups](#).

See also

[Automatic generation](#)[Diagnosing a failed generation](#)[Free groups](#)[Detecting and resolving conflicts](#)[Solver](#)

3.21 Detecting and resolving conflicts (at generation time)

Source: [help/en/timetables/conflicts.md](#) · id: [timetables.conflicts](#) · Audience: admin · Plan: standard · Updated: 2026-06-23

Omniscol detects conflicts **continuously** during the **preparation** of a timetable and at every **manual placement**. The diagnostic distinguishes **blocking conflicts** from **unmet constraints** and from **warnings** that call for a decision, and it encodes this severity with a **color** — on the candidate slots and the selection lists when you place a lesson, and on each row of the diagnostic panel. **Automatic generation** is a separate computation (the solver places under constraints, or leaves a lesson unplaced): its diagnostic is covered in [Diagnosing a failed generation](#).

What the diagnostic checks

- **Occupancy (double-booking)** — a teacher, a class, a group, a student, a classroom or a resource cannot serve two overlapping lessons, unless explicitly modeled (compatible groups, divisions, large rooms).
- **Availability and time constraints** — a period declared unavailable, or an unmet time constraint (teacher, class, group, classroom, subject, site), graded by the priority of the constraint. An imposed date window, however, is always blocking.
- **Capacity and service** — a classroom over capacity, a resource requested in more units than available, an unsuitable classroom (capacity, specialization, site), or a teacher's service hours exceeded.
- **Grid overflow** — the lesson hours to place exceed, or approach, the number of open slots in the grid, for a teacher or for a whole class.
- **Locations and travel** — insufficient travel time between two sites for a teacher or a class, or two subjects declared incompatible placed at the same time.
- **Calendar** — a lesson placed on an absence (teacher or class), a closed day / holidays, or an incompatible event.
- **Modeling** — a broken alignment (aligned groups without a shared lesson), a different number of lessons between groups of the same division or the same alignment, a missing teacher or classroom, or an incomplete configuration (site, grid, a setting required before generation).

Severity levels

The same severity scale — green, yellow, orange, red — colors several places of the editing screens.

On the candidate slots, when you position a lesson (see [Manual placement](#)). **And on each option of the selection lists**: when you choose a teacher, a classroom, a resource, a group or participants for a lesson, each

entry carries a calendar icon colored by its level at the target slot — green if the entity is free, red if it is busy — with the details on hover. Only a **red** unavailability (busy, closed period) prevents the choice; a classroom that is too small or wrongly specialized remains selectable, with a warning.

Color	Meaning
green	no conflict — the entity is free (or a resource stays under its limit)
yellow	minor, non-blocking inconvenience (medium-priority time constraint, fully substituted absence, acceptable but not ideal classroom)
orange	strong constraint (high-priority time constraint, classroom filled up to 110% of its capacity, service hours up to 110%, travel between sites too short, incompatible subjects, closed day)
red	blocking conflict (double-booking, mandatory time constraint or imposed date window, classroom beyond 110% of its capacity, uncovered absence, broken alignment)

In the diagnostic panel. Each listed problem carries a level, reused by the filter. The panel groups orange and yellow under a single “important” level, and adds an “advice” level for non-blocking information:

Level	Meaning	Matching slot colors
red	Blocking conflict	red
yellow	Unmet constraint / Warning	orange + yellow
blue	Advice / Near limit	grid pressure, divisions

When the Block conflicting choices (hard constraints) option is active, Omniscop prevents choices that would create a strong constraint; otherwise, the alert stays visible so the administrator can decide.

Grid overflow and pressure

Omniscop permanently compares the **lesson hours to place** with the **number of open slots** in the grid — for each teacher (per class and per site) and for each whole class. The message shows both values, for example “18 h / 16 h”. Three thresholds, depending on the fill level:

Threshold	Fill level	Reading
blue — pressure	grid filled at 90–100%	the limits will soon be reached; placement is already tight
orange — slight overflow	100–110% (hours > slots up to 110%)	a valid timetable will <i>probably</i> be impossible
red — heavy overflow	beyond 110%	a valid timetable will <i>almost certainly</i> be impossible



This computation only runs if at least one site is configured, and reports nothing as long as the grid stays under 90%. It measures the required hours against the available slots — not to be confused with a teacher's **service hours overrun** (taught hours against contractual hours), which shares the same colors but remains a separate detection.

All detections

Detection	Filter family	Level
Teacher, class or group already busy	Conflicts	red
Classroom already occupied	Conflicts	red
Broken alignment (aligned groups without a shared lesson)	Conflicts	red (orange if only the classroom or the resource differs)

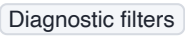
Detection	Filter family	Level
Unmet time constraint (teacher, class, group, classroom, subject, site)	Time constraints	mandatory → red, high → orange, medium → yellow (date window → always red)
Classroom over capacity	Capacity	orange up to 110% of the capacity, red beyond
Resource requested beyond its count	Capacity	orange
Unsuitable classroom (capacity, specialization, site)	Capacity	yellow — can be forced
Teacher's service hours exceeded	Capacity	orange up to 110%, red beyond
Grid overflow / pressure	Grid overflow	blue → orange → red (see above)
Insufficient travel time between sites (teacher, class)	Distance	orange
Incompatible subjects at the same time	Compatibility	orange
No classroom available among the allowed classrooms	Missing classroom	yellow (red if none)
Missing teacher (lesson without an assigned teacher)	Missing teacher	yellow
Missing classroom (lesson placed without a required classroom)	Missing classroom	yellow
Different number of lessons between groups of a division	Divisions	blue (informational)
Different number of lessons between aligned groups	Alignments	red (blocking configuration)
Teacher absent on their lesson	Absences	not covered → red, substituted → yellow, partial (co-teaching) → orange
Class absent	Absences	red
Lesson on a closed day / holidays	Calendar	orange (sometimes forced to red)
Incomplete configuration (site, grid, setting)	—	blocking before generation

Navigating from an alert

Each row of the panel is **clickable**: the magnifier  leads directly to the lesson, group or class concerned — the fastest way to isolate the context of a conflict. For a **group** conflict (division, alignment), the  icon keeps only the related groups, which makes the constraint stand out when the raw diagnostic is not enough.

PREMIUM

Filtering the diagnostic

The  button (warning triangle, at the top of the panel) opens a menu to keep only the alerts useful to the current analysis. Its icon turns orange while a filter remains active. It lets you set:

- **Levels** — show or hide red (blocking), yellow (important / to check), blue (advice / near limit).
- **Entities** — classes, teachers, classrooms, resources, groups, and the calendar event types (events, holidays, absences).
- **Class levels, campuses, sites** — restrict to a scope. A campus isolates an organizational entity (branch, faculty, division); a site refers to the physical location.

- **Reasons** — enable or disable each detection family: time constraints, conflicts, capacity, distance, missing teacher, missing classroom, compatibility, divisions, alignments.
- **Interval** (calendar timetables) — limit to the next weeks or months, and hide the past.
- **Behavior** — Block conflicting choices (hard constraints) actively prevents placements that would create a strong conflict.
- **Reset** — show everything again.

A filter does not resolve a conflict: it only hides what does not help the current analysis.

Resolving a conflict

The most frequent fixes are:

- moving a lesson;
- changing the teacher, classroom or resource;
- adjusting a constraint that is too strong;
- correcting an unavailability or an absence entered by mistake;
- creating or fixing the groups, divisions or alignments;
- adding a compatible classroom, a capacity or a specialization;
- locking a lesson that must stay fixed before rerunning a generation.

If generation does not find a complete solution, it can return a partial timetable: the unplaced lessons remain in the sticky notes bar of lessons to place. Fix the priority diagnostics, then rerun or manually position the remaining cases.

Cases that call for a decision

Some signals correspond to a management decision rather than a data-entry error. Examples:

- an exam with a main room and an extra-time room;
- a capacity declared above the seats entered when the institution knows that some of the enrolled will not be present;
- an annex room not modeled in Omniscol.

In these cases, it is better to model the situation explicitly when possible: multiple rooms, adapted groups, a corrected capacity or a separate event. Leaving a deliberate alert must remain a decision known to the team, not permanent noise.

How-to

1. Open the timetable's diagnostic.
2. If needed, filter by level, campus, site, entity or problem type.
3. Handle the blocking conflicts first.
4. Open the details of the problem to identify the lessons and resources involved.
5. Fix the lesson, the constraint or the modeling.
6. Rerun the diagnostic or the generation to check that the problem is gone.

See also

[Conflict](#)

[Diagnostic](#)

[Campus](#)

[Automatic generation](#)

[Manual placement](#)

[Diagnosing a failed generation](#)

3.22 Diagnosing a failed generation

Source: <help/en/timetables/diagnosing-generation.md> · id: `timetables.diagnosing-generation` · Audience: `admin` · Updated: 2026-06-24

Generation "fails" when the solver cannot **fit everything in**: it returns the best partial timetable it found, and the lessons it had to **sacrifice** remain in the bar of sticky notes to place (a red banner summarises the blockage). The problem is almost always **over-constraint**: too many lessons for the open time slots, or constraints too tight to fit together. The remedy is to **loosen** what can be loosened — that is what this page is about. To understand how the solver prioritises and what it sacrifices, see [Automatic generation](#); for the catalogue of conflicts on an already-built timetable, see [Conflicts and diagnostic](#).

The diagnostic is a lead, not a verdict

When a lesson cannot be placed, Omniscol highlights **the entity that caused the most problems** across its attempts — "this teacher does not have enough time slots", "this class does not fit in its grid". **This is a heuristic**: the reported entity is the one that comes up most often in the blockages, not necessarily *the* single cause. Constraints are **interdependent** — loosening elsewhere can unblock everything. So take the message as a **direction to investigate**: the real remedy is often to **ease the overall over-constraint** (a few more time slots, less strict availability) rather than fixating on the entity pointed out.

Most frequent causes

1. Availability too restrictive

A teacher has marked so many periods as **impossible** that not enough time slots remain to place their lessons.

Diagnosis: examine the availability screen of the teacher concerned. Count the free time slots and compare with their teaching hours.

Resolutions:

- Relax some periods (turn black into red — undesirable but not impossible),
- Redistribute the courses (assign part of them to another teacher),
- Talk with the teacher if the availability is negotiable.

2. No compatible classroom

A subject requires a **specialisation** for which no classroom has been created, or all the matching classrooms are occupied on the only possible time slots.

Resolutions:

- Create the missing specialisation on a suitable classroom,
- Reduce the specialisation constraint,
- Add a classroom.

3. Insufficient capacity

The headcount of a group / class exceeds the capacity of all the candidate classrooms.

Resolutions:

- Assign a larger classroom,
- Assign [several classrooms](#) to the same course (the total capacity is the sum),
- Split the class / group.

4. Structurally impossible alignment

The hours of the aligned groups differ, or the parent classes have incompatible time grids, or an aligned teacher can only be in one place.

Resolutions:

- Harmonise the hours across the aligned groups,
- Unalign and create one course per class,
- Check that the sites of the aligned classes share a common time grid on the target time slot.

5. Too many lessons for the grid

When a class or a teacher obviously has more hours than open time slots, a **warning flags it upstream** (tab **Generation**, hours vs time slots counters). The tricky case is more insidious: taken in isolation, no obvious overflow, but the **accumulation** of constraints between teachers and classes — and their **interplay** — eventually no longer fits.

Resolutions:

- Reduce the hours entered,
- Extend the time grid (open Wednesday afternoon, for example),
- Relax a few unavailability periods, targeting the **periods that many entities avoid at the same time**: the accumulation saturates the grid.

6. Too many incompatibilities

Stacking "no X after Y" constraints can create a system with no solution.

Resolution: replace some incompatibilities with softer **pedagogical weights**.

7. Not enough classrooms (bottleneck)

Very frequent: even if each classroom is suitable, their **number** is not enough to absorb the demand at the same moment. Lessons pile up on the same time slots and the shortage of classrooms becomes the **bottleneck** that keeps everything from fitting.

Resolutions:

- Add classrooms (or share some between classes),
- Spread the demand by widening the time grid,
- Reduce the constraints that concentrate lessons on few time slots.

8. Several sites: travel eats into the hours

With several sites, the **travel times** between locations reduce the time slots a teacher (or a class) can actually use: changing site costs time, which is no longer available to place a lesson. Effective availability tightens and the puzzle gets harder.

Resolutions:

- Group a teacher's lessons on the same site within the day,
- Check the declared travel times between sites,
- Limit back-and-forth trips between sites within the same day.

Checklist before relaunching a generation

- Do all classes have an assigned site?
- Do all classes have their subjects and their teachers?
- Do all groups used in courses have a theoretical headcount or a manageable capacity?
- Is the availability of all teachers validated?
- No critical (red) alert on the **Generation** tab?


Built-in diagnostic tools

- **Generation tab** — *upstream*: statistics (courses / hours per teacher, per class, per site) and warnings (hours vs time slots, critical alerts to fix before launching).

- **Unplaced lessons** (sticky notes, on the right) — *after a failure*: the output of an incomplete generation. The placement button (pin) on each one shows, through the colored placeholders, why no spot works.
- **Detection of forgotten groups**: groups defined but with no course assigned.

☰ How-to

Diagnosing a failed generation

1. **When generation fails**, Omniscol shows a red banner, keeps the partial timetable visible, and puts the lessons it could not fit in the **sticky notes bar**, on the right.
2. **Read the banner**: it gives a **direction** (a heuristic, see above), not a verdict.
3. **Start from the unplaced lessons, not from "conflicts"**. Generation never creates a conflict: it simply sets aside what does not fit. Take a lesson from the sticky notes and click its placement button (pin,  Place on timetable).
4. **Read the colored time slots**. Omniscol shows all the candidate time slots; for a lesson that is impossible to place, they are all blocked. Look for **why**, in the order of the three usual blockers: the **teacher** busy or unavailable, the **class** (or the group) busy, or **no classroom** free and compatible. The reason appears when hovering over the time slot.
5. **Loosen the blocking constraint** — depending on the case: relax availability, add or free up a classroom, harmonise an alignment, widen the grid (see the causes above) — then relaunch.
6. **Before relaunching**, go through the checklist above (sites assigned, subjects and teachers in place, headcounts entered, availability validated, no critical alert on the **Generation** tab).

🔗 See also

[Automatic generation](#)

[Conflicts and diagnostic](#)

[Diagnostic](#)

[Conflict](#)

[Special cases and advanced configurations](#)

3.23 Visualize, duplicate, reorganize a timetable

Source: [help/en/timetables/visualize-duplicate.md](https://help.en/timetables/visualize-duplicate.md) · id: `timetables.visualize-duplicate` · Audience: `admin` · Updated: 2026-06-25

A school often has **several timetables** at different stages: the active timetable for the current year, a draft for the next school year, a test scenario to explore a reorganization. This page summarizes how to manage these multiple versions without confusion, and what the **Visualize** and **Reorganize** screens are for once a timetable has been generated.


View the list of timetables

The **Timetable management** module opens on the list of the account's timetables: active ones and drafts. For each timetable, you see:

- its **label** (`Year 2026–2027`, `Draft school year 2027`, `Scenario Grade 6/Grade 7 merger`),
- its **status** (published or unpublished),
- the **range of weeks** over which it is published (where applicable),
- **quick statistics** (number of classes, number of courses, number of teachers).

The available actions include opening, duplicating or deleting a timetable. Deletion is disabled on a published timetable.

Visualize a generated timetable

The  **Visualize** button opens the timetable in **read-only** mode: you look at the final timetable without being able to modify it. It is the screen of choice to **proofread** before publication, **check** a week or a sequence of lessons,

show the timetable without giving access to the editing functions, **export** or **share**.

Risk-free inspection

The screen uses the usual timetable display engine, but in a **safe** mode: no moving lessons, no saving, no accidental correction. You will find the display filters, the day selector, the various display modes (grid, list, table, schedule overview, day, month, side-by-side — detailed in [Timetable display](#)) and the quick closing of several calendars open side by side.

Alternating weeks and variants




This is particularly useful when the timetable uses **alternating weeks**: you display the **All weeks** view, then the **variants** one by one via the dedicated tabs (A/B, 1/2, etc.). It is often the simplest way to check that a recurring timetable is consistent across all its variants, without the noise of the editing tools.

Share in read-only mode

From this screen, you open the **sharing** function: you distribute the complete theoretical schedule in read-only mode to a third party (external partner, management, site manager), via a web link, without opening access to the Omniscol account. For a calendar-type timetable, an **iCal** link is also available. Details in [Share a timetable via a public link](#).


Export (PDF, Excel, printing)

The Visualize screen is the natural place to produce a file from the displayed timetable:

-  Export data in PDF format to generate a **PDF**,
-  Export data in Excel format to export to **Excel / XLSX**,
-  Print for **printing**.



For class and teacher timetables, this is generally the most direct way to produce a clean document to hand over.

Quickly reorganize a finalized timetable

The  Reorganize button opens the same timetable in **quick edit** mode. The idea is not to redo the whole construction of the timetable, but to quickly correct marginal cases on an already completed schedule.

Typical cases:

- move a lesson that has already been generated,
- change a room, a time slot or a one-off assignment,
- add an isolated lesson,
- correct a few residual conflicts without going back through the whole **hours distribution** → **generation** chain.


On this screen, you will find the grid, the filters, the panel of available time slots, the  Add button and the  Save button to save the changes.

If the change becomes **structural** (classes, groups, subjects, hourly volumes, global constraints), go back instead to the timetable construction steps, or duplicate the timetable to work on a draft.

Duplicate a timetable

Three typical cases:

- **Prepare the next year** from the active timetable — see [Preparing the next school year](#).
- **Create a test scenario** from the active timetable to explore a structural change without risk (merging classes, reorganizing sites, adding a new track).
- **Start again from an old timetable kept in the account** — reuse a structure from a past year as the basis for a new one.

The  Duplicate action opens a dialog. The copy created is always an **unpublished draft**: it only becomes active when you explicitly publish it. The dialog shows a tree of checkboxes **all checked by default**: leaving everything

checked produces an identical copy; unchecking a box removes the corresponding element from the copy.

Choose what is carried over

The tree follows the structure of the timetable; uncheck a branch to rebuild it from scratch:

- **Sites** — with their rooms and their resources. Unchecking the sites also removes the rooms and resources carried by the lessons **and the lesson placements**: the copy then starts from a grid where everything has to be placed again.
- **Teachers** — the timetable's pool of teachers. Unchecking the pool also removes the teacher-class-subject assignments and the teachers carried by the lessons.
- **Classes** — with their groups, their availability, their incompatibilities and the hours distribution. Each sub-branch can be unchecked separately: groups, class availability, incompatibilities, or the lesson details (groups, rooms, resources, modality, comment, positions...).
- **Lesson positions** — unchecked on its own, the copy keeps the whole structure but **the grid is empty**: this is the clean way to clone a preparation in order to rerun a generation from scratch.

Dependencies are applied automatically: you cannot keep a room on a lesson while removing all the sites, for example. And whatever level of detail is unchecked, Omniscol cleans up references that have become orphaned — a class that pointed to a removed site, for example.

Convert the timetable type

PREMIUM

The same dialog can **convert** the copy to a type different from the original. The selector only offers the two **other** types. Converting re-fits the lessons onto the new representation.

- **To a calendar** (from a weekly or a cyclic timetable) — you specify a **period** (start and end dates). Each recurring lesson is **expanded into one dated lesson for each matching working day** of the period; public holidays, closures and absences are skipped. The hourly volume of each subject is adjusted to the number of weeks in the period, and alternating weeks are resolved date by date onto the right variant (A/B, A/B/C...).
- **To a cycle** — you specify the **cycle length** in working days; each position becomes a day number within the cycle.
- **To a weekly timetable** (from a cyclic or a calendar timetable) — each lesson falls back onto its **day of the week**.

Converting to a weekly timetable or a cycle is deliberately simplifying: several lessons that fell on the same day of the week are grouped onto the same time slot, and the dated availability of a calendar is not carried over.

Reserve conversion to a calendar for cases where you really want explicit dates.

PREMIUM

Shift the dates (calendar timetable)

For a **calendar**-type timetable, duplication can **re-anchor all the lessons** onto a new start date — handy for replaying the same organization in the following year. You check the shift option, specify the new start date, and Omniscol shifts the period then realigns each lesson.

The principle: Omniscol **preserves the succession of teaching days**. The Nth teaching day of the original period becomes the Nth teaching day of the new period — the realignment follows working days, not the raw calendar date.

A lesson can therefore **change its day of the week**, for two reasons. First, because the new start date does not necessarily fall on the same day as the old one: if the original starts on a Monday and the target on a Wednesday, everything slides by two days. Second — and this is the most frequent case — because the public holidays and closures of the two periods do not coincide. Lessons that fall outside the new period **revert to unpositioned**: the lessons and all their details remain, only the placement is lost.


Reorganize a school in Omniscol

For a structural reorganization (moving from an organization based on traditional classes to a group-based mode, merging year levels, opening a new site), work on a dedicated **draft timetable**:

1. Duplicate the active timetable to `Reorganization scenario 2027-2028`.
2. Modify the structure (classes, groups, sites) in the draft.
3. Run the diagnostic and the generation to measure feasibility.
4. Iterate without touching the production account.
5. When the scenario is validated, you can switch it to publication.

The advantage: as long as the scenario remains a draft, end users (student, teacher) see nothing. Only administrators have access to the draft.


Delete an unneeded draft

An unpublished timetable can be deleted from the list. To keep a record before deletion, first open the timetable then export it to JSON via . Details in [Automatic generation](#).

A published timetable should not be deleted directly: first remove its publication in the timetable allocation grid if you really need to take it out of operation.

How-to

Create a test scenario from an active timetable

1. **To explore a reorganization** (merging classes, a new site, switching to groups) without touching the production account: work on a **scenario draft** duplicated from the active timetable. Users see nothing as long as it remains a draft.
2. **Open the Timetable management module**: the list shows the account's timetables with their published/unpublished status, their ranges of weeks and their quick stats (classes, courses, teachers).
3. **Click  Duplicate** on the active timetable. Dialog: check what is carried over — **sites** (rooms, resources), **teachers**, **classes** (groups, availability), **courses** and their details. For a scenario, you often keep the structure and the courses, unchecking the teacher assignments to be redone.
4. **Give it a meaningful label**: `S1 26-27, T3 2027, New Paris site test, Option groups reorg`. The more explicit the label, the less likely you are to mix things up in 3 months.
5. **Modify the structure in the draft**: create or merge classes, add the site, reorganize the groups. **Run the generation** to measure feasibility. Iterate at your own pace — zero impact on the active timetable.
6. **If the scenario is validated**, switch it to publication (see [Publication](#)). Otherwise, keep it as a draft or export it to JSON before deletion.

See also

[Overview of the Timetable management module](#)

[Timetable display](#)

[Share a timetable via a public link](#)

[Preparing the next school year](#)

[Publication of a timetable](#)

3.24 Preparing the next school year

Source: <help/en/timetables/next-school-year.md> · id: `timetables.next-school-year` · Audience: `admin` · Plan: `standard` · Updated: 2026-06-25


Building the timetable for the next year can start six to nine months in advance, alongside day-to-day operations. The idea is to duplicate the active timetable to create a draft, then work on that draft without modifying the published timetable.

Three elements to carry over

Three families of data to carry over from the current year to the next one:

1. **Structure** — sites/rooms, classes, groups, divisions, alignments, groups of groups. Often 80-90% reusable from one year to the next, with adjustments at the margins.
2. **Lessons** — the list of courses to schedule (subject, class, teacher, duration, number of weekly occurrences). 70-80% reusable, allowing for curriculum changes.
3. **Teacher-class-subject assignments** — who teaches what. More volatile (departures, arrivals, changes in service hours).

Starting the duplication

From the Timetable management module, use the [Duplicate](#)  action on the timetable that will serve as the base. The dialog checks everything by default; untick what you prefer to rebuild:

- **Sites** — with their rooms and resources.
- **Teachers** — the assignment of teachers to the timetable; their availability follows. Depending on your policy, for a recurring timetable, you may prefer to ask them again for their availability for the new year rather than reusing the old one.
- **Classes** — with their courses, groups, alignments, availability and incompatibilities.
- **Lessons** (hours distribution) — with the details to keep or not: teacher assignments to lessons, groups, rooms, resources, positions. Often worth reviewing, since teachers and students will change.

The draft created is not published: it does not affect the active timetable for the current year. For weekly or cyclic timetables, publication on the weeks of the next year is then done in the hours distribution grid. The duplication can also, along the way, **shift the dates** of lessons (for a calendar timetable) or **convert the timetable type**: these two options and their effects — notably lesson realignment and regrouping — are detailed in [Visualize, duplicate, reorganize](#).

Working on the draft

The prepared timetable appears in the list as a draft. Rename it with an explicit label, for example `s1 27-28`. You can:


- open it to modify classes, add the new teachers, remove the departing ones,
- ask current teachers for their availability for the new year (bulk sending via the Administration module),
- run automatic generation to check feasibility,
- iterate at your own pace for months.

Publication at the start of the school year

When the draft timetable is ready and the new school year arrives, you **publish** it on the desired weeks. See [Publication](#). The draft becomes the active timetable, and the old timetable remains available as history.

☰ How-to

Preparing the N+1 school year

1. **Start N+1 alongside day-to-day operations:** Omniscol isolates the draft from the current timetable. You iterate with peace of mind for months without breaking anything.
2. **Prerequisite:** create school year N+1 in [Administration](#) → [School year](#) with its dates and holidays. It will be needed when publishing on the weeks of that year.
3. **In Timetable management**, use the [Duplicate](#)  action on the reference timetable. Everything is checked by default; untick what will be rebuilt:
 - **Sites**, rooms and resources.
 - **Teachers** assigned to the timetable (their availability follows) — often worth reviewing, or even requesting again for the new year.
 - **Classes**, groups and alignments — 80-90% reusable.
 - **Lessons** and their details (assignments, rooms, positions...) — worth reviewing, as teachers and students will change.
4. **The draft is created.** Rename it clearly, open it, then adjust the classes (arrivals, departures, mergers), add the new teachers and remove the departing ones.
5. **Request availability** from current teachers via bulk sending from Administration. **Run automatic generation** to check feasibility as early as possible — iterate on constraints, volumes, teachers.
6. **When you are ready at the start of the school year, publish** the draft on the desired weeks (see [Publication](#)). The draft becomes the active timetable for N+1, and the old year-N timetable remains available as history.

⚠ Do not forget the **right arrow** of the timeline, or switching the year via the selector, otherwise you stay on year N (see [Timeline and time navigation](#)).

📌 See also

[School year and holidays](#)

[Overview of the Timetable management module](#)

[Publishing a timetable](#)

[Visualize, duplicate, reorganize](#)

4. Dashboard

4.1 Overview of the Dashboard module

Source: <help/en/dashboard/overview.md> · id: `dashboard.overview` · Audience: `admin/teacher/staff` · Plan: `standard` · Updated: 2026-06-25

The **Dashboard** module (📊 icon in the left menu) gathers the statistics calculated on the operational timetable (lessons actually placed and published, taking [absences](#) and [substitutions](#) into account).

It is a **reporting** and **management** tool: how many hours each teacher has taught, what the classroom occupancy is, how many teaching hours per subject per class, how many days of attendance per student.

Available indicators


Six analysis dimensions, accessible through the module's tabs:

- **Teachers** — subjects taught, number of classes, hours counted, overtime hours, cancelled hours.
- **Classrooms** — hours occupied, opening days, hours per day, students accommodated.
- **Resources** — hours of use, days of use, quantity per day.
- **Subjects** — lesson hours, days, lessons, breakdown by class.
- **Classes** — total hours (with the share per group), days, breakdown by subject and by teacher.
- **Students** — lessons, hours, days of attendance, breakdown by subject.


Analysis period

Filterable by **week**, **month**, **school year** or **custom date range**. The timeline at the top is used to navigate through time.

Export

The  **Print** button opens a table that can be copied, printed or exported as CSV. To pass these figures on, see [Sharing link](#). An iCal link makes no sense here: the Dashboard only contains statistics, not dated lessons.

Sharing

The  **Sharing** button generates a public read-only link to the statistics of the period. Useful for external auditors, academic coordinators without an Omniscol account, etc.

📌 See also

[Dashboard](#)[Status](#)[Viewing and filtering](#)[Timetable management](#)

4.2 Using tables and charts

Source: <help/en/dashboard/tools-and-filters.md> · id: `dashboard.tools-and-filters` · Audience: `admin/teacher/staff` · Plan: `standard` · Updated: 2026-05-18

The **Dashboard** tabs combine a table and charts. The table is used to check the details; the charts give a quick view of the distribution or the extreme values.

Search and charts

The search filters the table rows. The charts are recalculated on the visible rows: if you search for a subject, a site or a teacher, the charts reflect only that scope.

The search ignores case, accents and extra spaces. Several words in the same search are interpreted as criteria that must all be met. A comma lets you search for several alternatives.

Examples:

- `math durand` displays the rows that contain both `math` and `durand`.
- `amphi, laboratoire` displays the rows matching either term.
- `site nord >50` can be used, in a classroom list, to isolate the classrooms of the north site whose capacity exceeds 50 seats.

Column sorting

Sortable column headers let you reorder the table: alphabetical order for names, numerical order for counters, time order for durations. Click the same header again to reverse the sort.

Sorting is useful before an export: for example, you can bring to the top the teachers with the most overtime, the most occupied classrooms or the classes with the largest volume of hours.


Copying a chart


To reuse a chart in a document, an email or a presentation:

1. filter the table if needed;
2. check that the chart displays the right scope;
3. right-click the chart;
4. choose **Copy image** in the browser.

The copied chart matches the state displayed on screen.

Export or share

The  **Print** button opens the copy, print or export options available for the displayed table. Depending on the tab, the export can be used to prepare an Excel file for accounting, to send a statement to an academic coordinator or to keep a record for audit purposes.

The  **Sharing** button creates a read-only share link. The recipient views the statistics for the period without being able to edit the timetable.

See also

[Search and filters](#)

[Teacher statistics](#)

[Classroom statistics](#)

4.3 Teacher statistics

Source: [help/en/dashboard/teachers.md](#) · id: `dashboard.teachers` · Audience: `admin/teacher/staff` · Plan: `standard` · Updated: 2026-06-13

The **Teachers** tab consolidates the hours actually accounted for over the displayed period. It is used for monitoring service hours, for payroll preparation, for accounting exports and for checking substitutions.

Main indicators

- **Accounted hours**: the volume retained for administrative monitoring and, depending on how the school is organized, for payroll.
- **Overtime** (Overtime): hours coming from a substitution in the Absence management module or from a dated lesson added manually to a recurring timetable.
- **Canceled lessons** (Canceled lessons): hours removed by an absence or by the manual deletion of a dated lesson from a recurring timetable.
- **Classes and subjects**: the teacher's teaching scope over the period.
- **Lesson types**: breakdown of the volumes per lesson type when lessons are typed.

Overtime and canceled hours make it possible to distinguish the planned service from the service actually delivered over the period.

Breakdown by lesson type

The breakdown by lesson type is useful when hour accounting depends on the teaching format: lecture, tutorial, practical work, support session, workshop, or any other classification configured by the school.

This breakdown helps check payable volumes, internal valuation rules and the supporting evidence needed by the accounting department or an audit body.

Export, sharing and accounting

The table can be copied, printed or exported for processing in Excel or in an accounting tool. The read-only share link lets you send a statement without granting access to timetable editing.

For a recurring connection to accounting software, an ETL pipeline or a dedicated integration can pull these statistics in the format expected by the school.

PREMIUM

On a **Premium** account, the export of this table adds two columns from detailed lesson tracking: **Actual duration** (the duration actually delivered, distinct from the accounted duration) and **Done** (the volume from lessons marked as done). Useful for reconciling planned service, accounted service and service actually delivered. Conversely, lessons in **Draft** or **Canceled** status are excluded from these figures. See [Status](#).

Checkpoints

Before using these figures for payroll, check that the period is correct, that absences have been validated, that substitutions are properly recorded and that the added or removed dated lessons reflect operational reality.

See also

[Tracking and exporting absences](#)

[Status](#)

[Using tables and charts](#)

[Overview of the Dashboard module](#)

4.4 Classroom statistics

Source: <help/en/dashboard/classrooms.md> · id: `dashboard.classrooms` · Audience: `admin/teacher/staff` · Plan: `standard` · Updated: 2026-06-13

The [Classrooms](#) tab analyzes the actual use of classrooms over the displayed period. It is used to spot underused classrooms, overloaded classrooms and imbalances by site or building.

Tracked indicators

In particular, the table shows the number of lessons, the hours occupied, the days of use, the hours per day and the number of students accommodated. When the data is filled in, the site, building, specialisation and tag columns make it easier to analyze by area or by type of room.

The charts help visualize the distribution of occupancy, the hour volumes and the differences in attendance.

Underuse and overuse

A classroom filled well below its capacity can signal a poor assignment, a specialised classroom put to poor use or a possible rebalancing. Classrooms used at less than 25% of their capacity generally warrant a check.

Conversely, a classroom used beyond its capacity signals an operational risk: too many students, a wrongly assigned classroom or capacity data that needs correcting.

Building analysis

The tab helps answer questions such as:

- which buildings account for most of the occupancy;
- which classrooms see little use;
- which room specialisations are missing or in oversupply;
- which sites could be reorganized.

For finer-grained analyses, for example comparing the hours occupied with a building's opening hours or with the possible lesson slots, the school can export the table and rework it in an external tool, or query the account from an external AI agent connected through [MCP](#), depending on the profile, the rights and the exposed tools.

Filtering classrooms

The search accepts capacity comparators (>50 , ≤ 100 , etc.) and site, building, specialisation or tag terms. The charts are recalculated on the filtered classrooms.

See also

[Search and filters](#)[Using tables and charts](#)[Sites, classrooms and resources](#)

4.5 Subject statistics

Source: help/en/dashboard/subjects.md · id: dashboard.subjects · Audience: admin/teacher/staff · Plan: standard · Updated: 2026-05-18

The [Subjects](#) tab shows how the teaching volumes are distributed over the displayed period: overall, by class, by teacher and by lesson type when this information is filled in.

What the page lets you check

- the total volume devoted to each subject;
- the distribution by class;
- the distribution by teacher;
- the share of each lesson type, for example practical work, tutorials, lectures or workshops;
- any teaching formats used by the school.

Pedagogical analysis

This view helps check that the volumes actually scheduled remain consistent with the curriculum plans, the training commitments and the expected distributions between types of teaching.

It is particularly useful for answering questions such as: how many hours of practical work were scheduled, which class receives the most hours in a subject, or which teachers deliver a subject over the period.

Charts and filters

The search lets you isolate a subject, a class, a teacher or a lesson type. The charts are recalculated on the visible rows, which makes it possible to quickly produce a targeted view before copying or exporting.

See also

[Class statistics](#)[Teacher statistics](#)[Using tables and charts](#)

4.6 Class statistics

Source: <help/en/dashboard/classes.md> · id: `dashboard.classes` · Audience: `admin/teacher/staff` · Plan: `standard` · Updated: 2026-05-18

The `Classes` tab consolidates the volumes attended by each class over the displayed period. It is used to check the actual teaching load, the distributions by subject and the supporting evidence requested during accreditations or certifications.

Tracked indicators

In particular, the page lets you track:

- the class's total hours;
- the theoretical and actual number of students when groups are filled in;
- the teaching days;
- the distribution by subject;
- the distribution by teacher;
- the breakdown by lesson type;
- the groups and subgroups involved.

Accreditations and certifications

Class statistics help document the volumes delivered in a program. They can be useful for processes such as Qualiopi or CTI in France, or AMBA, EQUIS and AACSB internationally, when the school must substantiate its teaching volumes, formats and distributions.

Data quality

Accuracy depends on the quality of the classes, groups, subjects, lesson types and teacher assignments in the timetable. Well-maintained groups yield more reliable figures on the volumes actually attended by the students concerned.

See also

[Subject statistics](#)

[Teacher statistics](#)

[Using tables and charts](#)

4.7 Student and resource statistics

Source: <help/en/dashboard/students-resources.md> · id: `dashboard.students-resources` · Audience: `admin/teacher/staff` · Plan: `standard` · Updated: 2026-05-18

The `Students` and `Resources` tabs complement the analysis of the Dashboard. They are more targeted than the teacher, classroom, subject or class views, but remain useful for checking operational data.

Students

The `Students` tab shows the lessons, hours and days attended per student over the displayed period. When classes, groups and subgroups are correctly filled in, the statistics more precisely reflect the volumes actually attended by each student.

Absences are taken into account with dedicated columns, including the number of hours, lessons and days of absence. This view helps spot individual situations that require a check or administrative follow-up.

Resources

The `Resources` tab tracks the use of material resources: hours of use, days of use and quantity used per day. It is mainly used to verify that shared equipment is actually used and to spot periods or resources in low demand.

Working with the data

As in the other tabs, the search filters the visible rows, the charts adapt to the displayed scope and the data can be copied, printed, exported or shared read-only.


See also

[Class statistics](#)[Class and student absences](#)[Using tables and charts](#)

5. Everyday use (Timetable module)

5.1 Viewing and filtering timetables

Source: help/en/schedules/consult-and-filter.md · id: `schedules.consult-and-filter` · Audience: `admin/teacher/student/staff` · Plan: `standard` · Updated: 2026-06-26

The **Timetable** module ( icon in the left-hand menu) is the day-to-day viewing screen. It displays the published timetables for the selected period, with all the useful filters and views.

Available filters

- **Class, teacher, group** (with or without the whole class),
- **Room, resource**,
- **Teachers by subject, pedagogical team** (all the teachers of a class), **subject**,
- **Type of course** (tutorial, practical, exam, lecture...) if configured,
- **Student** (if students are entered and assigned to classes).

You can **stack several timetables one after another**: handy for keeping the class and a particular teacher in front of you, or several classes in parallel.

Views

The display component combines two levels:



- a **rendering mode**: **grid**, **list** or spreadsheet-style **table**;
- a **time organization**: **week**, **day**, **month**, **schedule overview** (and **hourly schedule**) or **side-by-side**.



The grid is the standard calendar view. The table, for its part, presents the current view in a tabular format closer to a spreadsheet; it is not the base view.

See [Timetable display](#).

Timeline and navigation

The **timeline** at the top shows:

- **holidays** as gray horizontal bars,
- the **selected period** (week, month, sometimes school year) as a green area with the dates overlaid,
- **navigation**: left/right keyboard keys, a swipe on a smartphone, or the   buttons.

The  and  buttons at the **far ends** of the timeline let you **switch school years**.

Cache and performance



Omniscol makes intensive use of the browser cache. The application only reloads timetables when you switch weeks — clicking the active week again **forces a refresh**.

This lets Omniscol run even on a 2G (Edge) connection, and keep working through internet micro-outages (requests are replayed once the connection returns).

Edit mode

The  Reorganisation button switches to edit mode: see [Ad-hoc changes](#).

Export / share

-  **Print** — print the current selection (or save it as a PDF through the browser).
-  **Sharing** — open the sharing window: public [web link](#), [iCal](#) subscription or JSON representation (API) depending on your rights.

See also

[Ad-hoc changes](#)[Conflicts and diagnostic](#)[Timetable display](#)[Status](#)[Print and share](#)

5.2 Timetable display: grid, list, table, schedule overview, month, side-by-side





Source: <help/en/schedules/schedule-display.md> · id: [schedules.schedule-display](#) · Audience: [admin/teacher/student/staff](#) · Plan: [standard](#) · Updated: 2026-06-26

This page describes the **shared timetable display** in Omniscol. You find it in **Timetable**, in the **Hours distribution** screen when you switch to the **timetables** view, in the **Visualize** and **Reorganize** screens of **Timetable management**, and more broadly in the other screens that open a timetable calendar with the same display engine.

The same lessons can be read in several presentations without changing the displayed data: you change the display, not the timetable.

Switch the display

The transformation icons are attached to **each displayed timetable**. To reveal them, **hover over the timetable's title**: the title bar then shows the view transformation icons. Click the one you want:

-  for the **grid**,  for the **list**,  for the **table** (restricted to administrators, on desktop);
- on an **hourly schedule**,  **swaps the days and the columns**.

The title bar also carries the sharing and closing of the calendar. Since each timetable keeps its own display, with several stacked calendars you can read one as a list and another as a grid.

Three rendering modes for the same timetable

The first level of switching concerns the **rendering** of the same lessons.

Grid

The **grid** (Grid view) is the standard calendar view: days as columns, time slots or hours as rows, lessons placed in their cells. It is the reference view for reading a timetable week by week.

It is particularly useful for:

- spotting gaps, overlaps and day spans at a glance,
- visually comparing a class, a teacher or a room,
- editing or reviewing a timetable in a familiar time frame.

List

The **list** (List view) displays the lessons one below the other, in chronological order.

It is particularly useful for:

- quickly browsing a long period on a small screen,
- searching for a specific lesson with the filters,
- sequentially reviewing times, rooms and assignments.

Table

The **table** is a **spreadsheet-style rendering** of the current view. It is used to audit, print, copy the data to a spreadsheet or, depending on the screen, launch a file export directly. It is **not** the standard calendar view.

Depending on the context, it can reproduce:

- a days × time slots grid in a more tabular format,
- or one lesson per row when the source display is already a list.

It is particularly useful for:

- administrative checks,
- copy-pasting to spreadsheet software,
- direct export as **PDF**, **CSV** or **Excel / XLSX** when the screen offers it,
- row-by-row verification.

Time scope: week, day, month




The second level of switching concerns the **displayed period**.

- **Week**: the default view in most cases.
- **Day**: the same logic as the standard grid, but tightened to a single day to read a busy schedule in finer detail.
- **Month**: the long-range view. In calendar mode, the display takes the form of a monthly matrix grid; in table mode, you stay on a tabular rendering of the same content.

The selector additionally lets you hide or show specific days in the relevant views.

Schedule overview

A **schedule overview** is composed from the filters menu: it juxtaposes several entities or several days in a single view, to reason about sets. Two variants, depending on the axis you want to read:

- **Add a schedule overview** (icon ) places one **entity per row** (class, teacher, room, subject) and each day's hours as columns — to compare several entities over the same day.
- **Add hourly schedule** (icon ) places the **hours on a vertical axis** (to time scale) and one **column per entity and per day** — to read occupancy and sequences on a true hourly scale. An  button swaps the days and the columns when needed.

The schedule overview is used in particular to:

- display **several entities side by side for the same day**,
- display **several days for one or more entities**,
- read occupancy by month,
- check sequences, transitions or availability.

It complements the calendar grid and answers different questions: choose it to reason about sets of entities or days.

Side-by-side

The Split screen feature opens a **side-by-side view**: two to four timetables displayed next to each other, each keeping its own filters and its own display mode.

It is useful for:

- comparing two classes or two teachers,
- fitting in a shared lesson or a shared supervision duty,
- checking that a free time slot exists simultaneously on both sides.

What stays the same when you switch displays

Switching from one mode to another does not modify the lessons. You keep the same underlying data, with depending on the case:

- the same entity filters,
- the same selected period,
- the same visible conflicts or absences,
- the same print, export or share actions depending on the user's rights.

How-to

Choose the right display for the task

1. **To read a timetable quickly:** start from the **grid**. It is the standard calendar view, the most readable for a week of lessons.
2. **To browse the lessons one by one:** switch to the **list**. Useful on a small screen or to find a specific lesson.
3. **To copy, print, export or audit in spreadsheet mode:** use the **table**. It is a tabular rendering of the current view, not the timetable's base view.
4. **To zoom in on a single day:** choose the **day** scope.
5. **To step back over several weeks or several dates:** switch to **month** or open a **schedule overview** depending on the question at hand.
6. **To cross-read several entities or several days in a single view:** add a **schedule overview** (or an **hourly schedule** for an hourly scale).
7. **To compare two separate timetables:** use the **side-by-side view**.

See also

[Viewing and filtering](#)

[Visualize, duplicate, reorganize](#)


[Timetable display mode](#)

5.3 Ad-hoc changes to a published timetable

Source: <help/en/schedules/ad-hoc-changes.md> · id: [schedules.ad-hoc-changes](#) · Audience: *admin* · Plan: *standard* · Updated: *2026-06-26*

Omniscol is not just a timetable generator for the run-up to the school year — it is also an operational tool for handling day-to-day changes throughout the year.

Turn on edit mode

Click  **Reorganisation**. All open calendars switch to reorganization mode. Newly opened calendars are also directly editable.

Move to the week you want to change (timeline at the top).


Edit a lesson

Click a lesson: a panel appears (you can move it or make it semi-transparent) with all the editable attributes:

- teacher(s),
- group, if any,
- resources,
- room(s) (see [multi-room](#)),
- memos.

Edit several lessons at once


In reorganization mode, **Shift+click** on several lessons selects them together (from two onwards, they stand out highlighted). The panel then becomes a **shared form**: the room, the teacher(s), the resources, the memo or the duration you set there apply to **all the selected lessons** on save — handy for switching a whole block of lessons to another room in one go. See [Editing a lesson](#).

Rather than clicking each lesson, the  **Selection** menu of the form **widens the selection** in a single click:


- by **time window** — the whole **day**, **week** or **month** (depending on the timetable type);


- by **similarity** — the **following similar lessons** (same class, subject, group and time slot: the recurrence of a course);
- by **division** — the lessons of the same group in a division.

Each entry shows the number of lessons it adds, and a toggle restricts the scope to the current calendar or extends it to **all** open calendars.


The neighboring **duplication** menu () copies the lesson(s) — as several copies, or up to a given date on a calendar timetable. Its detailed behavior is described in [Distribute the hours and create the lessons](#).

Detach or delete a lesson

To **detach** a lesson from the grid, click the  icon in the panel or double-click the lesson. It joins the area of unplaced lessons on the right, where you can also edit it.

To **delete** it, open its panel and click  **Delete** : where detaching keeps the lesson in the area of lessons to replace, deleting removes it from the timetable. Like any change, it appears in the list of changes, where you can undo it (see below).

Move a lesson

Open the lesson and click its placement button ( **Place on timetable**), the pin): the possible time slots appear as colored dots (green → red depending on conflicts). Click the new time slot.

To move a lesson between weeks, two options:


- **month view** — display several weeks at once, then reposition the lesson on the desired time slot;
- **detach / switch week / re-place** — extract the lesson, change the active week, place it again.

Difference with calendar timetables

On a [calendar](#) timetable, a change **directly affects the dated lesson** (already unique for that date).

On a [weekly](#) or [cyclic](#) timetable, an ad-hoc change does **not** break the typical week — it is recorded as an exception on the date concerned. The other weeks keep following the typical week.

See and undo changes

The  **View list of changes** button lists the changes in progress over the period. You can undo them individually.

On a **weekly** or **cyclic** timetable, an ad-hoc change is recorded as an exception on its date: it remains undoable **at any time** from this list.

△ On a **calendar** timetable, by contrast, changes are **consolidated permanently on save**: undo them **before saving**.

These changes apply to **a specific date**, not to the structure of the timetable. The structure — classes, groups, **typical week** — is edited when **building the timetable**, not here. Editing a **past** date therefore corrects the history of that single date: do it knowingly.

Conflict detection

Conflict detection works in edit mode just as in the Timetable management module: as soon as you move or edit a lesson, Omniscol checks in real time for double bookings, capacity overruns, time constraints and absences.

On the calendar, a conflicting lesson carries a **double colored outline** — red for a blocking conflict, orange or yellow for a hindrance to arbitrate. The details (conflict type, lessons involved, and for instance a missing room) appear in the **tooltip** when hovering over the lesson.

The full list of conflict types, the severity levels and the diagnostic panel are described in [Conflicts and diagnostic](#).

See also

[Viewing and filtering timetables](#)

[Conflicts and diagnostic](#)

[One-off events](#)

5.4 One-off events (outside the timetable)

Source: [help/en/schedules/events.md](#) · id: [schedules.events](#) · Plan: [premium](#) · Updated: 2026-06-13

PREMIUM

PREMIUM

A **one-off event** is a session that is not part of the regular structure of the timetable: a special meeting, a school trip, a one-day exam, an open day, a class council. Rather than touching the regular structure of the courses, you create a dated event that sits **on top of** the grid over a precise interval.

Creating one-off events is available on **Premium** accounts.

When to create an event vs a course lesson

- **Regular course**: repeats every week (or every other week) throughout the year. Managed in the Timetable management module, included in automatic generation.
- **One-off event**: takes place over a dated interval, with no regular recurrence logic. Managed directly in the Timetable module.

Rule of thumb: if you would write it by hand on a calendar rather than in the weekly grid, it is an event.

Create an event — the events agenda

An event is not entered in a standalone form: you draw it **directly on the agenda**, as in a calendar. On the events grid, **click and drag** to draw an interval within a day (a preview appears while you draw). On release, a form opens, anchored to the selection, with the dates and times **already pre-filled**; you complete:

- **Title** — `Year 7B review meeting, Science museum trip, Mock English exam.`
- **Start and end** — date, start time, end time.
- **Participants** — classes, groups, teachers, students, or **free-text** participants.
- **Location** — a room known to Omniscol or a location entered as **free text**.
- **Resources** — equipment or a resource to book if necessary.
- **Videoconference link** — optional, to attend the event remotely.
- **Comment** — a free note that appears with the event.
- **Color** — a visual aid to tell it apart in the grid.

The mouse selection stays **contained within a single day**.

Where to find the agenda

- **In the Timetable module**. Display the **Events** filter: the agenda is editable (mouse drawing active) only in **reorganization** mode; in simple viewing, it stays read-only.
- **When editing a calendar-type timetable**. The **Events** filter is only offered there for a **calendar-type** timetable; it does not appear on a weekly-grid timetable.

A direct save, alongside the timetable

An event is saved **immediately and separately**, as soon as you confirm its form: creating, moving or deleting an event instantly triggers its own save (confirmed by a notification). You do **not need** the global **Save** button of the reorganization or of the timetable editing — events are already saved. Conversely, saving the reorganization or the hours distribution of a timetable covers the **course lessons**, not the events: it is a parallel subsystem, persisted as you go.

Conflicts with the timetable's lessons

An event references **the same entities** as the timetable — teachers, classes, groups, rooms, resources — which lets Omniscol detect overlaps with regular lessons. As you fill in the form, each participant and each room or resource is checked against what is already placed on the interval:

- for a **teacher**, a **class** or a **group**, the event is compared with course lessons **and with other events**;
- **rooms** and **resources** are checked to spot an occupation already existing on the time slot.

These checks feed the availability hints in the selectors and surface as conflicts in the reorganization view. The event overlays the grid: it **flags** the conflict but does not by itself free the lessons it overlaps. Handle them separately (move, ad-hoc change, absence...).

Participants: everybody, or anyone

Beyond the individually named participants, two special participants exist — and they do not behave the same way with regard to conflicts:

- **Everybody** — the event concerns **the whole school**, on a **mandatory** basis. It appears in all timetables **and conflicts** with the existing course lessons: the one to pick for an open day or an event that genuinely involves the whole school.
- **Anyone** — the event is open to whoever wishes to attend, on an **optional** basis. It also appears in all timetables, but **generates no conflict**: for an optional offer that must not block anyone's grid.

Both appear everywhere; only **Everybody** causes conflicts, because it makes attendance mandatory.

Projection into the timetables

Once created, an event is **projected into every timetable concerned**: it appears in the timetable of each of its participating teachers, classes, groups and students, as well as in that of each selected room and resource (and everywhere, for **Everybody** and **Anyone**). It is the same event seen from several angles — so it is normal to find it at once in the timetable of the class, the teacher and the room; these are not duplicates.

An event has no "type" field: it differs from a regular course by its title, its dates, its participants and its optional color.

Move, snap, free the times

On the events agenda, unlike course lessons, an event is **moved and resized by drag and drop**.

The **time range displayed** by the agenda follows the **opening hours** of the school: it spans from the earliest configured opening to the latest configured closing, and falls back to 07:00 – 20:00 if nothing is defined.

The form also carries a **pin** button: it switches to placement by free time slots, and a click on the chosen slot copies its date and times into the form — an alternative to mouse drawing to snap the event cleanly into place.

Finally, the start and end times are **entirely free**: they are not forced to fall on the grid's time slots. An event can start at 09:50 and end at 11:34 even if the regular courses sit on full slots: that is the whole point of an event laid on top of the grid.

Edit or cancel

An event is edited the same way it was created: move it in time, change the room, add or remove participants.

☰ How-to

Create a one-off event

1. A **one-off event** sits **on top of** the grid over a dated interval: a class council, a trip, a mock exam, an open day, a meeting. No regular recurrence to model.
2. **Click and drag on the events agenda** to draw the interval (in reorganization mode). The form opens with the dates pre-filled; enter the title (`Year 7B review meeting`, `Science museum trip`).
3. **Select the participants**: the classes, groups and teachers concerned, students if available, or free-text participants.
4. **Add a location, resources, a videoconference link and a comment** if necessary. These fields describe the event; they do not transform the regular courses.
5. **Save**. The event appears in the timetables concerned. If Omniscol flags a conflict with regular course lessons, fix the lessons concerned in the Reorganisation view.

🔗 See also


[Off-grid lessons](#)[Ad-hoc changes](#)[Viewing and filtering timetables](#)[FAQ — special cases](#)

5.5 Print and share

Source: [help/en/schedules/print-and-export.md](#) · id: `schedules.print-and-export` · Plan: `standard` · Updated: 2026-06-25

Every timetable you can view in the Timetable module can be **printed** or **shared** in several formats. The choice depends on the recipient: paper for offline use, PDF for distribution by email, iCal for an electronic calendar, web link for direct viewing. For a numeric analysis of volumes in a spreadsheet, the **Dashboard** aggregates the hours and exports them as CSV / XLSX.

Print

The  **Print** button (at the top of the view) opens the print preview with a layout optimized for paper:

- lessons are **hollowed out**: Omniscol prints only their **colored outline** on a white background, saving ink while keeping the visual color cue,
- margins, orientation and size are adjusted to the A4 / Letter format,
- the header shows the school name, the label of the timetable being viewed (student, class, room...) and the date range.

The preview relies on the browser's print system — so everything it can do (native PDF export, printer selection, double-sided printing, etc.) is available.

Export as PDF

Omniscol provides **dedicated PDF export buttons** that lay a timetable out cleanly. The most complete one is in the Visualize screen of Timetable management (see [Visualize](#), [duplicate](#), [reorganize](#)). The **table** view of a timetable (see [Timetable display](#)) also offers a PDF export, alongside CSV and Excel.

Failing that, on most modern browsers, the print dialog can produce a PDF (`Microsoft Print to PDF`, `Save as PDF`, `Print to File`).

Good practice: use the PDF for one-off communications (sending a weekly timetable to a parent, for example) and the share link for recurring communications (the PDF is frozen, the link follows changes).

Analyze in a spreadsheet (CSV / XLSX)

Two paths lead to the spreadsheet. **From the viewing screen**, switch the timetable view to a **table** (see [Timetable display](#)): this spreadsheet-style rendering can be printed, copy-pasted and exported directly as **CSV**, **Excel / XLSX** or **PDF**. **For an aggregated analysis of volumes**, go through the **Dashboard**, which computes the hours per teacher, per subject, per class or per room over the chosen period and offers the same export as **CSV** (universal) or **XLSX** (preformatted for Excel).

Typical uses of this spreadsheet export:

- counting the hours delivered per teacher, per subject, per class;
- feeding an external HR tool (payroll, working-time tracking);
- spotting gaps between planned and delivered service hours.

See [Using tables and charts](#) and [Teacher statistics](#).

To retrieve the lessons themselves in machine format, sharing offers a **JSON (API)** representation; see [Share a timetable via a public link](#).

Export as iCal

To share a timetable dynamically with an electronic calendar (Google Calendar, Apple Calendar, Outlook), use the **iCal subscription** rather than a one-off export. The recipient sees changes continuously. See [iCal — subscription and dynamic link](#).

The sharing window also displays a ready-to-scan **QR code**: the recipient points their phone at the computer screen to subscribe to the calendar without typing an address.


If you still want a frozen `.ics` file for archiving, the one-off export is also available from the sharing window.

Filter before exporting

Exports honor the **active filters** of the current view: if you have filtered on a specific class or date range, the export will contain only that scope. For a broader export (the whole year, all classes), adjust the filters accordingly before launching the export.

☰ How-to

Print or share a timetable

1. **Two outputs from the viewing screen**: printing (PDF) for a frozen distribution, and sharing (Web, iCal, JSON) for a dynamic one. Numeric spreadsheet analysis, for its part, goes through the Dashboard.
2. **Filter the view first** to the scope you want: class, teacher, room, date range, lesson type. Printing and sharing honor the active filters — a narrow filter = a narrow output.
3. **For a PDF** (email delivery, archiving): the most complete option is the PDF export of the Visualize screen (see [Visualize, duplicate, reorganize](#)); from the viewing screen, the  **Print** button opens the print preview (hollowed-out lessons, school header + timetable label + date range), from which you can also produce a PDF.
4. **In the browser's print dialog**, select **Save as PDF / Microsoft Print to PDF / Print to File**. The PDF is frozen: it does not follow later changes to the timetable. For a dynamic output, prefer the **share link** (see [Public share links](#)).
5. **For a spreadsheet analysis**: switch the timetable to a **table** for a direct export (CSV / XLSX / PDF, see [Timetable display](#)), or open the **Dashboard** for the aggregated volumes — hours per teacher / subject / class — then export as CSV or XLSX (see [Teacher statistics](#)).
6. **For an electronic calendar**, prefer the **iCal subscription** over a frozen file: the recipient sees updates continuously (see [iCal](#)).

See also

Timetable display

Visualize, duplicate, reorganize


iCal — subscription and dynamic link

Share a timetable via a public link

Import and export

5.6 Share a timetable via a public link

Source: <help/en/schedules/share-link.md> · id: [schedules.share-link](#) · Audience: [admin/teacher](#) · Plan: [standard](#) · Updated: 2026-05-15

The **Timetable** module offers a  **Sharing** button that generates signed URLs for the timetable being viewed. The recipient opens the web link and sees the timetable read-only, without having to sign in. It is the fastest way to communicate a timetable to someone who has no Omniscol account — typically a student's parent, an external partner or an administrative department.


Formats generated by sharing

Depending on the context and your rights, the sharing modal can offer several URLs:

- **responsive web page** — a browsable, portal-style display, with week / day / list views,
- **iCal .ics feed** — to subscribe from Google Calendar, Apple Calendar, Outlook,
- **JSON (API)** — for developers and machine-to-machine integrations, when this format is allowed.

The recipient picks whatever suits them: a parent will open the web page or subscribe via iCal, a developer will use the JSON.

What can be shared?

Sharing is not limited to the timetable viewing screen: the same  **Sharing** icon appears on many Omniscol screens.


Read-only, you can share:

- the **individual timetable** of a student or a teacher,
- the **timetable of a whole class**, of a **room** or of a **subject**,
- the **staffing grids** of the Staffing module,
- the **dashboard** and the **absence tracking**,
- the **visualization** and **reorganization** screens of a timetable.

Read-write — the one exception — you can share the teachers' (and staff's) **availability entry** screen: the dedicated link lets each person fill in their availability until the expiration date, **without an Omniscol account or password**. It is the only share that allows data entry rather than simple viewing.

Each signed URL carries the requested scope and an expiration date.

Create the link

From the viewing screen, click  **Sharing** in the toolbar. The modal offers:

- **Expiration date** — the date until which the generated links remain valid.
- **Web, iCal and JSON tabs** — visible depending on the formats generated for the current selection.

You then copy the URL you want. For the web and iCal links, the modal can also display a printable **QR code**.

Limit a share

Choose a short expiration date for one-off uses.

A link is carried by the account that generated it. To invalidate the existing links generated by an account, use the real levers:

- change or reset the carrying account's password;
- deactivate or delete the carrying account.

Tip: for shares managed by a planning team, you can use a **clearly identified service account** (for example `diffusion-planning`) rather than a colleague's personal account. That account then carries the responsibility for the links issued; its access and its password must be managed like administrative access.

Web links are read-only, except the teachers' **availability entry** link described above, which allows data entry until the expiration date without giving access to the Omniscol account.

Security

- The link contains a highly secure signed token.
- It gives access **only to the shared resource** (timetable or screen), not to the whole account.
- Validity is limited by the chosen expiration date.
- Links are invalidated if the carrying account's password changes, even if the new password is identical in plain text.
- Links stop working if the carrying account is deactivated or deleted.
- Anyone who has the URL can view the shared scope until it expires or is invalidated.

Difference from a limited Omniscol account


For sharing data between teams, for example with an accounting department, the two options can appear to compete.

- **Signed public link** = no sign-in, simple access, fast sharing.
- **Limited Omniscol account** = dedicated sign-in, longer lifespan, a user account to manage, with the option of [custom roles](#).

Prefer the link for one-off uses and a limited account for structured, long-term uses.

How-to

Create a share link

1. **A share link** gives read-only access to a timetable without a login. Ideal for a parent, an external partner, an administrative department.
2. **Open the timetable to share** in the Timetable module: student, class, room, teacher or subject. The viewing page is the starting point.
3. Click  **Sharing** in the toolbar. A modal opens with the sharing options.
4. **Choose the expiration date.** The more widely the link circulates, the closer that date should be.
5. **Copy the format you need.** Web for direct viewing, iCal for a calendar subscription, JSON if an API integration is planned. The URL is usable immediately: whoever opens it sees the shared scope read-only, without signing in.
6. **If the link spreads where it should not**, create a new, more limited link with another account and invalidate the links carried by the original account by changing its password or deactivating it if appropriate.

See also

[Public share links](#)

[iCal — subscription and dynamic link](#)

[Guest portal \(public links\)](#)

6. Absences and substitutions

6.1 Overview of the Absences module

Source: <help/en/absences/overview.md> · id: `absences.overview` · Audience: `admin/teacher/student/staff` · Plan: `standard` · Updated: 2026-06-13

The **Absence management** module centralizes the declaration and tracking of unavailability that must change how timetables are displayed: teachers, classes, students and, if the Staffing module is enabled, staff members.

Managed entities

Absent entity	Effect on the timetable
Teacher	Lesson canceled when the absent teacher is its only teacher, unless a substitute is assigned (the substitute then appears next to the struck-through regular teacher). With co-teaching , the lesson is kept : only the absent teacher is removed, and the remaining teacher(s) deliver it.
Class	The absent class's lessons are removed from the display over the period concerned.
Student	The lessons stay in place for the class; the absence affects the student's view and tracking.
Staff	Available only when the Staffing module is active; works like the absence of a staff planning entity.

The year's school holidays are not entered as absences: they are managed in the school year.

Supporting documents

The module does not handle **document upload**: timetable management is not meant to store sensitive data (medical certificates, confidential information), it focuses on planning. If you still need to attach a document, store it in your usual storage space and paste the **link in the comment field** of the absence.

Statuses

- **accepted**: the absence is validated and reflected in the displays.
- **pending**: request declared by the user concerned, to be validated by an administrator.
- **rejected**: rejected request.
- **aborted**: request canceled by its requester.

Only absences with the **accepted** status have an effect: they change how timetables are displayed, mark the affected lessons (canceled where applicable) and feed the absence **statistics** and **exports** (see [Tracking and exporting absences](#)). The other statuses — pending, rejected, aborted — have no effect on the timetable.

Screen layout

The module offers one tab per entity type:

- **Teachers**: teacher absences and substitution management.
- **Classes**: class absences, managed by administrators.
- **Students**: individual student absences.
- **Staff**: staff absences, if the Staffing module is enabled.

Each screen shows the absences of the period, their status, the affected lessons when the information is available, time navigation and an exportable list for administrators.

Prerequisites

- At least one timetable published for the school year concerned.
- For teacher absences: teachers must be assigned to courses.

- For student absences: students must be assigned to classes.
- For staff absences: the Staffing module must be active.

☰ How-to

Discovering the Absences module

1. Open the **Absences** module from the main navigation.
2. Choose the tab matching the entity concerned: teachers, classes, students or staff depending on the enabled modules.
3. Use **+ Declare an absence** to open the form: absent entity, dates, time range, reason, comment and status depending on your rights.
4. On an accepted teacher absence, open **Substitute management** to see the lessons concerned and assign single-lesson substitutes or substitution rules.
5. For administrators, the **Table** button exports the displayed table.

🔍 See also

[Declaring an absence](#)

[Substitution policies](#)

[Absence](#)

[Substitution / Replacement](#)

6.2 Declaring an absence (administrator / teacher / student)

Source: [help/en/absences/declaring.md](#) · id: [absences.declaring](#) · Audience: [admin/teacher/student/staff](#) · Plan: [standard](#) · Updated: 2026-06-13

Who can declare?

Role	For whom?	Initial status
Administrator	Teacher, class, student or staff member	accepted by default, editable
Teacher	Themselves only	pending
Student	Themselves only	pending
Staff	Themselves only, if the Staffing module is active	pending

Class absences are administrative operations: creating and modifying them is restricted to administrators.

Form fields


The **+ Declare an absence** button opens the form. Saving is done with **Add**.

- **Entity**: visible to administrators; other roles can only declare their own absence.
- **Date range**: start date and end date. For a one-day absence, use the same date. If the end is not filled in through the interface, Omniscol uses the end of the school year as a technical limit: do not use this to record a genuinely open-ended absence.
- **Time range**: all day by default, or specific time slots.
- **Optional filters**:
 - **Class** to restrict a teacher absence to certain classes.
 - **Subject** to target only the lessons of a given subject.
 - **Assignment** for staff absences, if the Staffing module is active.
- **Reason**: chosen from the list provided; administrators can also enter a custom reason.
- **Comment**: short free text.

- **Status:** only administrators can freely set the status (**accepted** status by default). For other roles, the request takes the **pending** status and awaits validation by an administrator; the requester can however switch their own request to the **aborted** status to abandon it.

How-to

First teacher absence

1. Open the `Teachers` tab of the Absence management module.
2. Click `+ Declare an absence`.
3. Select the teacher concerned if you are an administrator. A teacher declaring their own absence cannot select another teacher.
4. Enter the dates. For a single day, use the same date as start and end.
5. Leave **All day** or define the time slots concerned if the absence covers only part of the day.
6. Choose the reason, add a comment if needed, then save with `Add`.
7. If the absence is accepted and concerns a teacher, open the substitution management with `` `Substitute management` to assign a substitute or define a substitution rule.

Validating a pending request

When a teacher, a student or a staff member declares their own absence, the request is pending. Administrators can accept or reject it from the `✓ accepted` and `rejected ✕` shortcuts, or by opening the record.

Later changes

- An administrator can change the dates, the times, the reason, the comment and the status.
- The requester can modify a future absence that belongs to them; the change puts the request back to the pending status.
- The requester can abandon their own request.

Visibility

The fact that an entity is absent can be visible in the timetables or in the Absences module, depending on viewing rights. The details of the reason, the comment, the creation metadata and the status are not exposed in the same way to all roles: non-administrator users do not necessarily see the details of other people's absences.

See also

[Substitution policies](#)

[Single-lesson substitution](#)


[Absence](#)

6.3 Substitution policies

Source: help/en/absences/substitution-policies.md · id: absences.substitution-policies · Audience: admin · Plan: standard · Updated: 2026-06-13

A **substitution policy** covers several lessons affected by the same teacher absence. It saves you from picking a substitute lesson by lesson when a simple rule is enough.

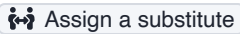
Access

On the row of a teacher absence, open ` Substitute management`.


The panel shows the lessons affected by the absence and the substitution rules already saved. Lessons without a substitute remain marked as affected by the absence of the regular teacher.

Two levels of rules

One-off substitution

On a specific lesson,  lets you pick a substitute for that lesson only. This assignment takes priority over long-term policies.

Long-term policy

In the Long-term substituting assignments section of the panel, the  button creates a substitution rule. For each rule, you fill in:

- the **substitute**;
- a **start date** and an **end date** for the rule's validity;
- **time slots** if the rule does not apply to the whole day;
- optionally **subjects**;
- optionally **classes**;
- an internal **comment**.

Several rules can coexist. They apply in the order displayed; reorder them by drag and drop if the priority changes.

Reading the panel

The list of lessons shows which substitute covers each of them. If a rule matches none of the absence's lessons, the panel displays a warning on that rule: fix the dates, time slots, subjects or classes.


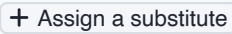

Typical case: splitting a long absence

For an absence lasting several weeks:

1. Create a first rule with the main substitute, without any subject or class filter.
2. Add a second, more targeted rule, for example limited to one subject or one time slot.
3. Place the most specific rule before the general rule if it must take precedence.

How-to

Split a long absence

1. Open the teacher absence with .
2. Click .
3. Pick the main substitute, set the validity dates, then save.
4. Add another rule if some lessons must be covered by a different substitute.
5. Check the list of affected lessons. They should show the expected substitute; lessons not covered remain visible as having no substitute.
6. Confirm the panel with .

See also

[Declaring an absence](#)[Single-lesson substitution](#)[Substitution / Replacement](#)

6.4 Single-lesson substitution

Source: help/en/absences/single-lesson-replacement.md · id: `absences.single-lesson-replacement` · Audience: `admin` · Plan: `standard` · Updated: 2026-06-13

A **single-lesson substitution** assigns a substitute to one single lesson affected by a teacher absence. It is set up in the absence's substitution management panel.

This mechanism does not move the lesson, does not create a notification by itself and does not offer several cancellation scenarios. It adds a substitute to the lesson concerned; the absent regular teacher remains visible as the substituted teacher.

Access

1. Open the **Teachers** tab of the Absence management module.
2. On the absence concerned, click **Substitute management**.
3. In the list of affected lessons, use **Assign a substitute** on the lesson to cover.

The panel shows the lessons covered by the absence and the substitution rules already defined.

Effect of a single-lesson substitution

- The chosen substitute is assigned only to that lesson.
- The classroom, the class, the date and the time of the lesson are not changed by this action.
- The single-lesson substitution takes precedence over the general substitution rules of the same absence.
- If no substitute is assigned, the lesson remains affected by the teacher's absence.

Choosing the substitute

The selection list is based on the teachers available for the time slot concerned. Check manually that the choice is consistent with the subject, the site and the school's internal organization before confirming.

Removing a single-lesson substitution

If a single-lesson substitution was added by mistake, use **X**. The lesson then falls back to the applicable long-absence rule, or remains without a substitute if no rule matches.

How-to

Assigning a substitute to a lesson

1. Create or open an accepted teacher absence.
2. Click **Substitute management**.
3. Locate the lesson to handle in the list of affected lessons.
4. Click **Assign a substitute**, choose the substitute, then confirm with **Assign**.
5. Check the lesson's display: the absent regular teacher must remain struck through and the substitute must appear on the lesson.

See also

[Substitution policies](#)[Declaring an absence](#)[Multi-day absences](#)[Substitution / Replacement](#)

6.5 Multi-day absences

Source: <help/en/absences/multi-day.md> · id: `absences.multi-day` · Audience: `admin` · Plan: `standard` · Updated: 2026-06-13

A **multi-day absence** is an absence declared with a start date and an end date. It covers a continuous period without creating one absence per lesson.

Omniscol then determines which lessons fall within the period. For a teacher absence, these lessons can be left without a substitute, covered by a substitution policy, or adjusted with single-lesson substitutions.

Declaring the period

From the **Teachers** tab of the Absence management module:

- select the teacher concerned;
- enter the start date and the end date;
- leave the whole day or specify the time slots concerned;
- choose the reason;
- add a comment if needed;
- save with [Add](#).

Handling the affected lessons

After the declaration:

- open [Substitute management](#);
- check the list of lessons affected by the absence;
- create one or more rules with [+ Assign a substitute](#) if a substitution logic repeats;
- use [Assign a substitute](#) on a particular lesson for lesson-by-lesson exceptions.

Substitution rules are the mechanism designed for long absences. A rule **with no filter** applies to **all** the lessons of the period; add a class, a subject, time slots or dates to target only a subset. To **leave everything without substitution**, create no rule: the lessons where the absent teacher is the only teacher are then canceled over the period.

Absence on recurring time slots

An absence that affects only certain slots of a long period — for example every Monday morning for two months — is declared **once**: enter a broad date range, then specify the **time slots** concerned (and if needed a class or a subject). Omniscol then selects, across the whole period, only the lessons that fall on those slots: there is no need to enter each occurrence.

Tracking and export

The absences table lets administrators export the displayed rows with [Table](#). For the tracking available in the module, see [Tracking and exporting absences](#).

☰ How-to

Managing a long absence

1. Declare the absence with a start date and an end date.
2. If the absence does not cover entire days, specify the time slots concerned.
3. Save the declaration.
4. Open [Substitute management](#).
5. Add the necessary substitution rules, then check the list of affected lessons.
6. Use single-lesson substitutions only for the lessons that must depart from the general rules.

🔍 See also

[Declaring an absence](#)

[Substitution policies](#)

[Single-lesson substitution](#)

[Tracking and exporting absences](#)

6.6 Class and student absences

Source: <help/en/absences/class-and-student-absences.md> · id: `absences.class-and-student-absences` · Audience: `admin/teacher/staff` · Plan: `standard` · Updated: 2026-06-13

The Absence management module distinguishes absences of **entire classes** from absences of **individual students**. Both follow the same declaration principle, but their effect on the timetable is not the same.

Absence of an entire class

Typical cases: school outing, trip, group internship, exam day.

Class absences are managed by administrators. From the `Classes` tab, the form lets you enter:

- the class concerned;
- the date range;
- the time range if the absence does not cover the whole day;
- optionally a subject to limit the effect to the lessons concerned;
- a reason;
- a comment.

For a recurring timetable, an accepted class absence removes that class's lessons from the display over the period concerned.

Absence of an individual student

Typical cases: illness, medical appointment, family reasons, authorized or unauthorized absence.

A student absence:

- applies to the selected student;
- marks the lessons of the period as **canceled** on the student's timetable, without removing the class's lessons or the teachers' lessons;
- can be declared by an administrator or by the student concerned, depending on their rights;
- becomes pending when it is declared by the student themselves.

Visibility

Administrators have the full view. Other roles see a limited version according to their rights: the existence of an absence can be visible without exposing the reason, the comment or the validation metadata.

How-to

Declaring a class absence

1. Open the `Classes` tab of the Absences module.
2. Click `+ Declare an absence`.
3. Select the class, the dates and the times concerned.
4. Add the reason and, if needed, a comment.
5. Save. That class's lessons are removed from the display for the accepted period; the other classes keep their lessons.

☰ How-to

Declaring a student absence

1. Open the **Students** tab.
2. Select the student concerned if you are an administrator.
3. Enter the dates, the times, the reason and the comment.
4. Save. If the student created the request themselves, it remains pending until it is validated.
5. Once accepted, the lessons of the period are marked as **canceled** on the student's timetable and counted in their tracking and statistics. The other students of the class are not affected.

🔍 See also

Declaring an absence

Tracking and exporting absences

Multi-day absences

One-off events

6.7 Tracking and exporting absences


Source: <help/en/absences/statistics.md> · id: `absences.statistics` · Audience: `admin` · Plan: `standard` · Updated: 2026-06-13


Absence tracking is available in several complementary places: the Absence management module for operational tracking, the **Dashboard** for aggregated statistics, and the Administration module for a student's history. The latter two views only count absences with the **accepted** status.

Tracking in the Absences module

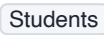
Depending on the tab and the selected period, the screen shows:

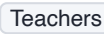
- the entities absent over the period;
- the status of each request, with counters;
- the lessons affected by absences with the accepted status;
- navigation by week, month, school year or date range;
- the tabular list of absences.

For a teacher absence, substitute management ( Substitute management) adds the affected lessons and the assigned substitutes.

Administrators export the displayed table with  Table : entity, dates, times, subject where relevant, reason, comment and status depending on access rights — convenient for an internal report or further processing outside Omniscol.

Statistics in the Dashboard

For **volumes and rates**, open the **Dashboard**. The  Students tab totals, per student and over the period, the **hours**, **lessons** and **days of absence** together with the corresponding **absence percentage**, exportable as CSV / XLSX (see [Student and resource statistics](#)).

On the **teacher** side, the  Teachers tab reflects the effect of absences on service hours: the **Canceled lessons** column counts the hours removed (an absence without a substitute cancels the lesson) and **Overtime** the hours added. Each column shows the number of lessons involved and offers a tooltip that details them (subject, class, day and time); for a lesson canceled by an absence, the reason appears there when it has been entered. This tab can also be exported as CSV / XLSX (see [Teacher statistics](#)).

History in Administration


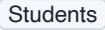
In the Administration module, the record of a **student** and that of a **teacher** both summarize their absences for the school year and offer their own table export — convenient for individual follow-up or for sharing (with the family for a student, with the person concerned for a teacher).

Limitations to keep in mind

- Only absences with the **accepted** status count in the display and the statistics; those with the **pending**, **rejected** or **aborted** status remain for information only.
- Reasons and comments may be hidden depending on the reader's role.

How-to

Produce a tracking export

1. Open the relevant tab of the Absence management module.
2. Select the period to review.
3. Check the statuses and the affected lessons if needed.
4. Click  Table .
5. Use the exported file for your internal checks or further processing.
6. For **volumes and an absence percentage** per student, open the Dashboard,  Students tab (see [Student and resource statistics](#)).

See also

[Overview of the Absences module](#)

[Class and student absences](#)

[Substitution policies](#)

[Student and resource statistics](#)

7. Staffing

7.1 Overview of the Staffing module

Source: <help/en/staffing/overview.md> · id: [staffing.overview](#) · Audience: [admin/staff](#) · Options: [staffing](#) · SKU: [omniscol/staffing-only](#) · Updated: 2026-06-13

👑 OPTION: STAFFING

The **Staffing** module schedules staff **by task** rather than by lesson: thinking in terms of "who covers which post at what time" rather than "who teaches what". It complements the teaching timetables when the institution has to manage student supervision, monitoring duties, study halls, exams or group-supervision activities.

Main use case: student supervision and education assistants

The module was created to meet the long-standing need of student supervision teams: pastoral staff, education assistants, monitors and personnel who cover a multitude of posts over the course of the day, independently of lessons.

Examples:

- supervising the corridors;
- filtering entries to and exits from the school;
- staffing reception;
- supervising the playground during breaks;
- supervising the cafeteria at noon;
- covering study hall and the study room;
- running or supervising the library, the common room or a workspace;
- accompanying students off site, for example for a PE outing entrusted only to accredited people;
- covering one-off between-lesson slots;
- supervising exams, competitive exams or special days.

A task describes a presence need. It carries no subject, no teaching syllabus and no mandatory student group, unlike a lesson. The grid is often much finer than a teaching timetable: in student supervision, the quarter hour is common. The need can change sharply during the day: 0 people in the playground during lessons, 3 people at noon, 1 person in the library all afternoon.

The module's screens

Grid

The grid defines the time frame of the service: working days, time slots — often finer than a lesson grid — and the periods of the year when it applies. It serves as a working template before naming people. Sites and their distances are declared in the same place: they are used to flag incompatible assignments between distant locations.

See [Building a service grid](#).

Assignments

The Assignments screen defines the posts or tasks to cover: label, required and ideal number of people, priority, site, authorized staff and compatibilities between tasks — for example two adjacent corridors that the same person can cover on a quiet slot when understaffed. Needs can vary slot by slot.

See [Defining the tasks to cover](#).

Planner

The planner assigns staff members to the actual tasks of a week or a date range. Assignment is done by selecting, moving, duplicating and correcting manually. Absences, unavailability, conflicts and assignments of unauthorized

people appear as alerts.

See [Assigning staff](#).

Roster

The duty roster gives each person their list of tasks or their week as a grid. It can be printed, exported or shared depending on the rights and the distribution mode chosen by the institution. It shows the totals useful for reading the week and keeps the absence or unavailability signals.

See [Rosters](#).

What the module checks

- people's availability;
- declared absences;
- assigned workload;
- authorized staff on each task;
- conflicts with other tasks or presences;
- consistency of needs per site and per slot, taking the distances between sites into account.

Staff planning is **manual**: the grid, the alerts and the correction tools guide each assignment, which you keep under control. Omniscol does not place staff automatically.

Operational benefit

The gain comes mostly from centralization:

- a shared needs grid instead of an isolated spreadsheet;
- assignments that can be changed week by week;
- duplication of stable weeks;
- absences visible in the staff schedule;
- printable or shareable duty rosters;
- fewer back-and-forths between the supervision team, the people assigned and the administration.

For a school of about a thousand students, customer feedback indicates around 20 hours of administrative work saved per week.

"We chose Omniscol to digitalize the schedule management of our student supervision department (the education assistants' timetable). The solution proved intuitive, comprehensive and perfectly suited to our needs. Everyone can clearly see where they are assigned."

— Jeanne Weeber, Deputy Head, Collège Sévigné (founded 1880), Paris

Other uses

- **Exam supervision**: midterm sessions, school-leaving exams, competitive exams, several days, several rooms, supervisor needs per exam paper.
- **Study halls and supervised study**: supervision posts that vary with the number of students expected.
- **Montessori and active-pedagogy supervision**: activities or workshops to cover without imposing the structure of a standard lesson.
- **Activity-leader and task schedules**: leisure center, after-school care, boarding school, schedules organized by mission.
- **Outing supervision**: school trip, educational outing, school visit.
- **One-off presences**: parent reception, open house day, class council.

Difference from a lesson

A lesson carries a subject, a group of students and a qualified teacher. A Staffing task carries a **presence need**: someone is needed on this post, with the right availability and, where necessary, the authorization to cover it.

This difference explains why the module works with needs grids, people assignments and duty rosters, instead of starting from a subject / class / teacher distribution.

Module scope

An account with Staffing includes:

- the Staffing module: grids, tasks, planner, duty rosters;
- the Absence management module for assignable staff;
- the administration screens needed for accounts and roles;
- cross-cutting search, according to the account's rights.

The Staff role

The **Staff** role is designed for student supervision teams (pastoral staff, education assistants, monitors, attendants). Depending on the account's rights, it gives access to Staffing, to the personal schedule and to staff absences, without opening up the school's entire global configuration.

The same user can combine several roles: for example a teacher involved in an active pedagogy can be both a teacher and a staff member scheduled by task.

See also

[Building a service grid](#)[Defining the tasks to cover](#)[Assigning staff](#)[Rosters](#)[Overview of the Absences module](#)[Staffing](#)[Study halls and supervised study \(primary and secondary\)](#)

7.2 Building a service grid

Source: [help/en/staffing/building-grids.md](#) · id: `staffing.building-grids` · Audience: `admin/staff` · Options: `staffing` · SKU: `omniscol/staffing-only` · Updated: 2026-06-26

👑 OPTION: STAFFING

A **service grid** defines the time frame of the service: the working days, the time slots and the periods of the year when it applies. This step happens **before** naming people — you set the frame, the tasks to cover are then described in the Assignments screen, and the filling is done in the planner.


Combined with the needs declared on each task, the grid answers questions like "how many people in the playground at 10 a.m.?" or "which slots must the student supervision team cover this week?".

Creating a grid

The `Grid` tab of the Staffing module shows the existing grids and how they are spread across the year. For each grid, you fill in:

- the **name** — short and recognizable, for example `Standard week`, `Term 1 exams`, `Holidays`;
- the **Working days** — for example Monday to Friday for the standard week, or a specific selection for an exam week;
- the **Time grid** — the grid's time slots, which you define freely. In student supervision, the quarter hour is common, and these slots can be finer than the teaching lesson grid.

The  **Grid allocation** button then assigns each grid to the weeks or date ranges of the year it covers.

On first opening, an empty grid is offered. To create an additional grid, duplicate an existing grid with  **Duplicate**, rename it, then adapt it.

Defining the tasks and their needs

The tasks themselves — label, number of people, site, authorized staff — are defined in the Assignments screen: see [Defining the tasks to cover](#). For each task, the needs editor then states, slot by slot on the grid, how many people are expected (0 = no need).

The need can vary sharply with the time of day: 0 people in the playground during lessons, 3 people at noon for the cafeteria, 1 person in the library all afternoon.

Sites and distances

If your institution has several sites, declare them at the bottom of the Grid screen with their distances. A task located on a distant site triggers an alert if the same person is expected elsewhere without enough travel time.

Compatibilities when understaffed

When the available headcount is shorter than ideal (a frequent case: cascading absences, exam periods), you can declare **compatibilities** between nearby tasks: the same person can then cover two tasks in parallel.

Typical example: `corridor 2 + corridor 3` can be merged — the same person can supervise both. You declare it on the task, in the Assignments screen, to guide assignment trade-offs.

Do not declare a compatibility to permanently hide understaffing. A compatibility must match a situation the institution genuinely accepts on the ground.

Grid templates

If you manage the same schedule structure every week, you can **duplicate an existing grid** to start from a base. Recurring cases:

- **Standard week** — the reference grid (start of school year to end of year).
- **Holiday week** — fewer tasks, reduced team.
- **Exam week** — lecture-hall supervision added on top of the regular tasks.
- **Special day** — open house, outing, internal event, council or special reception.

Checking consistency before assigning

Before moving on to the assignment step, use the grid as a business checklist:

- do the tasks cover every relevant period?
- are the needs realistic given the available team?
- do multi-site posts account for travel times?
- do the authorized-staff restrictions match the people actually accredited?
- do the compatibilities between tasks remain acceptable even when understaffed?

Once the grid is validated, move on to the **assignment** step. See [Assigning staff](#).

☰ How-to

Building a service grid

1. **A service grid** describes the frame of the service across the year: working days, time slots, sites.
2. **Create a grid.** Give it a short name (`standard week`). Select the working days, then define the time slots — typically in quarter-hour steps for education assistants. You can also duplicate an existing grid to start from a base.
3. **Spread the grid across the year** with Grid allocation: each grid covers the weeks or date ranges when it applies (standard week, exams, holidays).
4. **Declare the sites** if your institution has several. Distances between sites make it possible to flag incompatible assignments.
5. **Save the grid**, then describe the tasks and their needs slot by slot in the Assignments screen. See [Defining the tasks to cover](#).
6. **Next step: assign the staff** in the planner. See [Assigning staff](#).

🔗 See also

[Overview of the Staffing module](#)

[Defining the tasks to cover](#)

[Assigning staff](#)

7.3 Defining the tasks to cover

Source: [help/en/staffing/assignments.md](#) · id: `staffing.assignments` · Audience: `admin/staff` · Options: `staffing` · SKU: `omniscol/staffing-only` · Updated: 2026-06-13

👑 OPTION: STAFFING

The Assignments screen is used to define the tasks or posts that the Staffing module will have to cover. It is a configuration step: you describe the needs before assigning people in the [Planner](#).

Distribution to staff comes next, via the [Roster](#) (see [Rosters](#)).

What a task describes

A task can carry:

- a clear label: `Reception`, `Playground`, `Cafeteria`, `Exam supervision lecture hall A`;
- a minimum required number of people and an ideal number;
- a priority level;
- a site or a place;
- the specific list of staff members authorized to cover it (empty = all staff);
- compatibility rules with other tasks;
- detailed needs, slot by slot, on the service grid.

The task must be precise enough to be understood in the planner and in the duty roster handed to staff.

Frequent examples:

- `Corridor 2`;
- `North gate reception`;
- `Playground`;
- `Cafeteria`;
- `Study hall room B12`;
- `Exam supervision lecture hall A`;

- PE outing Charcot stadium.

Needs and authorized staff

Needs state how many people must cover a task, slot by slot on the service grid (0 = no need), on top of the task's required minimum and ideal headcount. The authorized staff list restricts who is eligible: reserve it for posts that require a specific accreditation or skill — a PE outing, specific supervision, special reception duties, access to a site.

This information keeps assignment from becoming a mere list of names: Omniscol keeps the link between the post, the need and the people authorized to cover it.

For a sensitive task, restrict the authorized staff rather than relying on verbal instructions: the planner then flags any assignment of an unauthorized person.

Compatibilities between tasks

Some nearby tasks can be covered by the same person if the institution allows it. Example: two adjacent corridors during a quiet slot.

Declare these compatibilities only when they match a real operating rule. An overly broad compatibility makes the schedule harder to read and can hide understaffing. Case to avoid: merging distant posts only because someone is missing.

Import and structure reuse

When many tasks already exist, the screen saves time with the import and structure-reuse tools available in the module. After importing, check the labels, the needs and the authorized staff before moving on to the planner.

A clean import should produce labels directly readable by the people assigned. If you import internal codes, rename them before distribution.

Why tasks and the planner are separate

The separation avoids mixing two decisions:

- **defining the need:** which posts exist, where, with which rules;
- **assigning people:** who covers these posts over a given week or period.

When the structure changes little, this separation lets you reuse the same tasks and change only the people or the exceptions.

How-to

1. Open Staffing > Assignments.
2. Create or import the tasks to cover.
3. Check the labels visible to the team.
4. Fill in the needs, the sites and the authorized staff.
5. Declare only the compatibilities that are actually accepted.
6. Save, then open the planner to name the people.

See also

[Building a service grid](#)

[Assigning staff](#)

[Rosters](#)

[Overview of the Staffing module](#)

[Overview of the Absences module](#)

7.4 Assigning staff

Source: <help/en/staffing/planner.md> · id: [staffing.planner](#) · Audience: [admin/staff](#) · Options: [staffing](#) · SKU: [omniscol/staffing-only](#) · Updated: 2026-06-13

👑 OPTION: STAFFING

The `Planner` is used to name the people who cover the tasks defined in the service grid. It is the operational working screen: choose a week, fill the cells, fix the alerts, then save.

Manual assignment

For each slot and each task, you place the available staff members in the corresponding cell. Omniscol flags the situations to check:

- absent person;
- declared unavailability or constraint;
- person already assigned elsewhere;
- person not authorized for the task;
- weekly workload to monitor.

The planner is designed for the real cases of student supervision: student education assistants whose availability changes, exam periods, absences declared in the morning, posts to cover again quickly and teams to spread fairly.

Planning is **manual**: the alerts, duplication and correction tools guide the construction of the schedule. Omniscol does not place staff automatically.

Reading the alerts

Colors and tooltips state why an assignment deserves a check: absence, unavailability, another task on the same slot, unauthorized person or exceeded service hours. The weekly summary lets you get back to the cell concerned.

Alerts are graded by severity: a red conflict requires an immediate correction; a warning flags a situation to arbitrate according to the institution's rules.

Correction tools

The planner notably lets you:

- navigate by week and by day;
- display one day or the whole week as needed;
- duplicate one week to another when the structure repeats, filtering the copied days or tasks if necessary;
- turn on an eraser mode to clear a cell, a row or a column;
- undo a recent correction;
- save the changes.

People locked on a cell stay fixed during subsequent manual corrections.

Duplicating without starting from scratch

When the structure is stable, duplicate the previous week then correct only the exceptions: an absence, a one-off unavailability, an exam change, an outing or a reinforced need on a post. This is the usual method to avoid rebuilding the whole supervision schedule by hand every week.

For atypical weeks, use a suitable grid: holidays, exams, open house or a period with a reduced team.

Absences

Staff absences declared in the Absence management module are taken into account in the assignment. An absent person can appear as an alert on tasks already placed, which helps identify the posts to cover again.

Absences make the problem visible in the right place: the administrator or the coordinator then reassigns the task with full knowledge of the situation.

Workload

The planner computes service information per person and per day, including the periods covered and the durations. Use these indications to prevent one person from absorbing too many slots, or the hard tasks from always being concentrated on the same team.

Sharing

The planner can be shared according to the sharing options available on the account. To hand each person their individual schedule, use the [duty roster](#) instead, more readable and designed for printing or distribution.

☰ How-to

1. Open Staffing > Planner.
2. Choose the week or the range to work on.
3. Check that the right service grid is active.
4. Assign people to the cells.
5. Handle the availability, absence or unauthorized-person alerts.
6. Duplicate the week if the structure repeats.
7. Save, then open the duty roster for distribution.

📌 See also

[Building a service grid](#)

[Defining the tasks to cover](#)

[Rosters](#)

[Overview of the Absences module](#)

7.5 Create and share rosters

Source: [help/en/staffing/roster.md](#) · id: [staffing.roster](#) · Audience: [admin/staff](#) · Options: [staffing](#) · SKU: [omniscol/staffing-only](#) · Updated: 2026-05-18

👑 OPTION: STAFFING

The [Roster](#) presents the tasks assigned to one person or to all staff over the selected period. It is used to distribute the schedule once the assignment work is done.

This screen is designed to give a simple answer to the question: "where do I need to be, at what time, and for which task?"

Two views

- **List view:** tasks sorted by day and in chronological order, with the information useful for individual reading.
- **Grid view:** the week as a table, with time slots as rows and days as columns.

Absences, unavailabilities and useful comments stay visible to avoid distributing an inconsistent roster. The list view displays the totals per day and per week.

Printing and export

The roster can be printed. The grid view also offers an export as a readable table, with the option to produce a PDF from the export window.

Depending on how the school is organized, the roster can be:

- printed for posting;
- exported as a PDF;
- sent by email outside Omniscol;
- viewed directly by signed-in staff;
- shared via a link when that option is used on the account.

Printing remains frequent for a student supervision team; the same screen is also used to prepare an individual PDF, to send a view to one person, or to let staff check their schedule with their own account.

Sharing and calendar

Sharing can include a web link and an iCal subscription depending on the available options. As with other share links, set an expiry date suited to the period being sent.

For individual communication, always check that the selected person is the right one before printing, exporting or sharing.

All staff or a single person

You can view one specific person or all staff. The individual view is suited to distribution to a named person; the global view is more for internal checking before distribution.

How-to

1. Open Staffing > Roster.
2. Choose one person or all staff.
3. Select the period.
4. Choose List view or Grid view.
5. Check absences and any alerts.
6. Print, export or share according to the intended distribution method.

See also

[Assigning staff](#)

[Defining the tasks to cover](#)

[Public share links](#)

8. Administration

8.1 Users and roles

Source: [help/en/admin/users-and-roles.md](https://help.en/admin/users-and-roles.md) · id: `admin.users-and-roles` · Audience: `admin` · Updated: 2026-05-15

User administration brings together the accounts that can log in to Omniscol. The main views are administrators, teachers, students and, if the Staffing module is active, the staff.

Main roles

Technical role	Interface label	Usage
<code>admin</code>	Administrator	Administration of the school's account.
<code>teacher</code>	Teacher	Teacher portal, availability, personal timetable, own absences.
<code>student</code>	Student	Student portal, personal timetable, own absences.
<code>staff</code>	Staff	Staff members scheduled by Staffing, if that module is active.

A user can hold several roles at once. The menu then displays the modules corresponding to all of their rights.

The technical role `share` exists for certain signed links. It is not a standard user account: it opens a specific scope, often read-only, from a sharing URL.

Available screens

- Administrators: administrator accounts.
- Staff: the school's staff (if the module is active).
- Teachers: teachers and instructors.
- Students: students, pupils or learners depending on the school's vocabulary.
- All users: consolidated view of all accounts.

Common fields







The user form manages, among other things:

- first name, last name and gender;
- date of birth;
- email and phone;
- external identifier / registration number;
- login, generated according to the account settings;
- roles;
- comment.

Additional fields appear depending on the role: service hours, subjects and external status for teachers; placements in classes and groups for students; service settings for the staff.

List actions

Depending on the selected role, the screen offers:

-  `Import data` for bulk import or modification;
-  `Invite user` to send invitations;
-  `Change password` to set passwords;
-  `Send email` to send a message;
-  `Assign to a class` and  `Groups` for students.

The visible actions depend on the available APIs, the account type and the rights of the logged-in user.

OPTION: CUSTOM ROLES

Custom roles


The **Custom roles** option adds restricted administration roles for administrator accounts. It lets you exclude certain modules or certain operations for an account that only needs to administer part of the scope. See [Custom roles](#).

Sharing links

Public links and signed shares do not necessarily involve creating a user account. Timetable web links are read-only; some targeted links can nevertheless allow a limited action, for example entering a teacher's availability until an expiration date.

☰ How-to

Inviting a group of users

1. Open the screen matching the role: teachers, students, staff or all users.
2. Select the rows concerned.
3. Click  Invite user.
4. Check that each user has a login and an email.
5. Send the invitations. Users then set their password via the link received.

🔍 See also

[Architecture and roles](#)

[Inviting and activating your users](#)

[Custom roles](#)

[Manage staff members](#)

8.2 Manage administrators




Source: [help/en/admin/admins.md](#) · id: [admin.admins](#) · Audience: [admin](#) · Updated: 2026-05-14

The **Administrators** screen is the filtered view of the users who hold the administrator role (see the role table in [Users and roles](#)). This role gives access to the school's configuration operations: users, timetables, absences, settings, imports, exports and publication.

Create an administrator

Use  Add an administrator from the list. The form is the same one used for users:

- first name and last name;
- gender;
- email and phone, if relevant;
- external identifier / personnel number, if the institution uses one;
- login generated according to the account settings;
- administrator role;
- comment.

The created account already has a login identifier. To activate the person's access, open their record in the user list: you can **set a password** () , **send them an email** () and **activate or deactivate** the account () . Signing in through an identity provider (SSO) is another option, see [OIDC / SSO](#).

Combine several roles

The same account can hold several roles. A coordinator can be both a teacher and an administrator: they keep their teacher portal and also access the administration screens.

Custom roles can limit an administrator to a narrower scope. Without a custom role, the administrator role remains global.


Remove administrator access

To remove administration rights, edit the account and remove the administrator role. Avoid creating a second account for the same person unless a real constraint requires it: combining roles on a single user is designed for this case.

Always keep at least one working administrator account before changing the rights of the other administrators.

How-to

Create an administrator

1. Open Administrators.
2. Click  Add an administrator.
3. Fill in the identity, the email address if you plan to send the person a message, then check that the administrator role is present.
4. Save. You must enter **your** administrator password to confirm any creation or modification of an administrator account, as a security measure.
5. If needed, select the account in the list to **set its password**, **send an email** or **activate / deactivate** access, depending on your deployment approach.

See also

[Users and roles](#)[Custom roles](#)[OIDC / SSO](#)

8.3 Custom roles for administration

Source: [help/en/admin/customroles.md](#) · id: [admin.customroles](#) · Audience: [admin](#) · Options: [customroles](#) · Updated: 2026-06-13

OPTION: CUSTOM ROLES

Custom roles restrict the rights of an administrator account. The principle is subtractive: you start from an administration scope, then remove the modules or operations the person must not be able to use.

This is the option to enable in order to delegate part of the administration without granting all global rights.

Why not grant all administration rights?

The standard administrator role grants all rights on the school account. For many coordination functions, this is too much:

- an absence coordinator needs to view and manage absences, but not edit the timetables or the global settings;
- a timetable manager needs to build the timetables without necessarily having access to import / export;
- an HR department may need to edit teacher records without touching the courses;
- a facilities team may need to view rooms, display panels or resources without managing users.

What the form offers

The **Custom roles** screen offers:

- a **role name**;
- permissions per module;
- permissions per operation, depending on what each module exposes.

Permissions apply to modules and their operations, not to the data itself: a custom role does not restrict access to a specific class, site, subject or period. It delimits what the person can do, not the data they can act on.

Assign a custom role

Once created, the role appears in the role selectors of the user records. You assign it to an administrator account like any other role.

A user can combine several standard and custom roles. They then obtain the union of the remaining permissions.

Typical use cases

- **Student affairs / supervisors** — rights on absences, Staffing and timetable viewing, without access to the configuration.
- **Academic coordinators** — rights on the relevant timetable screens, without global settings.
- **Facilities team** — access to rooms, display panels or resources depending on the authorized operations.
- **Administrative assistants** — management of students or teachers without access to sensitive settings.

Auditability

Actions performed with a custom role appear in the logs when the logs option is active and the route in question is logged. See [Activity log \(logs\)](#).

☰ How-to

Create a “Coordinateur absences” role

1. Open **Custom roles** in the administration module.
2. Create a role named `Coordinateur absences`.
3. Allow the necessary actions on the Absence management module.
4. Keep read access to the timetables if the coordinator needs to put an absence in context, but remove edit access.
5. Exclude the modules that must not be accessible, in particular Timetable management, global settings or import / export if this is outside their scope.
6. Save, then assign the role to the administrator accounts concerned.
7. If activity logs are active, review the actions from [Activity log \(logs\)](#).

📌 See also

[Users and roles](#)

[Manage administrators](#)

[Visibility restrictions](#)

[Activity log \(logs\)](#)

8.4 Managing teachers

Source: [help/en/admin/teachers.md](#) · id: `admin.teachers` · Audience: `admin` · Updated: 2026-05-15

The **Teachers** screen lists the users who hold the teacher role. Depending on the school's context, they are also called instructors, lecturers or trainers.

This screen is used to create the accounts, manage their information, their subjects and their availability, and view their timetable.

Record fields

A teacher record contains the common fields of a user and fields specific to teachers:

- first name, last name and gender;
- date of birth;
- email and phone;
- external identifier / registration number;
- login;
- service hours;
- subjects taught;
- external status for adjuncts or external instructors;
- availability and time constraints;
- comment.

The **Timetable** tab brings together the teacher's complete timetable, across all school years, with their consolidated availability in the background. It is a read-only view: teaching assignments are made in the timetables.

From the record, the **Sharing** button opens the sharing screen, which provides, among other things, an iCal subscription specific to the teacher, valid across all school years.

Subjects taught

The subjects attached to a teacher are used to filter or prepare assignments. They are chosen from the repository of official or custom subjects.

Availability

The **Availability** tab lets the administrator enter or correct a teacher's availability. Depending on the school's settings, entry is weekly, calendar-based, mixed or disabled.

The teacher can also enter their own availability from their portal if the school allows it.

Bulk actions

By selecting teachers in the main list, you can also invite them, set a password for them or send them a message, depending on the features enabled for the account. You can filter them beforehand.

The **Import data** button opens a spreadsheet-like grid: you add, correct or delete several teachers at once, and export the list. Two of its columns are read-only, intended for export only:

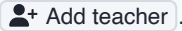
- **Availability**: rendered as text, exactly as it was recorded;
- **Classes**: those where the teacher has lessons, consolidated from the active timetables of the current year.

PREMIUM

Teachers can be synchronized from an external system: **Synchronize** opens the matching with that system. See [Synchronization with external systems](#).

☰ How-to

Creating a teacher

1. Open Teachers.
2. Click .
3. Fill in the identity, email, login or external identifier according to your internal policy.
4. Add the subjects taught.
5. Fill in the service hours if they are used by your school.
6. Save, then invite the teacher if you want them to log in.

🔗 See also

[Users and roles](#)[Managing students](#)[Managing subjects](#)[Teacher availability](#)[External faculty \(adjunct instructors\)](#)

8.5 Managing students

Source: <help/en/admin/students.md> · id: `admin.students` · Audience: `admin` · Plan: `standard` · Updated: 2026-05-15

The **Students** screen lists the users who hold the student role. Depending on the school's context, they are also called pupils or learners.

This screen is used to create the accounts, track their status and manage their placements in classes and groups, school year by school year.

Record fields

A student record contains the common fields of a user:

- first name, last name and gender;
- date of birth;
- email and phone;
- external identifier / registration number;
- login;
- comment.

Legal guardians, supporting documents and parent notifications are not fields of this record.



Placements in classes and groups

Teaching assignments are managed in the student's placement tab. Among other things, a placement indicates:

- the main class;
- the school year concerned;
- the start and end dates if the placement does not cover the whole year;
- the groups attended, week by week if necessary;
- any overlaps to correct.

A student can have several placements over the course of a school year, for example after a class change, a partial curriculum or a partial grade repetition.

Bulk actions

By selecting students in the main list, you can assign them to a class with  or to groups with  (if they belong to the same class), invite them, set a password for them, send them a message,

deactivate them when they leave the school or delete them permanently. You can filter them beforehand, in particular by assigned class.

The **Import data** button opens a spreadsheet-like grid: you add or correct several students at once, and export the list (CSV, Excel, PDF). For each school year, two of its columns are read-only, intended for export only:

- **Classes:** the class or classes where the student is placed for the current year and the following ones;
- **Groups:** the groups attended for the current year and the following ones.

Grade repetitions and off-curriculum paths

A partial grade repetition — a student who takes lessons spread across two levels — is handled by placing the student in both classes concerned over the same period: their timetable then combines the lessons of both classes. Omniscol does not check for time overlaps between them; it is up to you to verify them.

How-to

Creating and placing a student

1. Open Students.
2. Create the student or import them in bulk.
3. Open their record, then the placement tab of the school year concerned.
4. Add a class, check the dates, then select the groups attended if necessary.
5. Save the placements.
6. Invite the student if you want them to log in to the portal.

See also

[Users and roles](#)

[Class, group, subgroup](#)

[Import and export](#)

8.6 Manage staff members

Source: <help/en/admin/staff.md> · id: *admin.staff* · Audience: *admin* · Options: *staffing* · SKU: *omniscol/staffing-only* · Updated: 2026-06-25

OPTION: STAFFING


The **Staff** screen lists the users who hold the **Staff** role. It only appears on accounts equipped with the **Staffing** module: this is where you declare the people (supervisors, monitors, education assistants...) that you will later schedule in [Staffing](#).

This tab is used to create the accounts and manage their information and their service hours. **Assignment to tasks** (duty periods, supervision, exams...) happens in the Staffing module — see [Assignments](#).

Record fields

A staff member record contains the common fields of a user and a few staff-specific fields:


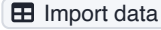
- first name, last name and gender;
- date of birth;
- email and phone;
- external identifier / staff number;
- login;
- **service hours** (reference volume for tracking);
- roles and comment.

A **detailed record** is accessible from the list ( Details) to view and complete this information member by member.

Staff member and Staffing


The separation is deliberate: **here** you manage the *people* (the directory of members), **in Staffing** you manage their *schedule* — grids, tasks and assignments. A member created on this screen becomes selectable as an authorized supervisor in the [assignments](#).

List actions

From the list,  adds a member, and  opens **bulk** creation or correction. The selection actions also let you invite members, set passwords, activate accounts or delete them, depending on the available APIs.

How-to

Create a staff member

1. Open Staff.
2. Click .
3. Fill in the identity, the email, the login or the external identifier according to your internal rule.
4. Fill in the **service hours** if your school tracks them.
5. Save, then invite the member if you want them to sign in to their portal.
6. Then schedule their tasks in [Staffing](#).

See also

[Users and roles](#)[Managing teachers](#)[Staffing — overview](#)[Assignments](#)[Staffing](#)

8.7 Managing subjects (official and custom)

Source: help/en/admin/subjects.md · id: `admin.subjects` · Audience: `admin` · Updated: 2026-05-14

The **Subjects** screen manages the school's custom subjects and, if the shared repository is active, lets you search the official subjects of the configured country.

Common and custom subjects


- **Common subjects** come from the Omniscol repository for the school's country. They can be searched and added when this repository is not disabled.
- **Custom subjects** belong to the school. They cover local titles, modules, electives or units that are not in the shared repository.

Fields of a custom subject

Field	Required?	Purpose
code	Yes	Short code for the subject, paired with the name (which can be long, especially in higher education). Often an official reference — for example the subject codes of ministries of education (France, Italy) — usually in capitals and/or digits.
name	Yes	Full label of the subject, displayed in lesson selectors, on the timetable and in exports.
short name	No	Abbreviated label for display on the timetable, shorter than the full name.
parent subject	No	Attaches the subject to another subject to build a subject hierarchy.
family	No	Groups subjects together. The selector offers the default families of the country's repository and the account's own families.
color	No	Color for visual identification. At creation, you choose from a palette of shades that are readable on screen; otherwise, a color is computed automatically from the name.

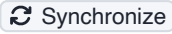
The code and the name are both requested at creation. Avoid changing the code once it is used as a reference in your imports, exports or exchanges with an external system.

Bulk creation and modification

The  button opens **bulk** creation and modification of custom subjects: enter or replace several subjects in a single operation, in a spreadsheet-like grid. Deleting a row from the grid and confirming removes the corresponding subject from the catalog: the grid replaces the whole set of custom subjects with its content. In this grid, the color field accepts any hexadecimal value, without the palette of shades offered at creation — at the risk of a less readable display.

A custom subject can be deleted even if it was used in past or published timetables. You can therefore clean up the subject catalog from one year to the next without touching those timetables: each class keeps a local copy of the subject's label (name, short name, code), and the timetable keeps the color that was assigned to it. The subject therefore continues to be displayed after its deletion from the catalog. See [Complete data model](#) for the local-copy principle.


PREMIUM

If synchronization with an external system is configured,  opens the matching with that system. See [Synchronization with external systems](#).

Lesson types

The same screen also displays the simple list of lesson types. See [Types of course](#).

PREMIUM

Lesson types can also be synchronized from an external system:  opens the matching. See [Synchronization with external systems](#).

☰ How-to

Adding a subject

1. Open Subjects.
2. For a common subject, use the repository search if it is available.
3. For a subject specific to the school, click **Create** with `+ Create`.
4. Fill in the code and the name, both required, then complete the short name, the family, the parent subject or the color if necessary.
5. Save. The subject becomes available in lesson selectors and teacher records.

🔗 See also

[Types of course](#)[Managing teachers](#)[Subject](#)

8.8 Types of course

Source: <help/en/admin/lesson-types.md> · id: `admin.lesson-types` · Audience: `admin` · Updated: 2026-05-14

In the **Administration** module, a **type of course** is a short label attached to lessons to distinguish formats such as `Lecture`, `Tutorial`, `Lab`, `Exam` or `Project`. The types of course form a list of plain text labels, with no associated color or icon.

Where to manage them

The list of types of course is managed from the **Subjects** screen and from the **Settings** screen, in the **Type of course** section. The buttons used are:

- `+ Create` to add a new type;
- `Save` to save the list.

The display order can be rearranged by drag and drop.

The “Mandatory course type” setting

The `Mandatory course type` setting is configured on the **Settings** screen. When it is enabled, a course entry — that is, a subject assigned to a class in a timetable — must include a type. This is useful when you export lessons to an ERP that requires this information.

Good practices

- Keep the list short and stable.
- Avoid renaming a type used in production without checking the imports and exports that rely on it.
- Use types to distinguish teaching formats, not to replace subjects or course delivery modes (on-site/remote).

☰ How-to

Add a type

1. Open the **Subjects** screen.
2. In the **Type of course** section, click `+ Create`.
3. Enter the label of the type.
4. Reorder the list if necessary.
5. Save with `Save`.

See also

[Courses, lessons, course types](#)

[Managing subjects](#)

8.9 School year and holidays

Source: <help/en/admin/school-year.md> · id: `admin.school-year` · Audience: `admin` · Plan: `standard` · Updated: 2026-06-26

In the **Administration** module, the **School years** screen defines the periods over which timetables, holidays, absences and student placements are interpreted. It is also called the academic year.

Create a school year

A school year contains:

- a **name**;
- a **start date**;
- an **end date**.

On creation, Omniscol can offer to import the common holidays of the configured country when data is available for the chosen range. You can deselect those that do not apply to you.

Current school year

The **Current school year** selector indicates the year displayed by default in the application and the portals. Changing this value switches the users' default context. It is also the only year accessible to students.

Keep the current year set to the year actually used day to day. Preparing the next year is done by creating a new year and then working on the corresponding timetables, without setting it as the current year too early.

Holidays

Each school year contains a list of holidays:

- name;
- start date;
- end date.

You can add them one by one, edit them, delete them or manage them as a table with **Table**.

Alternate weeks

If alternate weeks are enabled in the general settings, the screen displays a timeline for adjusting the year's alternations. Saving is done with **Save**.

The label format (`A, B, ...`, `1, 2, ...`, or disabled) is set in the general settings.

Shift across holidays: if you were on week A just before the holidays and want to resume on week B afterwards, click the relevant weeks on the timeline to create a **virtual shift**. The alternation then realigns itself with the rest of the calendar.

☰ How-to

Prepare the next year

1. Open the **School years** screen.
2. Create the next year with its name, its start date and its end date.
3. Import or enter the holidays.
4. Configure the alternate weeks if your school uses them.
5. Then create or duplicate the timetables in the **Timetable management** module.
6. Only change the current school year when users should see that year by default.

🔗 See also

[School year](#)[School year](#)[Publishing \(activating\) a timetable](#)[Preparing the next school year](#)

8.10 General school settings

Source: [help/en/admin/parameters.md](#) · id: [admin.parameters](#) · Audience: [admin](#) · Updated: 2026-05-14

In the **Administration** module, the **Settings** screen contains the school's global configuration. Some settings apply immediately to the whole account and others serve as default values for future timetables.

Main settings

The screen lets you configure, among other things:

- **school name**;
- **time zone**;
- **current school year**;
- **class levels** with [+ Create](#);
- **opening hours** for accounts that use Staffing;
- **teacher availability entry mode**;
- **student visibility restriction**;
- **teacher visibility restriction**;
- **login restriction by role**;
- **middle name handling**;
- **name capitalization rule**;
- **login syntax**;
- **first day of the week**;
- **alternate weeks format**;
- **whether the course type is mandatory or not**;
- **enabling or disabling the shared subject repository**;
- **school logo**.

Backup point quotas and the log retention depth are not ordinary settings on this screen. They are enabled or adjusted by Omniscop according to the school's contract.

Levels

Levels are used to rank classes by educational progression: Grade 6, Grade 7, Bachelor, Master, year 1, year 2, etc.

They are part of the account's **general** settings. You can create, delete and reorder them from this screen. They are then used in classes, in some filters and in several diagnostic views.

Visibility and login settings

The Time span schedule restriction for students, Restrictions on teachers consulting timetables and Application login restriction settings are documented in [Visibility restrictions](#).

Course types and common subjects


The [Mandatory course type](#) setting controls whether a type is required when entering a course. The common subjects setting controls access to the country's official repository.

Logo

The uploaded logo is stored in the account and used by the interface where the context calls for it, notably on the login screens and in the application.

How-to

Set up the account

1. Open **Settings**.
2. Check the school name, the time zone and the current school year.
3. Configure the levels, the first day of the week and the alternate weeks before creating the timetables.
4. Set the visibility and login restrictions according to your school policy.
5. Save the general settings with  **Save** .

See also

[Set up the school account](#)[School year and holidays](#)[Managing subjects](#)[Visibility restrictions](#)[Advanced settings and customization](#)

8.11 Advanced settings and customization

Source: [help/en/admin/advanced-parameters.md](#) · id: [admin.advanced-parameters](#) · Audience: [admin](#) · Plan: [premium](#) · Updated: 2026-06-13

PREMIUM


PREMIUM

In the **Administration** module, the **Settings** screen brings together, on Premium accounts, **advanced** settings in addition to the general ones. This page documents these advanced functions; for the basic settings, see [General school settings](#).

Campuses

Campuses are used to organize classes by internal structure: location, faculty, department, program or any other grouping specific to the institution. This concept is **optional** and remains distinct from sites: a site is a physical place, with its own time grid, rooms and travel times, whereas a campus is simply a logical organizational unit.

Campuses become especially useful when geography does not match your logical organization well: several faculties in the same buildings, or several schools of a group sharing several sites in overlapping ways. Even when campuses and sites overlap, this level of organization can still be useful for filters and groupings.

Use  **Create** to add only the campuses that genuinely serve to organize or filter the work. They then become available in the class creation screens and some diagnostic filters.

Translation and terminology overrides

The screen also lets you **override translations** to adapt the displayed vocabulary to your institution. This area is used to customize the labels visible in the interface, without changing the functional structure of the product.

There are two distinct mechanisms:

- **Translation replacement rules**: they replace one term with another across a set of labels;
- **Translation redefinitions**: they replace the value of a specific translation directly.

Replacement rules

Replacement rules are used to harmonize vocabulary across many labels at once. Typical examples:

- replacing `Professeur` with `Intervenant` ;
- replacing `Élève` with `Étudiant` ;
- adapting certain terms to the culture of the institution.

Rules are **ordered**. Order matters: a rule placed higher applies before the following ones. The screen lets you:

- enter a source text and a target text, then confirm with **+ Add** ;
- preview the affected labels;
- reorder the rules;
- delete a rule.

The available options refine the type of replacement:

- `Aa` : case-sensitive matching;
- underlined `ab` : whole-word matching;
- `.*` : regular expression.

Use rules when you want to change a term consistently across several screens.

Rules are saved **in their own area**: adding, deleting and reordering are recorded separately from the other settings on the page.

Full key overrides

Full key overrides are used to correct or rewrite a label **case by case**.

The screen lets you:

- search for an existing translation;
- add it to the list of overrides;
- enter your own version;
- save or delete this override.

Use this mode when a general rule would be too broad, or when you want to rewrite an entire sentence rather than a single term.

Each key override is saved **line by line** with its own save or delete button.

Good practice

Rules and overrides are meant to customize the **displayed vocabulary**. They change neither the underlying business concepts, nor the internal identifiers, nor the behavior of the application.

Prefer:

- a **rule** for a recurring renaming;
- a **key override** for a local exception or a complete label.

Avoid overly broad or ambiguous replacements that would make help, searches or terminology inconsistent from one screen to the next.

☰ How-to

1. Open **Settings** in the **Administration** module.
2. Locate the advanced area that matches your need: campuses or translation overrides.
3. For campuses, create and order the labels that are useful to your organization.
4. For translations, choose between a global rule and a one-off override depending on the scope of the change.
5. Save each area with its own mechanism: rules are recorded in their block, and key overrides are confirmed line by line.

🔗 See also

[General school settings](#)[Visibility restrictions](#)[Campus](#)[Conflicts and diagnostic](#)

8.12 Visibility and login restrictions

Source: [help/en/admin/visibility-restrictions.md](#) · id: `admin.visibility-restrictions` · Audience: `admin` · Plan: `standard` · Updated: 2026-05-14

In the **Administration** module, the **Settings** screen brings together a few targeted restrictions: a viewing horizon for students, hiding teachers' timetables from one another and blocking login by role.

Student-side restriction

The `Time span schedule restriction for students` setting defines the number of future weeks, including the current week, during which the published timetables remain visible to students. You set it to a number of weeks or to **Unlimited**.

This restriction applies to the time window open to students: each one views the published timetables up to the chosen horizon.

Teacher-side restriction


The `Restrictions on teachers consulting timetables` setting hides the timetables of the other teachers. It acts on the filters and the viewing screens: a teacher sees their own timetable, without access to their colleagues'.

Login restriction

The `Application login restriction` setting blocks login for the selected profiles: **Teachers**, **Staff** or **Students**. A user who holds several roles can log in as long as at least one of their roles remains allowed.

☰ How-to

Limiting student visibility

1. Open the **Settings** screen.
2. Locate `Time span schedule restriction for students`.
3. Choose **Unlimited** or a number of weeks.
4. Confirm with  **Save**.

🔗 See also


[General settings](#)[Users and roles](#)[Student portal](#)[Teacher portal](#)

8.13 Import and export

Source: [help/en/admin/import-export.md](https://help.en/admin/import-export.md) · id: `admin.import-export` · Audience: `admin` · Updated: 2026-06-03

In the **Administration** module, the **Import/Export** screen gathers the account's data transfer operations: JSON export, aSc exchanges, snapshots, synchronization with external systems and API options enabled on the account. Depending on the country, exchanges based on standard protocols may also be available. For schools overseen by the French Ministry of Education, France-specific exchanges (STS, UnDeuxTemps, École Directe) are described in [admin.french-formats](#).

JSON export

The  **Download** button starts a full JSON export of the account. This file contains the account's data and must be treated as a sensitive backup.

The JSON export is available to all administrators. If you want to restrict access, you need the [Custom roles](#) option.

JSON import

The full JSON import uses the same format as the export. It is reserved for the Omniscol team as part of support: re-importing is done on request, not as a routine operation on the administration screen.

To restore an account, prefer snapshots when they are enabled, or contact Omniscol if a JSON re-import is necessary.

Before any destructive import, create a backup point if the feature is active.

aSc Timetables

The screen also contains an aSc Timetables section:

- import of an aSc XML file;
- aSc XML export with options.

This exchange covers a complete aSc file. To add courses from a spreadsheet, use [Mass import of courses from a spreadsheet](#) instead.


Backup points

If **backup points** are enabled, the  **Backup points** button opens their management. See [Backup points](#).

Synchronization with external systems and API options


PREMIUM

The screen also offers:

- the configuration of synchronization with external systems via  **Configure**;
- global sharing / OpenAPI;
- the API settings;
- the MCP configuration;
- OAuth2 / OIDC management.

How-to

Export the account

1. Open **Import/Export**.
2. Click  **Download**.
3. Keep the file in a secure location.
4. If the goal is a risky operation, create a snapshot in addition to the export.

See also

[Preparing your data for a mass import](#)

[Mass import of courses from a spreadsheet](#)

[Backup points](#)

[Overview](#)

[Synchronization with external systems](#)

[admin.french-formats](#)

8.14 Backup points

Source: [help/en/admin/snapshots.md](#) · id: [admin.snapshots](#) · Audience: [admin](#) · Plan: [premium](#) · Feature: [snapshots](#) · Updated: 2026-05-15

PREMIUM

PREMIUM

A **backup point** or *snapshot* is a point-in-time image of the Omniscol account. The feature is available only if the Omniscol team has enabled a backup point quota for the account, according to the school's contract. This quota is defined with Omniscol according to the school's needs (manual backups, automations, expected retention) and can be adjusted on request. It is not set directly by the administrator from this panel.

Open the backup points

In the **Administration** module, open **Import/Export** then click [Backup points](#). The panel lists the existing backup points, the quota used and, when available, a **digest** of the content: user volumes, school years, panels, events, timetables and the main business counters.

Available actions

Depending on your rights, you can:

- create a backup point with [Create](#);
- export a backup point as JSON with [Download](#);
- delete a backup point with [Delete](#);
- lock a backup point to prevent its deletion;
- set or change an expiry date;
- inspect the digest to quickly identify what the backup point contains;
- restore a backup point (in full or in part) with [Restore](#).

Backup points are kept on replicated storage. Each backup point can be exported as JSON for analysis or comparison carried out with your own tools.

Restore

Restoring requires explicit confirmation. The panel offers three modes, depending on the data families selected:

- **Overwrite data**: the selected data is replaced by the state of the backup point. This is the rollback mode;
- **Restore only deleted data**: Omniscol reinjects only the missing items when the mode is supported, for example after users were deleted by mistake;
- **Restore without overwriting**: Omniscol restores as a copy when supported, in order to compare or manually reintegrate without replacing the current state, for example when a timetable has diverged and you do not want to overwrite what has been done since the moment the backup point was created. Duplicates may therefore appear.

The restore can cover the whole account or subsets such as timetables, events, panels, absences, users, subjects, school years or configuration.

Automations

The **Automation** button lets you configure automatic creation of backup points. The pace depends on the configuration chosen for the account: daily, weekly, in the middle of the night — you decide on the policy suited to your school.

Expiry dates and locking drive retention:

- an expired backup point can be deleted automatically, unless it is locked;
- a locked backup point is protected against manual deletion and against automatic cleanup;
- when an automatic backup has to fit into a quota that is already full, Omniscol deletes the oldest unlocked automatic backup point to make room. If no backup point can be deleted (all locked), the new backup fails.

This behavior gives a backup history close to a Time Machine in an Apple environment: useful for going back in time, recovering data deleted by mistake or restoring a copy for comparison. It remains bounded by the quota defined with Omniscol, adjustable on request.

What to scope before activation

Before enabling or extending the option, check with Omniscol:

- the backup point quota and the margin desired for manual backups;
- the pace of the automatic backups you will want to configure;
- the expected retention period;
- the data families concerned by partial restores;
- the export format expected for external audits.

Each backup point can be exported as JSON.

☰ How-to

Create a backup point before a risky operation

1. Open **Import/Export**.
2. Click **Backup points**.
3. Click **Create**.
4. Check that the new backup point appears in the list.
5. Then start the planned import, reorganization or configuration change. If a serious problem occurs, you will be able to go back in time.

🔗 See also

[Import and export](#)

[Activity log \(logs\)](#)

8.15 Activity log (logs)

Source: <help/en/admin/logs.md> · id: [admin.logs](#) · Audience: [admin](#) · Feature: [logs](#) · Updated: 2026-05-15

👑 PREMIUM

In the **Administration** module, the **Activity log** screen displays the log entries available for the account when the logs option is active. The retention depth is defined with the Omniscol team according to the school's contract and can be adjusted on request. The panel is used to browse and export the traces present in the log stream.

What is logged

The log records the actions that modify the account, not every view.

- **Every change** to the data (user record, timetable, settings...) — creation, update or deletion — generates an entry with its author, the action and the timestamp.
- **Logins** — every login attempt records the identifier used.
- **Password actions** — setting, changing, deleting or resetting a password is traced: the action and its author, never the password itself.
- **Exports** — the JSON export of the account is logged.

Simple views (opening a screen, listing data) are not recorded, apart from a few sensitive operations. The log therefore answers "who did what and when", not "who viewed what".


What the screen displays

Each line can contain:

- a timestamp;
- the action performed;
- the resolved route when it is known;
- the recorded parameters and extras;
- the HTTP method;
- the associated web URL;
- the user concerned.

The exact content depends on what Omniscol has logged: the screen reflects the traces present in the account's log stream.

Search and export

When the list contains enough entries, a local search lets you filter the table. The  **Timestamp** button opens an exportable view of the lines.

Scope of the logs

- Each line gives the action performed, the route, the parameters and the user concerned; the detail covers what Omniscol has logged.
- Retention depends on the option enabled on the account and on the periodic cleanup of the logs.
- The logs are viewed and exported from this screen; Omniscol does not automatically forward them to an external monitoring tool.

To precisely reconstruct the state of an account at a given date, combine the logs with backup points.

What to check in a contract

The retention depth, the audit obligations and the expected exports are covered by the contract and the security documents provided on request. This page does not replace those documents.

See also

[Backup points](#)

[Users and roles](#)

9. Integrations

9.1 Integrations overview

Source: <help/en/integrations/overview.md> · id: [integrations.overview](#) · Audience: [admin](#) · Updated: 2026-06-13

Omniscol integrates with your IT system through several channels, depending on the need.

For examples of third-party tools and partners regularly connected to Omniscol, see also [integrations.partners](#).

At a glance

Mechanism	What it is for	Flow direction
iCal	Let teachers / students see their timetable in their personal calendar	Omniscol → calendar
Share links	Distribute a read-only timetable or open targeted data entry depending on the context	Omniscol → public web
Iframe	Embed a timetable in a page on the same domain as Omniscol; blocked on an external site (security)	Omniscol → same domain
Omniscol API	System-to-system integrations according to the account's rights and options	Depends on configuration
MCP	Connect an MCP-compatible AI agent to Omniscol	AI agent ↔ Omniscol
OAuth2 / OIDC (server)	Connect a third-party service identified by a registered client + consent + scoped token (the foundation of MCP and OneRoster)	Service ↔ Omniscol
OIDC / SSO (sign-in)	User authentication via an identity provider (Google Workspace, Microsoft Entra ID, generic OIDC)	Depends on configuration
Synchronization with external systems	ERP connector framework: Aurion, Auriga, new connectors on a project basis	Bidirectional
Display panels	Show the timetable on public screens	Omniscol → screens

Choosing the right integration

- **Distribute a timetable to a teacher / student** — an [iCal](#) subscription.
- **Give read access to a parent / an auditor** — [share link](#).
- **Synchronize learners from your ERP / IT system** — see [Synchronization with external systems](#), which lists the available connectors. If your ERP is not listed, adding a connector is scoped with Omniscol support.
- **Single sign-on** — [OIDC / SSO](#).
- **External AI assistant** on the account, when the option is enabled — [MCP](#) for compatible clients.
- **Public display on the premises** — [display panels](#).

Security best practices

- Use [OIDC / SSO](#), when the option is enabled, to authenticate real users.
- Reserve **API tokens** for system-to-system integrations, with minimal scope and periodic rotation.
- Set short **expiry dates** on share links.
- Never put an API token in a public Git repository.

See also

[iCal](#)[Omniscol API](#)[MCP](#)[OAuth2 / OIDC \(server\)](#)[OIDC / SSO \(sign-in\)](#)[Synchronization with external systems](#)[integrations.partners](#)

9.2 iCal – subscription and dynamic link

Source: [help/en/integrations/ical.md](#) · id: [integrations.ical](#) · Plan: [standard](#) · Updated: 2026-05-15

The **iCal** format (`.ics`) is the standard for sharing calendars between applications. Omniscol exposes each timetable (individual, class, classroom, teacher) as an **iCal subscription link** that the user can paste into Google Calendar, Apple Calendar, Outlook, Thunderbird or any other application that consumes standardized calendars.

Subscription link vs one-off export

Two very different uses:

- **Subscription link** (recommended) — an HTTPS URL pointing to Omniscol. The client application polls it regularly (hourly, daily depending on the client's settings) and keeps the user's calendar continuously up to date. When a lesson moves in Omniscol, the user sees it in their calendar without doing anything.
- **One-off export** — download of an `.ics` file at a given point in time. Handy for archiving, sharing by email, or feeding a system that cannot subscribe. But it is frozen: if the timetable changes, the file is not updated.

Prefer the subscription link when the end client is a modern application that can refresh a remote calendar (Google Calendar, Apple Calendar, Outlook).

Getting your subscription link

From the screens where sharing is available, the [Sharing](#) button opens a modal that can offer:

- one or more signed iCal subscription URLs,
- a "Copy URL" button,
- a QR code to scan,
- an `.ics` file download.

The URL contains a secret token: do not share it — anyone who obtains it can view the timetable covered by the link's scope until it expires or is invalidated.

Where to generate an iCal

The main entry points are:

- **At the top of the Timetable module** — the [Sharing](#) button can generate an iCal of all displayed timetables, a combined iCal that merges them into a single subscription, then the individual iCals. These links are bounded by the displayed school year.
- **On each displayed timetable** — the timetable's title bar gives access, on hover, to an atomic share for that timetable. The link covers the relevant school year.
- **On a teacher's record** — in **Administration**, the record's [Sharing](#) button generates the teacher's personal iCal, designed to remain valid over time as long as the link, the holding account and the chosen expiration allow it.
- **When previewing a calendar timetable** — in **Timetable management**, the preview's [Sharing](#) button generates the iCal of what is displayed, even if that timetable is not yet activated or published.

At each generation, the user can set an invalidation date. This date is encoded in the URL's token: it cannot be changed afterwards. To change the expiration, generate a new link.

Like the other share links, the iCal is also tied to the account that generated it: a password change, deactivation or deletion of the holding account invalidates the associated links.

Public link for a class or a classroom

The administrator can generate a subscription link for a **class** or a **classroom** (for example to share with parents or with an external service). These links are signed and carry an expiration date.

See also [Share a timetable via a public link](#) which describes the full set of sharing options (iCal, responsive web page, JSON via API).

Client-side setup

Google Calendar

1. In Google Calendar, on the left, next to **Other calendars**, click the **+ > From URL**.
2. Paste the Omniscol subscription URL.
3. The calendar appears with an indicative label and a default color — both changeable.

Google refreshes the subscription roughly every 8-24 hours. Urgent changes: expect a delay.

Apple Calendar (macOS, iOS)

1. **File > New Calendar Subscription**.
2. Paste the Omniscol URL.

The Omniscol iCal link is set up for automatic refresh every half hour.

Outlook (Microsoft 365)

1. **Add calendar > From Internet**.
2. Paste the Omniscol URL.


Outlook updates at a rate that varies across versions; for older desktop versions, expect delays.

Limitations

- Client-side changes (moving an event in Google Calendar) are not propagated back to Omniscol — the subscription is read-only.
- The user must have an active Omniscol account to generate their personal URL.
- The link's expiration is written into the generated URL: it cannot be extended afterwards; generate a new link if needed.
- Events outside the timetable are included only if the share or the iCal filter includes them.

☰ How-to

Subscribe to your timetable from Google Calendar

1. **The iCal subscription** synchronizes your Omniscol timetable with your personal calendar (Google Calendar, Apple Calendar, Outlook). Continuous automatic updates: no copy-paste to redo at every timetable change.
2. **From your Omniscol screen** (student, teacher or shared view), click  **Sharing** when the button is available. A modal opens with the URLs generated for your scope.
3. **Copy the subscription URL** (Copy button). Do not share it: anyone who has it can view the scope covered by the link until it expires or is invalidated.
4. **In Google Calendar**: left panel, **Other calendars** → + → **From URL**, paste, confirm. The calendar appears with a label and a color (both changeable).

In Apple Calendar: File → New Calendar Subscription, paste.

In Outlook M365: Add calendar → From Internet, paste.

5. **Important**: subscription = **read-only**. Changes in your personal calendar are not propagated back to Omniscol. Changes on the Omniscol side reach your calendar at the next client-side synchronization.

🔗 See also

[iCal \(calendar export\)](#)

[Share a timetable via a public link](#)

[Integrations overview](#)

9.3 Omniscol API — authentication tokens

Source: [help/en/integrations/api-tokens.md](#) · id: [integrations.api-tokens](#) · Audience: [admin](#) · Plan: [premium](#) · Updated: 2026-06-13

👑 PREMIUM

PREMIUM

The **Omniscol API** is a REST API documented in OpenAPI. It is used for system-to-system integrations: an external dashboard, a custom display, ERP or information system synchronization, or an AI agent via MCP.

API tokens managed from the interface are available on accounts that offer this integration. A derived token only grants access to the API endpoints selected when it was generated; do not assume it covers the whole API.

Key and token: two distinct objects

Omniscol distinguishes two objects that must not be confused.

A **key** is a persistent object, kept on the Omniscol side. It combines:

- a short **identifier**, randomly generated, public: it is what the list displays, and it travels in each token to designate the key to verify;
- a descriptive **label**, editable at any time;
- an optional **expiration date**, editable at any time;
- a long **random secret**, drawn on the server side, which serves as the cryptographic signing key.

The identifier, the label and the expiration date are administrative information: the identifier is only a public reference, not a secret element. The **secret**, on the other hand, is the only signing material: randomly generated when the key is created, it stays on the server side, is not editable, and **is never displayed or returned** — neither

at creation, nor in the key list. And even if it leaked, it would not be enough to forge a token: the signature combines it with an **account-specific salt** and a **server secret** which, likewise, never leave Omniscol.

A **token** is a self-contained **JWT** (JSON Web Token), signed with the secret of the key. It is what you hand to the external system. Its signed content carries the account concerned, the list of authorized endpoints, the key it derives from and its own expiration date.

In other words: **a key signs, a token is signed**. The same key can sign several tokens — all verifiable with the same secret, so all revoked together if the key disappears.

Omniscol does not store the token: it generates it, displays it once, then re-verifies it on every call by recomputing its signature from the secret of the key (HMAC SHA-256). No permission carried by the token can therefore be altered without that secret.

Creating a token

The Sharing screen is available in Administration → Import/Export on Premium accounts. Creation happens in two steps: you first create a **key**, then **generate a token** from that key.

1. **Create a key** — enter a meaningful label and, if access is temporary, an expiration date. Its label and its expiration remain editable afterwards; its signing secret does not.
2. **Generate a token** — select the key, check the authorized endpoints, optionally choose an expiration date specific to the token, then generate the JWT.

At generation, Omniscol displays the token **only once**. Copy it immediately into a secrets manager: it will not be displayed again. The token's expiration is written into its signed content: it cannot be changed afterwards. To change that date, generate a new token.

There are therefore two levels of expiration, independent of each other:

- **key expiration** — editable from the key list; once it is reached, **all** tokens derived from that key are rejected (the call fails with a 401);
- **token expiration** — set at generation, written into the JWT, and not editable afterwards.

Authorized endpoints

A token only grants access **to the endpoints checked at its generation**: never assume it covers the whole API. The authorized list is carried in the token and verified on every call; a call to an unauthorized endpoint is rejected (401).

Beyond individual endpoints, the selection list offers **per-module shortcuts**:

- a module's entry on its own authorizes **all** of its endpoints;
- the per-operation variants — **read, update, creation, deletion** — restrict the module to a single call type.

Checking "**Timetable** [read]" thus grants full read access to timetables without opening any write access, and without checking each endpoint one by one. Keep it tight: grant only the modules and the operations strictly necessary for the integration.

Using a token

Two usual ways to send the token to the API:

- **HTTP header**: `Authorization: Bearer <token>` (recommended).
- **Query string**: `?auth=<token>` (handy for debugging, but it appears in HTTP logs — avoid in production).

`curl` example, to adapt with a real endpoint taken from the OpenAPI:

```
curl -H "Authorization: Bearer $TOKEN" \  
https://your-school.omniscol.com/api/<module>/<endpoint>
```

OpenAPI documentation

The Import/Export screen displays an **OpenAPI 3.1** link that opens the specification available for the account in Swagger Editor. There you will find:

- the list of exposed API endpoints,
- the schemas of the data exchanged,
- the methods and parameters,
- the expected responses.

The specification is also served directly by your account, without authentication:

- `/api/guest/openapi.json` — specification in JSON format;
- `/api/guest/openapi.yaml` (or `/api/guest/openapi?yaml=true`) — same content in YAML format;
- `/api/guest/school_schema.json` — JSON schema of the account's data.

Revoking access

Revocation happens at the level of the **key**, not of the individual token.

The screen lists the **keys** that have been used to generate tokens. **Deleting a key** erases its signing secret on the Omniscol side: from then on, no signature derived from that secret can be verified any more, and any call presenting a token issued from that key fails with a **401**. It is the absence of the secret that invalidates the tokens, not a revocation list.

For the IT department, two consequences:

- **There is no token-by-token revocation.** Omniscol does not keep the issued JWTs and cannot disable one in isolation: a token already generated remains valid until its own expiration, or until its key is deleted or expired.
- **Changing a key's expiration** acts immediately on all of its tokens: moving that date earlier cuts off access for every token issued from the key.

Best practice: delete without delay any key you suspect has leaked, then recreate a replacement key with a meaningful label.

One key per target and per use

Since revocation is per key, **dedicate a key to each integration** (one external software = one key). A key's secret only signs its own tokens: deleting that key only invalidates the access of the integration concerned, without touching the others.

Conversely, a single key shared between several systems makes any revocation all-or-nothing: deleting the compromised key cuts off at once all the systems that were using it.

Which integrations it serves


The API is typically used for:

- custom **signage systems** (beyond the native Omniscol display panels; see [Panel customization](#)),
- **external dashboards** consolidating Omniscol and other sources,
- **connectors or synchronizations** with an ERP or a business information system,
- **MCP-compatible AI agents** consuming Omniscol through the MCP server; see [MCP — connect an external AI agent](#).

For delegated access for an **identified third-party service** (scoped token, user consent, revocable by disabling the client), prefer Omniscol's OAuth2 server: see [OAuth2 / OIDC \(provider\)](#).

☰ How-to

Generate an API token

1. An **authentication token** allows an external system (dashboard, signage, AI agent via MCP) to query the API endpoints you have selected.
2. **Go to Administration** → **Import/Export**, then open the sharing screen with  **Sharing**. There you see the existing keys, their labels and their optional expiration dates.
3. **Create a key** if needed. Enter a meaningful label (`Finance dashboard`, `Main hall signage`, `Claude desktop AI agent`) and an expiration if the integration is temporary. You will be able to change this key expiration later.
4. **Select the key**, then check the API endpoints the token must be able to call. Keep the list as short as possible. Also choose the token's expiration if access must be bounded.
5. **Generate the token**, then immediately copy the displayed JWT. It will not be displayed again and its expiration cannot be changed. Then use it in the HTTP header `Authorization: Bearer <token>`.
6. **To revoke access**, delete the corresponding key. Tokens derived from that key become invalid.

🔗 See also

[Integrations overview](#)

[MCP — connect an external AI agent](#)

[OAuth2 / OIDC \(provider\)](#)

[Panel customization](#)

9.4 OAuth2 / OIDC — connect a service to Omniscol

Source: <help/en/integrations/oauth-server.md> · id: `integrations.oauth-server` · Audience: `admin` · Plan: `premium` · Updated: 2026-06-29

👑 PREMIUM

PREMIUM

This page is intended for the **IT department**. It describes Omniscol in its role of **OAuth2 / OpenID Connect authorization server**: how a third-party service registers as a **client**, how a user **consents** to give it access, and how the service receives a **token** limited to the granted **scopes**.

What Omniscol does as an OAuth2 server

An external service — an AI agent, a connector, a dashboard — becomes a **client** declared in your account; a school user **approves** its access through a consent screen; the service then receives a short-lived **access token**, whose reach is bounded by the granted **scopes** and the user's rights.

This mechanism is distinct from the **user SSO** described on [OIDC / SSO](#), where Omniscol is, conversely, a **client** of your identity provider to sign your users in. Here, Omniscol is on the **server** side: services connect to it.

Two Omniscol integrations consume this server:

- **MCP** — the standard authentication of an AI agent goes through this OAuth2 server (see [MCP — connect an external AI agent](#));
- **OneRoster** — the OneRoster producer authenticates callers with a machine-to-machine OAuth2 token issued by this same server (see [OneRoster](#)).

Any other OAuth2 / OIDC-compliant service can connect to it in the same way.

Discovery and protocol endpoints

Omniscol publishes its **discovery metadata** at the standard `.well-known` addresses, served at the root of your account's domain (for example `https://your-school.omniscol.com`). A compliant client finds all the endpoints there on its own, without manual configuration:

- `/.well-known/oauth-authorization-server` — authorization server metadata (RFC 8414);
- `/.well-known/openid-configuration` — OpenID Connect metadata (same content as the previous one);
- `/.well-known/jwks.json` — public verification keys (JWKS), which make it possible to verify the signature of `id_tokens`;
- `/.well-known/oauth-protected-resource` — protected resource metadata (RFC 9728).

This metadata advertises the protocol endpoints:

- `/oauth/authorize` — authorization request (login screen then consent screen);
- `/oauth/token` — exchange of the code for a token, and refresh;
- `/oauth/register` — **dynamic client registration** (RFC 7591);
- `/oidc/userinfo` — information about the signed-in user (OIDC);
- `/oauth/ revoke` — token revocation (RFC 7009).

These protocol endpoints are **public**: they are not reserved for Premium accounts and do not need to be opened manually. Only the **client management screen** described below falls under Premium.

Authorization flows

Omniscol supports three OAuth2 flows:

- **Authorization code with PKCE** — the default flow for a service acting **on behalf of a user**. The service redirects the user to `/oauth/authorize`; after login and consent, Omniscol returns an **authorization code** (valid for 5 minutes) that the service exchanges on `/oauth/token`. The supported PKCE method is **S256**, and the only accepted `response_type` is `code`.
- **Refresh (`refresh_token`)** — to extend a delegated access without going through consent again.
- **Client credentials** — a **machine-to-machine** flow, with no user, used in particular by **OneRoster** consumers. The client authenticates directly and receives an access token.

At the exchange, the server issues an **access token** (`Bearer`, valid for **1 hour**) and, for user flows, a **refresh token** (valid for **30 days**). When the `openid` scope is requested, a signed OIDC `id_token` is also issued; its signature can be verified via `/.well-known/jwks.json`. The `client_credentials` flow issues only an access token, with no refresh token and no `id_token`.

The access token is then presented in the HTTP header `Authorization: Bearer <token>`. On every call, Omniscol verifies its signature, checks that the client is still **active**, that the user still exists and holds the required role, and that the token's **scopes** do cover the endpoint being called — otherwise the call is refused.

Scopes and consent

The scopes you manage on a client are:

- `read:basic` — read access to timetables, dashboards and consultation screens (Home, Timetable, Dashboard, Timetable management modules);
- `read:user` — read access to the user list;
- `write:data` — write access on those same consultation modules;
- `admin` — administration access (Administration, Absence management, Timetable management modules).

The server also knows the OIDC scopes (`openid`, `email`, `profile`) and the read-only **OneRoster** scopes (`imglobal.org` prefix). The latter are **privileged**: a client **cannot** self-assign them through dynamic registration; they must be provisioned by an administrator on the client's record.

The scope actually granted is the **intersection** of what the client requests and what is registered for it: a client never obtains more than what its record lists. During the user flow, the **consent screen** (`/oauth/consent`) shows the **name** and **logo** of the requesting service along with a **readable list** of the requested scopes, with the **Accept** and **Deny** buttons. Approving issues the authorization code and sends the user back to the service; denying sends them back with an `access_denied` error.

Dynamic client registration

The `/oauth/register` endpoint implements **dynamic registration** (Dynamic Client Registration, RFC 7591): a compliant service can **declare itself** as a client, with no prior manual intervention. This is what lets an MCP agent configure itself from the server URL alone.

Dynamic registration **cannot** grant itself a privileged scope (the OneRoster scopes): those are silently discarded, and if no valid scope remains, `read:basic` is granted by default. Privileged scopes remain reserved for provisioning by an administrator.

The OAuth2 client management screen

On **Premium** accounts, you administer clients from **Administration** → **Import/Export, OAuth2** section, with the  **OAuth2** button. Access first requires the **administrator password** — an extra confirmation before the screen opens.

The screen lists the registered clients and, for each one, shows its **state** (active / inactive), its **name**, its **scopes**, its **contacts** and its **URIs** (website, logo, redirect URIs). You can:

- **Register a client** — fill in the name, an optional `software_id`, the scopes, the contacts, the website, the logo and the redirect URIs. At creation, Omniscol shows the `client_id` and the `client_secret` **only once**.
- **Edit** a client — only safe fields can be changed: name, scopes, contacts, website and logo. The redirect URIs, the `software_id` and the secret cannot be changed here.
- **Activate or deactivate** a client — a deactivated client has its tokens refused from the very next call.
- **Delete** a client — deletion is **permanent**.

The client secret

The `client_secret` is shown **only once**, at registration. Omniscol keeps only a **fingerprint** of the secret alongside, never the secret in clear text: it **cannot** be shown again or recovered later. Copy it immediately into a secrets manager.

The `client_id`, on the other hand, is **deterministic**: it derives from the account name, the client name and a fingerprint of its technical metadata. Two strictly identical registrations therefore land on the same identifier.

Renewing the secret (rotation)

Secret **rotation** exists: it issues a new secret, keeps only the new fingerprint, and returns this new secret **only once**. It is performed through the dynamic registration management endpoint (`/oauth/register/<client_id>/rotation`) and requires presenting the **registration token** handed to the client during its dynamic registration. It is therefore **not** triggered from the management screen above, which does not handle that token.

OAuth2 or API key: which one to choose

Omniscol offers two machine access mechanisms, with different trust models:

- **OAuth2** (this page) — a **registered third party** obtains, after a user's **consent**, a **short-lived** token (1 h), bounded by the granted scopes, **refreshable** and **revocable** (by deactivating the client). The `client_credentials` flow additionally covers machine-to-machine access with no user. It is the mode suited to an identified third-party service, to MCP and to OneRoster.
- **API key** (see [Omniscol API](#)) — a **self-contained token** signed by the server, which embeds a **list of authorized endpoints** and which you hand to the external system yourself: no registered third party, no consent screen, no refresh. It is a delegation, by the administrator, of their own rights to a system they control.

In short: the **API key** fits when you yourself hand access to a system you control; **OAuth2** fits when an identified third-party service must obtain a delegated, scoped and revocable access, or when the protocol requires it (MCP, OneRoster).

☰ How-to

Registering an OAuth2 client

1. **Open the OAuth2 screen.** In **Administration** → **Import/Export**, OAuth2 section, click , then enter the administrator password.
2. **Register the client.** Fill in its name, its scopes (`read:basic` , `read:user` , `write:data` , `admin`), its redirect URLs and, if useful, contacts, website and logo. OneRoster scopes are not assigned here nor through dynamic registration; they fall under administrator provisioning.
3. **Copy the `client_id` and the `client_secret`** that are shown. The secret appears **only once**: keep it in a secrets manager.
4. **On the service side**, configure the client with these credentials and the server URL; a compliant client discovers the endpoints on its own via `/.well-known/` .
5. **To cut off an access**, come back to the screen and **deactivate** or **delete** the client.

📌 See also

[MCP — connect an external AI agent](#)

[Omniscol API](#)

[OneRoster](#)

[OIDC / SSO](#)

[Integrations overview](#)

9.5 API tweaks: endpoint overrides and hooks

Source: [help/en/integrations/api-customization.md](#) · id: [integrations.api-customization](#) · Audience: [admin](#) · Plan: [premium](#) · Updated: 2026-06-27

👑 PREMIUM

PREMIUM

API tweaks is a technical screen intended for the **IT department**. It lets you adapt the behavior of the Omniscol API for your account through **three levers**:

- a **general configuration** (shared HTTP headers);
- an **endpoint override** — redirecting the web application to another URL, or disabling an endpoint;
- **hooks** — outbound calls (webhooks) that Omniscol sends to your system when an operation takes place.

Where to find it

Administration module, **Import/Export** screen, **API tweaks** section,  **API**  button.

The screen is **reserved for Premium accounts** and protected by **reinforced authentication**: Omniscol asks for the administrator's password again before opening the window. Since every action on this screen touches the technical contract of your integration, it is set up in coordination with Omniscol.

General configuration

Here you define **HTTP headers** (format `key1:value1;key2:value2`) applied to **all hooks**. It is the natural place to carry an **authentication token** to your own server (for example an `Authorization: Bearer ...` your server expects). For an **override**, authentication goes instead in the headers specific to the endpoint (see below).

Overriding an endpoint

An **override** redefines an endpoint of the Omniscol API so that the **web application** calls **another URL** in its place. Three use cases, from the most powerful to the simplest.

Serving live data from your information system

This is the most powerful use case. You redirect a **read** endpoint to an **external URL** — typically an **ETL** that interfaces with your institution's **internal information system**. The web application then fetches the data **live** from that system, instead of the local copy held by Omniscol.

In practice: when the redirect URL is an **absolute external address**, the web application calls it directly, without going through Omniscol's servers, and **consumes the response as-is** — exactly as if it came from Omniscol. The only constraint is that your system responds in the **format Omniscol expects** for that endpoint (same JSON structure): there is no intermediate transcoding. For that endpoint, Omniscol's local copy is not queried; the data displayed is the live data of your system.

Example: the institution's **subject catalog** served on the fly from your information system, so that any update on the school side is immediately visible in Omniscol, without a re-import.

Authentication to your system goes here in the **headers specific to the endpoint** (for example an `Authorization: Bearer ...`), entered on the override's row. The HTTP **method** can also be enforced per endpoint.

Rewiring an endpoint of the application itself to a live external source is a possibility as powerful as it is demanding: to be handled with your IT department, and in coordination with Omniscol.

Inserting your own server or a proxy

You can also route the calls through **your own server** or a **proxy** — for example to allow cross-origin resource sharing (CORS), or to insert in-house logic between the web application and Omniscol.

Disabling an endpoint

You **disable** an endpoint by giving it **no redirect URL** (method `null`).

Weigh the effect on the interface. Omniscol is a **single-page web application** (SPA) whose interface elements are driven by the **available endpoints**: buttons, tabs and menus only appear if the endpoint they depend on exists.

Disabling an endpoint therefore **dynamically removes**, at the next refresh, the interface elements that depend on it — and if you disable all the endpoints of a module, the **entire module** disappears from the navigation. These elements are removed, not merely hidden, and all of this happens **without any change to the code**: just edit the configuration and reload the application.

The table lists, per endpoint: its **key** (the internal code of the operation), its **original URL**, the HTTP **method**, the **new URL** of the redirect and specific **headers**. A search field lets you find the endpoint to override.

Hooks (outbound calls)

A **hook** asks Omniscol to **send an HTTP request to your URL after** an operation has succeeded. It is the mechanism for **keeping an external system informed in real time** — a display panel, a digital workspace, a human resources system, an in-house synchronization...

A hook can be attached in two ways:

- to a **specific endpoint** (the operation's key);
- to a **grouped event**, which covers a whole family of operations at once. Three grouped events exist:
 - **timetable change** (creation, move or deletion of lessons, activation of a timetable, and teacher or class absences as soon as a date is affected);
 - **teacher change** (addition, update or deletion);
 - **subject change** (custom subjects).

For each hook, you fill in the **callback URL**, the HTTP **method**, the **"with data"** checkbox (should the original request body be attached?) and its own **headers**.

What your server receives

The call is sent as `application/json` and carries, in addition to your headers:

- the **body of the original request** if the "with data" option is enabled;
- a block of Omniscol **metadata**: the called URL, the endpoint code, the method, the parameters, the user's authentication token and the school identifier;
- traceability headers: `X-OS-original-query`, `X-OS-original-endpoint`, `X-OS-auth` and `X-School`.

For the **timetable change** event, when the "with data" option is active, the call additionally includes a **diff of the lessons** (lessons added, changed, deleted) — handy for propagating only what has changed.

Behavior

Hooks fire **in the background**, **after** the user's operation has completed: they do not slow down the interface and **do not block it** if your server fails. An outbound call that does not go through is **logged**, without interrupting work in Omniscol. Each call has a short timeout (a few seconds): your server must **acknowledge receipt quickly** and process the rest on its side.

Good to know

- The Omniscol API exposes a **subset** of operations; an override or a hook only applies to the endpoints actually exposed. See [Omniscol API](#) for the list and authentication.
- For an **integration with a software package** (ERP, HR system, digital workspace), the dedicated synchronization is often a better fit — see [Synchronization with external systems](#). Internally, the same hook system is used.
- For an **AI agent** to read your data without any development, see [MCP — connect an external AI agent](#).

See also

[Omniscol API](#)

[MCP — connect an external AI agent](#)

[Synchronization with external systems](#)

9.6 Complete data model: JSON entities, relationships and ontology

Source: [help/en/integrations/data-model-full.md](#) · id: [integrations.data-model-full](#) · Audience: [admin](#) · Status: [stable](#) · Updated: [2026-06-18](#)

This page describes the **complete data model** of an Omniscol account: the entities, their main fields and their relationships. It extends the conceptual page [Data organization](#), which explains the *why* (school repository ↔ timetable, local copies); here, the goal is the *what* — an actionable structural map, in particular to **map the model to an external repository** (management software, directory, information system).

Technical page. This reference is aimed at integration and matching with an external system. It is **not required for everyday use** of Omniscol: to understand how the data is organized on the business side, see [Data organization](#).

All of a school's data is organized as a tree. The entities described below (users, school years, timetables, absences, events, etc.) correspond to the different branches of this structure. The normative reference is the account's **JSON schema**. It can be consulted in two forms:

- the **raw source** (the JSON schema as such), served by the account at `https://api.omniscol.com/api/guest/school_schema.json`;
- the **tree viewer**, readable at [omniscol.com/en/datamodel](#) (and the corresponding API reference at [omniscol.com/en/developers](#)).

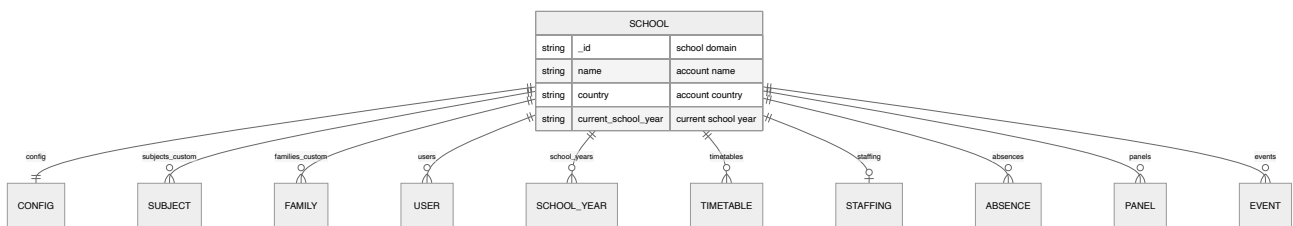
This schema describes exactly the physical storage in the database, as a JSON document. This page provides an organized reading of it, stable over time.

The school document: the root

The root document gathers the school's durable repository and all its planning data. Its top-level subtrees:

Root key	Entity	Role
<code>config</code>	<code>SchoolConfig</code>	Account settings (country, options, time zone, external synchronization)
<code>subjects_custom</code>	dictionary of <code>SubjectFull</code>	The school's custom subjects
<code>families_custom</code>	dictionary of <code>Family</code>	Custom subject families
<code>users</code>	dictionary of <code>User</code>	User directory (all roles)
<code>school_years</code>	array of <code>SchoolYear</code>	School years and holidays
<code>timetables</code>	dictionary of <code>Timetable</code>	Timetables
<code>absences</code>	grouping of <code>Absence*</code>	Absences (teachers, classes, staff, students)
<code>staffing</code>	<code>Staffing</code>	Staff members and supervision module
<code>panels</code>	dictionary of <code>Panel</code>	Display panels
<code>events</code>	dictionary of <code>Event</code>	Agenda-type events
<code>api</code> , <code>snapshots</code> , <code>jobs</code> , <code>translations</code> , <code>logo</code>	miscellaneous	Technical settings and history

The subtrees described as "dictionary" are JSON objects whose **keys are stable identifiers** (the pivot of any external mapping — see the final section).



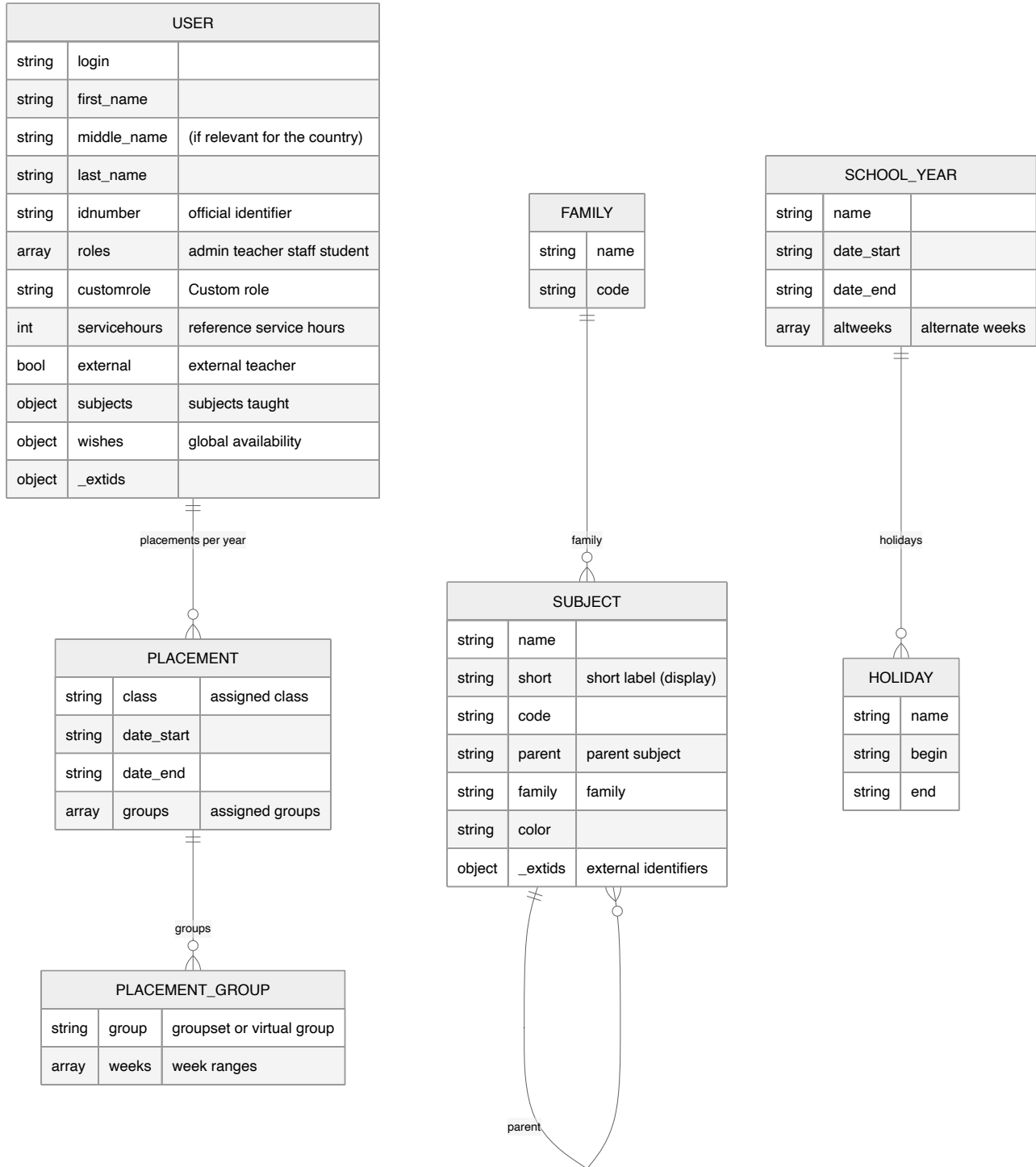
Cross-cutting fields: `_extids` and `wishes`

Two fields appear on many entities. To avoid repeating them in every schema, they are described once here.

- **`_extids`** (`ExternalIds`) — table of the entity's identifiers in external systems, for example `{ "auriga": "12345" }`. Present on synchronizable entities (subjects, families, users, sites, classrooms, resources, teachers, classes, groups...). It is the **anchor point for matching with an external repository** (see the final section).
- **`wishes`** — **time constraints**: availability, preferred or avoided time slots, maximum hourly volume, preferred classroom... Present on most schedulable entities: users, teachers and subjects of a timetable, groups, classes, classrooms, site hour ranges, staffing grids. Depending on the scope, these constraints are global (school level) or specific to one timetable.

Level 1 — The school repository

The school repository contains what is true for the institution, independently of any timetable: the subject catalog, the user directory, the calendar of school years.



Subjects — SubjectFull / Subject / Family

A custom subject (`SubjectFull`) extends the base subject (`Subject`: `name`, `short`, `code`, `type`, `_extids`) with `parent`, `family` and `color`. Two origins coexist at the school level: the country's **common subjects** (read-only) and the school's **custom subjects**. Functional details: [Managing subjects](#).

Users — user (including teachers)

A `User` is the **single entity** of the directory: identity (last name, first name, `idnumber`, contact), authentication (`login`), `roles`, reference service hours (`servicehours`), global availability (`wishes` field) and `placements`. A

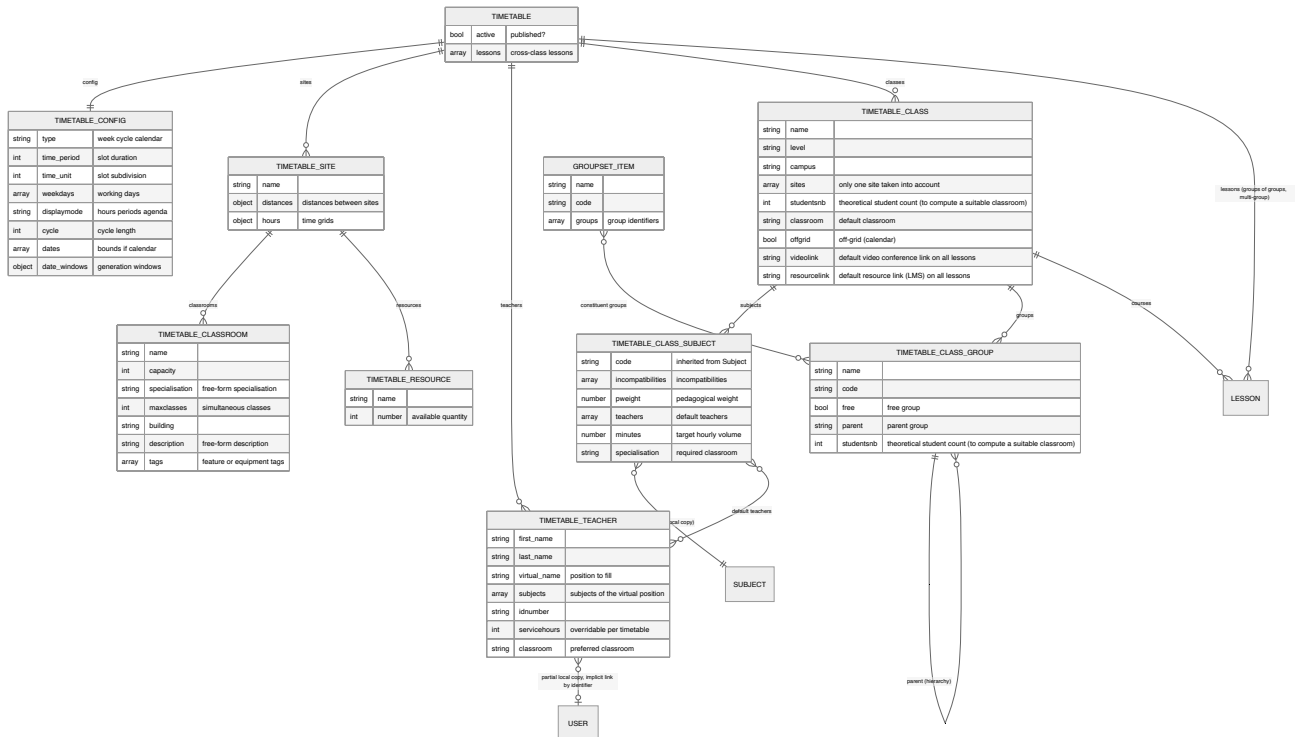
teacher is not a separate entity: it is a `User` with `teacher` among its `roles` (the same user can hold several roles). The `placements` field (indexed by school year) attaches a **student** to a `class` and to `groups` over a period. Details: [Managing teachers](#).

School years — `SchoolYear` / `Holiday`

A `SchoolYear` defines `date_start` → `date_end`, the list of `holidays` and the `altweeks` (alternate weeks). It is a **time frame**, not a container: timetables unfold within it without being nested in it. See [School year](#).

Level 2 — The timetable

A `Timetable` is a coherent planning unit. It carries its own local copy of teachers, classes, groups and subjects — see the local-copy principle in [Data organization](#).



Configuration — `TimetableConfig`

The decisive field is `type`: `week` (repeated weekly), `cycle` (multi-day cycle, see `cycle`) or `calendar` (real dates, see `dates`). `time_period` and `time_unit` define the grid's framework; `weekdays` the working days; `date_windows` the constraint windows for automatic generation.

Sites, classrooms, resources

A `TimetableSite` (site) contains its `classrooms` (`TimetableClassroom`: `capacity`, `specialisation`, `maxclasses`, `building`) and its `resources` (`TimetableResource`: `number` = available quantity, for which the system prevents overbooking). `distances` models travel times between sites. See [Sites, classrooms and resources](#) and [Classroom specialisations](#).

Timetable teachers — `TimetableTeacher`

An enriched partial copy of a `User`: only a few identifying fields are copied (`first_name`, `last_name`, `idnumber`), plus fields specific to planning (overridable `servicehours`, preferred `classroom`, per-timetable wishes). A non-empty `virtual_name` designates a **virtual teacher** (a position to fill, with no real `User` behind it).

Classes, class subjects, groups

- `TimetableClass`: `name`, `level`, `campus` / `sites`, `studentsnb`, and three key subtrees — `subjects`, `groups`, `courses`.

- **TimetableClassSubject** : local copy of a subject (extends `subject`) enriched for planning — `minutes` (target volume), `pweight` (pedagogical weight), `incompatibilities`, default `teachers`, `classroom specialisation`. Assigning a subject **with a course type** creates a separate entry per type. See [Courses](#), [lessons](#), [course types](#).
- **TimetableClassGroup** : a subset of a class. `free=true` disables conflicts with the rest of the class (**free group**); `parent` establishes the [group hierarchy](#).

Rules between groups — divisions, alignments, groups of groups

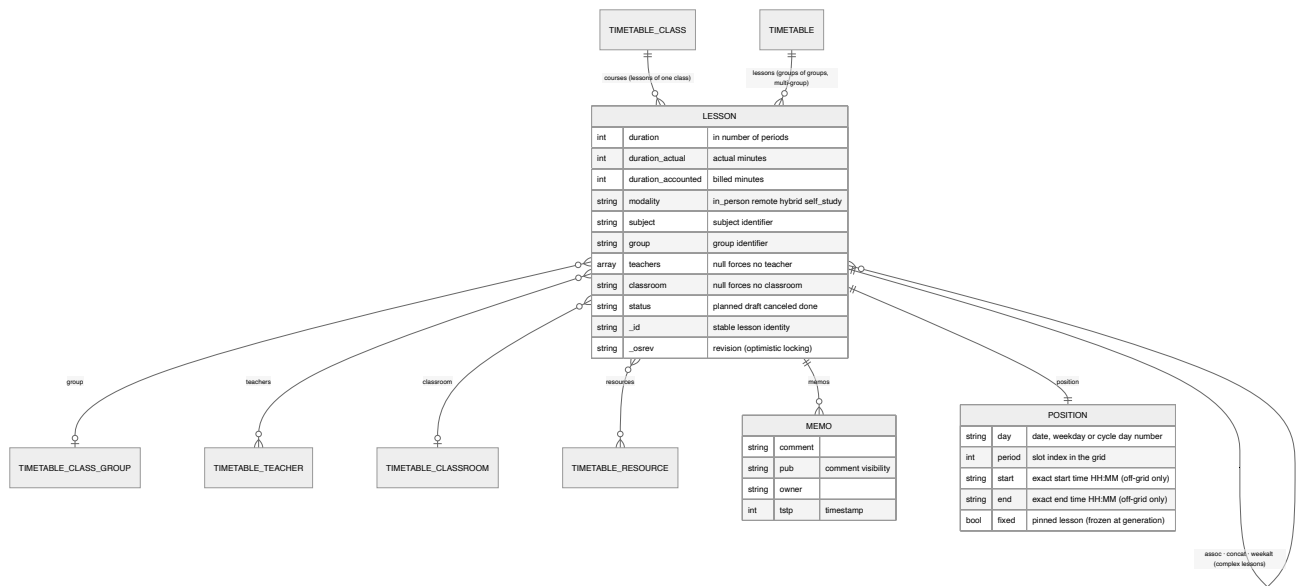
The relationships between groups are carried by the timetable's `groups` subtree, in three forms:

Schema	Entity	Meaning	Page
<code>TimetableGroupTimeset</code>	division	Mutually exclusive groups of the same class , placed in parallel (half-classes, electives)	Class divisions
<code>TimetableGroupSpaceset</code>	alignment	Groups from different classes working together on mirrored time slots	Alignments
<code>TimetableGroupGroupset</code>	group of groups	Meta-group (<code>GroupsetItem : name , code , groups []</code>) combining several groups	Groups of groups

See also the overview [Class, group, subgroup](#).

Level 3 — Lessons

A **lesson** (`Lesson`) is the teaching unit actually placed in the timetable: duration, subject, group, teacher(s), classroom, resources, position and status. It is generated from a class subject (see level 2) and attached **either to a class** (`class.courses`), **or directly to the timetable** (`timetable.lessons`) — the latter case for groups of groups and multi-group lessons spanning several classes.



Core of a lesson

A lesson carries the planning fields — `duration` (in periods), `duration_actual` / `duration_accounted` (actual / billed minutes), `modality`, `subject`, `group`, `teachers`, `classroom`, `resources` — plus its `position`, its `status`, its `memos` and an identity (`_id`, `_osrev`). Setting `teachers` or `classroom` to `null` **explicitly forces** the absence of a teacher or classroom.

Complex lessons

A lesson can **group other lessons**:

- `assoc` — **associated** lessons: the groups swap together.
- `concat` — **concatenated** lessons: strictly consecutive.
- `weekalt` — **alternate weeks**: one lesson variant per week.

These mechanisms cover **complex lessons** — see [Complex lessons: alternate, associated, concatenated](#).

Position of a lesson

The `position` is an **object** that places the lesson in time. It gathers the following fields:

- `position.day` — the day of the lesson, **polymorphic depending on the timetable mode**: a day name (`monday` ... `sunday`) in weekly mode, a **cycle day number** (`"1"` , `"2"` ...) in cycle mode, or a `YYYY-MM-DD` **date** in calendar mode. It is one and the same field that changes form depending on the mode; it is always present.
- `position.period` — the **slot index** in the time grid. A lesson with neither `period` nor `start/end` corresponds to a public holiday.
- `position.start` / `position.end` — exact start and end times (`HH:MM`), for **off-grid** lessons or calendar-mode lessons that do not align with a standard time slot.
- `position.fixed` — boolean (false by default): a **pinned** lesson, which **automatic generation will not move**.

The lessons of a class live in `class.courses`; the `timetable.lessons` array carries the **cross-class** lessons tied to groups of groups.

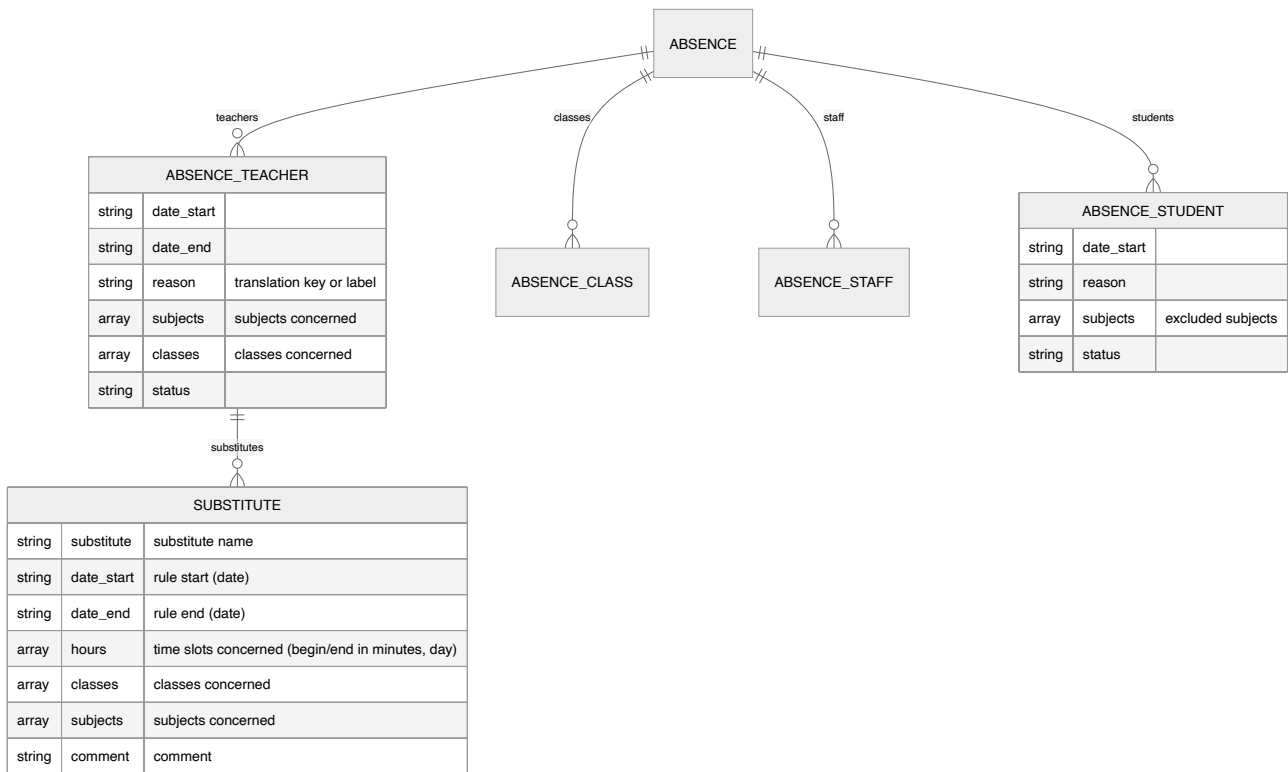
Identity of a lesson

Each lesson carries two technical identity fields, **present on Premium accounts**:

- `_id` — a **stable** identifier, assigned when the lesson is created and immutable afterwards (it combines a timestamp and a fingerprint of the content). It is the **key of a lesson** for collaborative editing and for an external tool that tracks lessons over time. For the **recurring occurrences** of a calendar timetable (alternate weeks, concatenations, associations), the `_id` of each occurrence is **derived** from that of the base lesson, which remains recognizable.
- `_osrev` — a **revision** token (optimistic locking), incremented on every modification (timestamp + modification index), in a strictly alphabetical order (which makes it easy to tell whether one revision is more recent than another). It lets collaborative editing **detect and merge concurrent modifications**: an outdated client revision signals a conflict, which the server resolves or merges instead of silently overwriting.

Absences

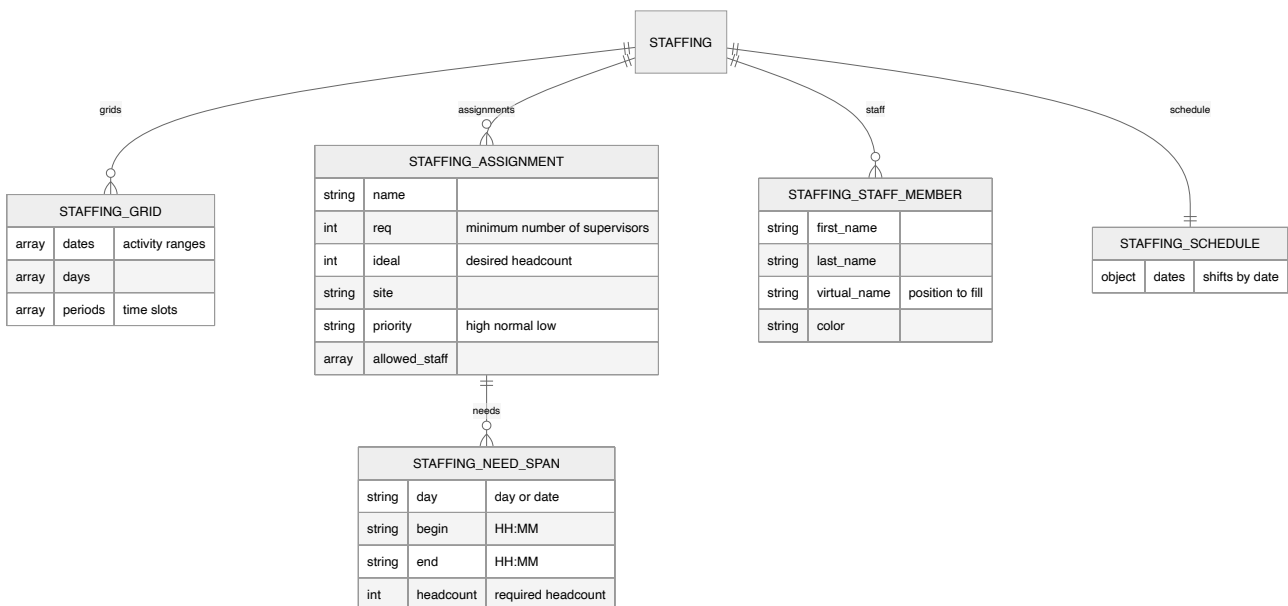
Absences share a common base (`Absence` : `date_start` , `date_end` , `reason` , `hours` , `comment` , `status`) specialized per type of absentee.



An `AbsenceTeacher` can be restricted to certain `subjects / classes` and carries an **array of substitution rules** (`substitutes`, each an `AbsenceTeacherSubstitute`: `substitute`, `period`, `time slots`, `classes` and `subjects` covered). Statuses differ per absentee type (for example `ok / aborted` for a class). See the glossary entry [Substitution / Replacement](#) and the **Absence management** module.

Staff members

A separate module (also sold standalone) for supervision and staff duty services.



A `staffingGrid` is the template (`days`, `periods`); the `assignments` describe the positions (`req / ideal`, `priority`, detailed `needs` per time slot via `StaffingNeedSpan`); the `schedule` materializes the actual shifts by date. See the Staffing module.

Events

An **Event** is an **agenda entry laid over the grid**: something that happens in the school **without being a regular lesson** — a class council, a parent-teacher meeting, a one-day exam, a field trip, an open day. Events are stored in the `events` dictionary (keys `event-<n>`) and are part of the Premium features.

Fields of an `Event` (required: `title`, `start`, `end`):

- `title` — displayed title.
- `start` / `end` — start and end, in `YYYYMMDDTHHmmSS` format.
- `rrule` — optional **recurrence** rule.
- `attendees` — participants: a user, a class, a group, a free-form label (`custom`), or the whole school (`everybody`) / anyone willing (`anyone`).
- `location` — place(s): a timetable classroom or a free-form label.
- `resources` — resources booked for the event.
- `videolink` — video conference link to attend remotely.
- `memos` — comments; `color` — color (hexadecimal).

How it works in practice (creation, placement on the grid, interface fields) is described in [One-off events](#).

Display panels

A `Panel` is a **display panel** showing the day's lessons (lobby, room, welcome screen...). It defines the selection of `columns`, the `topline`, the `filters/exclusions` and appearance settings. See the glossary entry [Display panel](#).

Identifiers, local copies and mapping to an ERP

To link the Omniscol model to an external repository, three principles come first:

1. **Dictionary keys are the stable identifiers.** Subjects, users, teachers, classes, groups, classrooms, resources are indexed by an immutable identifier (the JSON key), not by their label. This pivot — never the `name` — is what must be used for any correspondence.
2. **The `_extids` field (`ExternalIds`) carries the external identifiers.** Present on synchronizable entities (`Subject`, `Family`, `TimetableSite`, `TimetableClassroom`, `TimetableResource`, `TimetableTeacher`, `TimetableClass`, `TimetableClassGroup`, `User` ...), it associates each entity with its identifiers in third-party systems, for example `{ "auriga": "12345", "aimaira": "67890" }`. It is the canonical anchor point for a bidirectional mapping.
3. **The school ↔ timetable local copy is deliberate.** A class subject or a timetable teacher is an enriched copy, not a live reference. An external repository must therefore decide at which level it maps: the **school repository** (durable catalog) or a specific **timetable** (dated planning). The detailed propagation rules are in [Data organization](#).

The synchronization configuration lives in `config.extsync` (`SchoolConfigExtsync`): `systems` (configured connectors), `sync` (direction and entities), `export` and `schedules`, plus correspondence tables (`mappings`) by key and Omniscol identifier. For programmatic access, the REST API and tokens are described in [API tokens](#); the synchronization connectors in [External synchronization](#).

The exact and exhaustive shape of each field remains defined by the account's reference JSON schema (https://api.omniscol.com/api/guest/school_schema.json, readable viewer at omniscol.com/en/datamodel). If this page and the schema diverge, **the schema prevails**.

See also

Data organization

Class, group, subgroup

Class divisions

Alignments

Groups of groups

Courses, lessons, course types

Complex lessons

Sites, classrooms and resources

School year

API tokens

External synchronization

9.7 MCP — connect an external AI agent to Omniscol

Source: <help/en/integrations/mcp.md> · id: [integrations.mcp](#) · Audience: admin · Plan: premium · Updated: 2026-06-13

PREMIUM

PREMIUM

The **Model Context Protocol (MCP)** is an open standard that lets an MCP-compatible AI assistant consume business tools exposed by an external service. Omniscol exposes a large part of its API as an **MCP server**: an AI agent can query your account through the authorized API endpoints or permissions, once authenticated — by OAuth2, MCP's standard mode, or by a token.

What the agent can do

The MCP agent performs very well on **read-only questions** where the data lives in Omniscol:

- "Give me the room occupancy rate against opening hours this week."
- "Find me a room available on three Mondays in October on the same 2-hour slot."
- "How many lesson hours has Jean Dupont taught over the current school year?"
- "What are all of today's math lessons?"
- "List the teachers who have completed less than 70% of their service hours this term."

These queries typically cross several tools: timetable, statistics dashboards, computed availability, teacher records. The agent orchestrates the calls and phrases the answer in natural language.

The exposed tools

The MCP server builds its tools from the Omniscol API routes authorized for MCP. Explicitly ignored routes, and some technical modules (for example, those related to user authentication) do not become tools. The list therefore reflects the API exposed to MCP, not the application's entire internal surface. It is nonetheless a very large part. What is more, a number of routes are dedicated to MCP and not used by Omniscol itself. This is the case for advanced search tools, so that free text naming an entity ("teacher Jean Dupont", "class 6A") is resolved by Omniscol to its technical identifier, then used to query the data precisely. It is also the case for complex tools where Omniscol has no graphical interface, because a prompt suits them better: entity occupancy and availability search, for example ("find me a room available for 2 hours in the afternoon on 3 Mondays in a row", "is a math teacher available in the week of 14 October for 3 hours?").


The tools notably cover:

- the **Administration** module (users, subjects, school years, settings),
- the **Timetable management** module (configuration, sites, rooms, classes, lessons),
- the **Timetable** module (schedule, dashboards, searches),
- the **Absence management** module (declaring absences, statistics),
- global search.

Read routes are the best suited to agentic use. Some write operations may exist in the catalog, depending on the token's rights and the available API, but they must remain supervised: a request that modifies several business

objects must be verified by a user before being considered reliable.

Enabling the MCP server

The MCP server is available on **Premium** accounts. Everything starts from the **MCP screen**, opened from the **Administration** module with  **Configure**: it shows the **server URL** for the chosen scope (global or restricted to one module) and provides, ready to copy, the configuration elements for your client.

MCP's standard authentication is OAuth2. A compatible client (such as Claude) simply connects to the server URL, discovers the account's OAuth2 configuration there, and the user approves the access through a consent screen: there is **nothing to provide other than the URL**. The granted scope follows the account's rights and its visibility restrictions. Omniscol's OAuth2 server, its client management screen and the details of consent are described on [OAuth2 / OIDC \(provider\)](#).

For a client that does not support OAuth2, the same screen generates a **token** (API key, expiry, optional associated user, write rights to tick) then offers, ready to paste, the formats useful for each case: `Authorization: Bearer` header, URL with token, **Claude Desktop** configuration block (free mode) and local proxy command. The token relies on the key system described in [Omniscol API](#).


Security best practices

- **OAuth2 authentication** — more convenient, and to be preferred when your agent supports it (paid version of Claude).
- **Token dedicated to the AI** — create a token with a meaningful label (`AI agent - Claude desktop`) that you can revoke if needed.
- **Minimal scope** — with an API token, select only the API endpoints you need. With an OAuth token, limit the scopes to the actual need.
- **Activity log (logs)** — a call appears in the logs associated with the token when the route in question is logged. These logs trace the call; they keep neither the detail of the data returned nor a replayable copy of the request.
- **Visibility restrictions** — the agent sees what Omniscol returns to it. If you have configured strict visibility restrictions for the token's role, they apply.

How-to

Connecting Claude to Omniscol

The recommended path is **OAuth2 authentication**: you connect Claude to the MCP server through its URL, without handling any token.

1. **Enable the MCP server** on your Premium account and retrieve its URL (typically `https://your-school.omniscol.com/mcp`). The MCP screen (**Administration** module,  **Configure** button) shows it for the chosen scope, with a copy button.
2. **Add Omniscol as a connector in Claude.** In Claude's connector settings, add a custom connector and paste the Omniscol MCP server URL.
3. **Approve the access.** Claude redirects you to the Omniscol consent screen: sign in and authorize the access. The granted scope follows your account's rights and any visibility restrictions.
4. **The tools appear in Claude**, which calls them whenever your request lends itself to it.
5. **First test:** *"How many lesson hours has Jean Dupont taught this year?"* — Claude cross-references the accessible data and answers in natural language.

Alternative method with a token. For an MCP client that does not support OAuth2, generate a dedicated token from the MCP screen (meaningful label, endpoints limited to the actual need, write rights ticked explicitly) and pass it in an `Authorization: Bearer` header. The MCP screen provides the matching configuration block. Prefer OAuth2 as soon as your client supports it.

See also

[Built-in AI assistant](#)[Omniscol API](#)[OAuth2 / OIDC \(provider\)](#)[Integrations overview](#)

9.8 Built-in AI assistant

Source: [help/en/integrations/ai-assistant.md](#) · id: [integrations.ai-assistant](#) · Audience: [admin](#) · Status: [stable](#) · Updated: 2026-06-28

The **built-in AI assistant** is Omniscol's conversation surface for asking a question in natural language from the interface. It opens in the [AI Assistant](#) tab of the help panel, reserved for administrators.

The assistant relies on the help corpus and, depending on the account's contract, on the Omniscol tools exposed to the agent. The scope depends on the account: a documentation assistant answers from the online help, and accounts whose contract includes it add access to business tools that query the account's data.

To connect an external AI agent to your account, use [MCP — connect an external AI agent](#).

What the assistant covers

The assistant is designed for two families of requests:

- **documentation questions** — finding a procedure, explaining a concept, pointing to the right help page;
- **operational questions** — querying the account's data through the authorized tools, for example about lessons, rooms, teachers, absences or statistics.

Examples:

- "How do I publish a timetable?"
- "Where do I configure the iCal links?"
- "Which math lessons are scheduled today?"
- "Why does this timetable generation leave lessons unplaced?"

Rights and scope

The assistant does not have unlimited access to the account. Its answers and its actions depend on:

- the account's contract, which determines whether business tools are enabled;
- the rights of the signed-in administrator;
- the tools actually exposed to the agent;
- the data available in Omniscol.

An action that changes business data goes through an explicit confirmation before being applied. For external agentic integrations, use [MCP — connect an external AI agent](#) and dedicated tokens.

Best practices

- Review the proposed actions before applying them.
- Do not copy sensitive data into a conversation if it is not needed for the request.
- For an external agent, create a dedicated token with a minimal scope and a suitable expiration.

See also

[MCP — connect an external AI agent](#)[Omniscol API](#)[Omniscol plans and options](#)

9.9 OIDC / SSO — sign-in via an identity provider

Source: [help/en/integrations/oauth2.md](https://help.en/integrations/oauth2.md) · id: [integrations.oauth2](#) · Audience: [admin](#) · Plan: [premium](#) · Updated: 2026-06-13

👑 PREMIUM

PREMIUM

OpenID Connect (OIDC), built on OAuth2, lets your users sign in to Omniscol **with their organization account** rather than with a dedicated Omniscol password. Three benefits: a single password to remember, centralized deactivation when someone joins or leaves, and automatic enforcement of the security policy (MFA, password length, expiry...) managed on the IT side.


This page describes **user SSO**: people signing in to Omniscol through your identity provider. The opposite case — Omniscol as an OAuth2 / OIDC **server** that third-party services connect to, with its client management screen — is covered on [OAuth2 / OIDC \(provider\)](#).

Supported providers

Omniscol supports:

- **Google Workspace**,
- **Microsoft Entra ID** (formerly Azure AD),
- **Keycloak**,
- a generic **OIDC** provider via its issuer / discovery URL (any OIDC-compliant provider).

Configuring on the Omniscol side

In **Administration** → **Settings**, in the **Security** section, the  **Configure** button opens the OIDC configuration screen, where you declare the provider:

- **Display name** — the label shown on the sign-in button (for example `Continue with Google`).
- **Provider type** — Google, Microsoft Entra ID (Azure AD), Keycloak or generic OIDC.
- **Tenant / issuer** — the Microsoft tenant for Entra ID, or the issuer URL for Keycloak and generic OIDC providers. The OIDC metadata is then discovered by the application.
- **Client ID** and **Client secret** — obtained by creating an application on the provider side (see the provider's documentation).
- **Scopes** — `openid profile email` by default. Add provider-specific scopes if needed.
- **Authorized domains** — an optional list of email domains accepted at SSO sign-in; when left empty, no domain filtering is applied.
- **Redirect URL** — Omniscol gives you the exact value to copy into the provider-side configuration (along with the post-logout redirect URL if the provider requires it).
- **SSO-only login** — once SSO is validated, this option refuses Omniscol password sign-in, except for the **break-glass accounts** described below.

The current SSO configuration is defined at the school account level.

Break-glass access

In **SSO-only login** mode, Omniscol password sign-in is refused: everyone goes through the identity provider. If that provider becomes unavailable — outage, expired secret, configuration error — access to Omniscol would be blocked for everyone.

The **break-glass account** is the safeguard against this lock-out: an **administrator** account allowed to sign in with an Omniscol password even while exclusive SSO mode is active.

- In the account record of an **administrator**, when the **SSO-only login** option is active, a **Break-glass (SSO only)** checkbox appears. Ticking it allows this account to sign in with a password if the identity provider is unavailable. Only administrator accounts can be designated.

- Omniscol support also keeps its own independent emergency access.

Designate at least one break-glass account **before** enabling exclusive SSO mode, and keep its password in a safe place: it is your safety net if the identity provider goes down.

Configuring on the provider side (examples)

Google Workspace

1. Google Cloud console > **APIs & Services > Credentials**.
2. **Create an OAuth 2.0 credential**, type **Web application**.
3. Paste the redirect URL provided by Omniscol into **Authorized redirect URIs**.
4. Retrieve the **Client ID** and **Client secret** and paste them into Omniscol.

Microsoft Entra ID

1. Azure portal > **Entra ID > App registrations**.
2. **New registration**, type **Web**, Omniscol redirect URL.
3. **Certificates & secrets** > create a client secret.
4. **API permissions** > add `openid`, `profile`, `email`.
5. Retrieve the **Application (client) ID** and the secret.

Keycloak

1. Realm > **Clients > Create client**, type **OpenID Connect**.
2. **Valid redirect URIs**: the URL provided by Omniscol.
3. **Credentials** > retrieve the client secret.

Linking to Omniscol accounts

When a user clicks the SSO button, Omniscol:

1. Redirects them to the provider,
2. Retrieves their email address after authentication,
3. Looks for an Omniscol account with **the same email address**.


If the account exists and is active, the user is signed in. Otherwise, access is denied. Consequence: for SSO to work, users must exist in Omniscol with the right email address — creation is not automatic on the Omniscol side by default.

Automatic account provisioning

Automatic account creation at the first SSO sign-in (`just-in-time provisioning`) is not the default behavior of the configuration screen. If this mode is needed, it must be scoped with Omniscol.

☰ How-to

Configure an OIDC / SSO provider

1. **OIDC / SSO** lets your users sign in with their organization account (Google Workspace, Microsoft Entra ID, Keycloak, Okta...). The provider-side part depends on the provider.
2. **On the provider side**, first create a **Web application** OIDC application: retrieve the **Client ID** and the **Client secret**, and copy the **redirect URL** provided by Omniscol into the provider's *Authorized redirect URIs*.
3. **On the Omniscol side**, open the settings then the OIDC configuration. Fill in the **display name** (the sign-in button label), the provider type, the issuer or tenant as applicable, the Client ID, the Client secret and the **scopes** (default: `openid profile email`).
4. **Click**  **Test**. Check that the configuration works. Omniscol performs an OIDC round trip; on a discovery, secret or redirect error, the error message is explicit.
5. **Save**. The SSO sign-in button appears on the login page when the configuration is active.
6. **Important: accounts must exist** in Omniscol with the right email address. SSO sign-in matches by email; no automatic provisioning by default.

🔗 See also

[First login](#)[Users and roles](#)[OAuth2 / OIDC \(provider\)](#)[Integrations overview](#)

9.10 Synchronization with external systems (ERP)

Source: [help/en/integrations/extsync.md](#) · id: [integrations.extsync](#) · Audience: [admin](#) · Plan: [premium](#) · Updated: 2026-06-13

👑 PREMIUM

PREMIUM

Omniscol interfaces with schools' external ERPs and information systems through a unified **synchronization** framework. Omniscol acts as the **central point**: it drives the imports and the exports, while the external system remains the administrative source of truth.

Synchronization with external systems is available on **Premium** accounts, when a connector is configured for the account. What is handled **as a project** is the addition of a new ERP or a new information system not yet supported, with scoping of the mapping and the flows.


Flow model

The framework covers two directions of flow:

- **Entities (inbound)** — Omniscol can retrieve from the ERP the teachers, subjects, classrooms, classes, students and their external identifiers, depending on the connector and the configuration. The ext-id references are kept in Omniscol for matching.
- **Lessons (outbound)** — Omniscol pushes the lessons produced by its scheduling to the ERP, with the correct ext-id references. Depending on the connector, the export can be triggered on demand, scheduled, or replayed as a full batch to catch up on divergences.

Omniscol orchestrates the configured imports and exports, keeps the external identifiers, manages the synchronization passes and exposes verification screens to reconcile the data between Omniscol and the external system.

Source and local creation rights

For each entity type (teachers, subjects, classrooms, etc.), you select the source system and indicate whether local creation and local updates remain allowed. The screen also exposes an  **Association** button to match a local Omniscol entity with an ERP entity when both already exist.

Bridging ontologies

Each ERP has its own vocabulary (for example a "subject" in Omniscol = **Unité Pédagogique** in Auriga = **Cours** in Aurion). The framework handles the **mapping**: Omniscol links the concepts on one side to those on the other, so you do not have to rewrite your nomenclature.

Available connectors

Available

- **Aurion** — connector available in production. An ERP **published by the Auriga company**, particularly widespread in French higher education (engineering and business schools). See [integrations.aurion](#).
- **Auriga** — connector available in production. **Auriga the software** is the full-web evolution of Aurion, by the same company. See [integrations.auriga](#).

Directory standard

- **OneRoster** — instead of a vendor-specific connector, Omniscol supports the international **OneRoster** standard (1.2 and advanced groups) to exchange a school directory with a compliant SIS, digital workspace or LMS. Importing a OneRoster directory falls under this Premium synchronization; the export side is served read-only and controlled by OAuth2 scopes. See [OneRoster](#).

On request

Any ERP or information system exposing a **documented API** for the import / export actions and a working **sandbox** can be integrated as a project. Contact Omniscol support to scope the addition of a new connector.

Difference with systems that query Omniscol

Some systems can also consume Omniscol through the **GET** endpoints of the public API: in that case, Omniscol does not orchestrate an ERP synchronization, the third-party system only queries the data its token gives access to. This model is different from the synchronization orchestrated by Omniscol:

Model	Who orchestrates?	Direction	Example
Synchronization with external systems	Omniscol	Bidirectional (entities in, lessons out)	Auriga, Aurion
Partner pull	The partner	The partner performs GET calls	Third-party connector via API

The partner-pull model needs no synchronization-with-external-systems configuration: the public API and the authentication tokens are enough. See [Omniscol API](#).

Configuration

Setting up synchronization with external systems typically requires a session with Omniscol support to:

1. **List the objects to synchronize** — users, classes, instructors, subjects, classrooms, etc.
2. **Calibrate the ontology mapping** — which ERP concept corresponds to which Omniscol concept.
3. **Define exclusivity** per entity type.
4. **Schedule the frequency** of the passes (on demand, hourly, daily, more for catching up on divergences).
5. **Test on an ERP sandbox** before going live.

Verifications and routine operations

- **Verification** of the configured favorites or datasets, depending on the connector.
- **Manual triggering** of a lesson export from the administration screen.
- **Scheduled full export** when the option is configured in the export block.
- **Aurion XML Pivot** when the Aurion connector is used.
- **Verification of the matches**: the association screens flag the entities to pair or to correct.

☰ How-to

Verify and trigger a synchronization

1. **Once synchronization has been configured** by Omniscol support, routine operations are done from the administration screen: verify the favorites/datasets, pair the entities, trigger a one-off export, download the Aurion pivot when available, or purge the export state in case of a switchover.
2. **In the system configuration**, use the **Verify** buttons of the configured favorites or datasets. Omniscol retrieves the connector's data and reports configuration or format errors when the connector returns them.
3. **To push the lessons immediately** to the ERP without waiting for the next scheduled pass, click **▶ Trigger**. Useful after an urgent change.
4. **With Aurion**, the **XML Pivot** button downloads a file that the ERP administrator injects manually through the ERP console when this mode is chosen.
5. **In case of a major switchover** (structure change, end of year), the **🗑️ Purge Omniscol** button deletes **on the ERP side** all the lessons Omniscol exported there, to start from scratch. The action targets the selected connector and is **irreversible**: only to be used after agreement from support.

🔗 See also

[integrations.aurion](#)[integrations.auriga](#)[OneRoster](#)[Omniscol API](#)[Import and export](#)[Integrations overview](#)

9.11 OneRoster (1.2 and advanced groups)

Source: [help/en/integrations/oneroster.md](#) · id: [integrations.oneroster](#) · Audience: admin · Plan: standard · Updated: 2026-06-29

OneRoster is an international standard for exchanging school rosters (published by 1EdTech): it describes, in a common format, a school's **organizations, school years, courses, classes, enrollments and users**. It is used to connect an SIS, a VLE or an LMS to another tool without re-keying data or proprietary formats.

This page is intended for the school's **IT department**. It describes what Omniscol exposes and consumes in OneRoster, the expected authentication, and the actual scope of each direction of exchange.

Supported versions

Omniscol supports **OneRoster 1.2 Rostering** — the base layer (Org, AcademicSession, Course, Class, Enrollment, User, Demographics) — as well as the **OR-Groups (Advanced Groups Service)** layer, the additive groups standard that adds three entities (Group, GroupMembership, GroupAssociation) on top of 1.2.

The OR-Groups layer is a **standard still being published**. It is strictly additive on top of 1.2: a Group only **references** the Class / User / Org / AcademicSession entities of 1.2, read-only. OneRoster version 1.3 is expected to make Class a specialization of Group — a direction the Omniscol implementation is already converging toward.

Omniscol as producer (Omniscol → VLE / LMS)

As a **producer**, Omniscol exposes the school's roster **read-only**, via a OneRoster-compliant REST API, for a VLE or an LMS to read.

- The Rostering 1.2 base layer is served under `/ims/oneroster/rostering/v1p2/...` (for example `/orgs`, `/schools`, `/academicSessions`, `/courses`, `/classes`, `/enrollments`, `/users`, `/teachers`, `/students`, `/demographics`, each with its nested variants).
- The groups layer is served under `/ims/oneroster/groups/v1p0/...` (`/groups`, `/groupMemberships`, `/groupAssociations` and their nested variants).
- The scope is the **current** school year by default; a parameter targets another year or covers all years.

The data reflects Omniscol's consolidated planning (active / published timetables). The producer is **read-only by design**: no write operation (PUT / DELETE) is exposed — a remote system does not drive the creation or deletion of entities in Omniscol.

Producer authentication

The producer endpoints are protected by **OAuth2 with the client_credentials flow** (a machine-to-machine token, with no user), issued by Omniscol's OAuth2 server. They are **not** reserved for Premium accounts: access is controlled by OAuth2 **scopes**, not by the plan. Each scope is specific to one service, and access to demographic data is compartmentalized within 1.2:

Endpoints	Accepted scopes
Rostering 1.2 (excluding demographics)	<code>https://purl.imsglobal.org/spec/or/v1p2/scope/roster-core.readonly</code> , <code>https://purl.imsglobal.org/spec/or/v1p2/scope/roster.readonly</code>
Rostering 1.2 <code>/demographics</code>	<code>https://purl.imsglobal.org/spec/or/v1p2/scope/roster.readonly</code> , <code>https://purl.imsglobal.org/spec/or/v1p2/scope/roster-demographics.readonly</code>
OR-Groups (all)	<code>https://purl.imsglobal.org/spec/or-groups/v1p0/scope/roster-group.readonly</code>

An OR-Groups token does not grant access to the Rostering base layer, and vice versa. No write scope is advertised. Privileged scopes are granted by the Omniscol administration when the OAuth2 client is registered; a client cannot self-assign them. OAuth2 client and token management is described on [OAuth2 / OIDC \(provider\)](#) and [Omniscol API](#).

Student membership is carried by the groups layer

A structuring choice: the list of students in a class or a group is exposed via **OR-Groups** (GroupMembership), not by the 1.2 base layer alone. A consumer that only reads Rostering 1.2 gets the catalog (Courses, Classes), the teacher → course enrollments and the user directory, but **not** student membership. To know who is in which class or group, the consumer must implement the OR-Groups layer. This choice reflects the French model: a student is enrolled in a **class** or a **group**, not subject by subject.

The producer is exposed as REST. The **OneRoster CSV bundle** format (a zip archive with one file per collection) is supported on the import side, for providers that deliver their roster as files rather than through an API — see the consumer section below.

Omniscol as consumer (SIS → Omniscol)

PREMIUM

As a **consumer**, Omniscol imports the roster of a OneRoster-compliant SIS and reconciles it into the school. This consumer configuration is part of **synchronization with external systems**: it is available on **Premium** accounts and is scoped as a project with the Omniscol team (see [Synchronization with external systems](#)).

- **Transport** — either the provider's **REST API** (the base address is configured, standard OneRoster pagination is followed), with OAuth2 `client_credentials` authentication against the provider's server; or a **CSV bundle** (zip archive).
- **Profile** — a configuration profile sets the scope to read: the 1.2 base layer alone, the groups layer, or the French mapping for school education. A purely 1.2 provider, with no groups service, imports cleanly.
- **Controlled application** — the import follows the same principle as the other connectors: Omniscol fetches then reconciles the data, and an **administrator validates** its application to the school. The remote system never pushes directly into Omniscol.
- **Idempotent re-imports** — external identifier mappings are kept, so a re-import creates no duplicates even when the provider renames a label.

French profile (school education)

The **French profile for school education** maps French notions onto the OneRoster model: a **division** (the class in the enrollment sense) becomes a Group of the main organizational type; a **group** becomes a Group for teaching delivery; a **group of groups** a cross-cutting Group; **class divisions** and **alignments** become group associations; the assignment of a **teacher** to a course becomes a teacher enrollment. Identifiers such as the INE or the staff identifier are carried in the user's `userIds`.

Identifiers and privacy

Each exposed entity carries a stable identifier, the `sourcedId`.

- **Structural** identifiers (organization, year, course, class, group) are stable and immutable from one export to the next: a consumer can rely on them to correlate data over time.
- The **user** identifier is **anonymized**: Omniscol never puts its own nominative identifier on the wire. The token put on the wire is derived by a one-way function (HMAC) specific to the account; enrollments and memberships reuse this token without ever exposing the original identifier.

Status

- **Producer** — available and demonstrable without a remote partner (the export stands on its own). Rostering 1.2 and the OR-Groups layer are served read-only, with OAuth2 authentication.
- **Consumer** — the import configuration is part of Premium synchronization. Bringing a **live** synchronization with an SIS into service depends on the provider and, where applicable, on ministerial calendars: it is **scoped on request**, connector by connector. It is not an immediate turnkey synchronization.

See also

[Synchronization with external systems](#)

[OAuth2 / OIDC \(provider\)](#)

[Omniscol API](#)

[Complete data model](#)

[integrations.partners](#)

9.12 Linked accounts and shared resources

Source: <help/en/integrations/linked-accounts.md> · id: `integrations.linked-accounts` · Audience: `admin` · Updated: 2026-06-13

Linked accounts connect **several distinct Omniscol accounts** — hence **several schools** — that share, in real life, **teachers and classrooms**. The link makes the **occupancy** of these shared resources visible from one account to the other: when you view or build a timetable, Omniscol surfaces the lessons from linked accounts that involve a teacher or a classroom you share, and flags them as already busy. You thus avoid booking the same person or the same room twice, without having to call the other school. It is an option enabled on request by the Omniscol team.

Each account remains **fully autonomous**: it keeps its users, its settings, its school years and its administrative scope.

What the link shares

- **Shared**: the **occupancy** of the teachers and classrooms present in both accounts. It is what feeds the occupancy views and cross-account conflict detection.
- **Not shared**: the users, students, classes, settings and administrative data specific to each account. Seeing a busy time slot does not open access to the other account's details.

Matching is **automatic**: Omniscol recognizes the same teacher by name (the date of birth settles homonyms) and the same classroom by its site name and room name. For it to work, these teachers and classrooms must therefore carry the **same names** in each account.

Use cases

All involve **several distinct schools**, each with its own Omniscol account:

- A **group of schools** that pool teachers or classrooms — for example a network of schools in the same city.
- **Legally separate institutions** located in the same building, and therefore sharing the rooms.
- A campus where **several entities** (engineering school, business school...), managed on distinct accounts, use the same lecture halls and the same instructors.

Activation

Linked accounts are scoped with the Omniscol team, because you need to decide together:

- which accounts are linked;
- which common resources — teachers, classrooms — are shared;
- the harmonization of the names of the shared teachers and classrooms, on which automatic matching relies;
- which account remains responsible for each piece of data;
- how to handle the conflicts detected across scopes.

Once the option is active, shared occupancy appears in the occupancy views and feeds cross-account conflict detection, with no further action on your part.

Enabling or disabling a synchronization

When a timetable is synchronized with other active timetables — from the same account or from linked accounts — a **Synchronization** button appears: in a timetable's **reorganization mode**, as well as when **editing a calendar timetable**, on the **shared dates**. It opens the list of these synchronizations: each **other active timetable** of the account and each **linked account** that shares teachers or classrooms.

All are **enabled by default**. Untick one to **stop taking into account** the occupancy of that timetable or that account in the current view, then tick it again to re-enable it. The button only appears when there is at least one synchronization to manage.

Related case: several active timetables in a single account

Linked accounts coordinate distinct schools; the same need also arises **within a single account**, when a school runs **several active timetables in parallel**. The same shared-occupancy principle then prevents a classroom, a teacher or a **class** from being booked twice by two simultaneous timetables.

This is typically the case for a **multi-program organization** with different rhythms — an 18-month EMBA, a Grande École running calendar semesters, a preparatory class on a weekly timetable — that shares rooms and instructors across its programs; or for a **class straddling two calendars**, with a recurring organization in the morning and a one-off organization in the afternoon.

This coordination is included with **Premium** and can, exceptionally, be enabled on a Standard account after scoping with Omniscol. See [Multiple active timetables in parallel](#).

 **See also**

[Multiple active timetables in parallel](#)

[Sites, classrooms, resources](#)

[Synchronization with external systems](#)

10. Portals

10.1 Student portal

Source: <help/en/portal/student-portal.md> · id: [portal.student-portal](#) · Audience: [student/admin](#) · Plan: [standard](#) · Updated: 2026-06-13

The **student portal** is the interface dedicated to learners. Simpler and more focused than the administrator interface, it exposes exactly what a student needs day to day: their personal timetable, that of their class or classes, their absences and the sharing of their calendar.

Personal timetable

On opening, the student sees **their own timetable** for the current week: subjects, times, classrooms, teachers. Three classic views are available:

- **Week** — standard 5- or 7-day grid depending on the school configuration.
- **Day** — chronological agenda, useful on mobile.
- **List** — sequential view of the upcoming lessons, without a grid.


Last-minute changes (a lesson moved, a classroom change, a cancellation) are reflected: the student sees the up-to-date state each time the view is opened or reloaded.

Viewing the class timetable

Beyond their personal timetable, the student can view the **timetable of their class or classes** and that of the **groups** of those classes. Their access stops there: they cannot view the timetable of another class, of a teacher, or of a classroom.

The administrator can limit the students' **viewing horizon**: a setting defines the number of upcoming weeks visible, or sets access to **Unlimited**. See [Visibility restrictions](#).

Syncing with a personal calendar

The share icon  **Sharing** in the timetable view opens the Sharing window. Its **iCal** tab provides the personal subscription URL, to paste into Google Calendar, Apple Calendar or Outlook. The URL can be copied, downloaded as an `.ics` file or scanned via a QR code.

See [iCal — subscription and dynamic link](#) for the details. The calendar application refreshes the subscription at regular intervals: when the timetable changes in Omniscol, the update appears in the student's calendar.

Declaring an absence

If the school allows it, the student can declare their own absence from the portal: dates, times if applicable, reason and comment. See [Declaring an absence](#).

The request appears in the Absence management module with a pending status until an administrator approves it. The form does not include uploading a supporting document.

Interface language

The interface language follows the language of the user's browser. The student can choose another one from the Language entry in the user menu; this choice is remembered on their account for subsequent logins.

See also

[Teacher portal](#)[Guest portal \(parents, observers\)](#)[iCal — subscription and dynamic link](#)[Declaring an absence](#)[Visibility restrictions](#)

10.2 Teacher portal

Source: <help/en/portal/teacher-portal.md> · id: `portal.teacher-portal` · Audience: `teacher/admin` · Plan: `standard` · Updated: 2026-06-26

The **teacher portal** is the interface dedicated to teachers. It covers a teacher's three daily needs: viewing their timetable, entering their availability, declaring their absences. By default, it also gives access to the timetables of colleagues, classes and classrooms.


Teacher timetable

For the current week, the teacher sees the **lessons they teach**: subject, class, group, classroom, times. Three views available:

- **Week** — standard grid view.
- **Day** — chronological agenda view.
- **List** — chronological sequence without a grid, convenient on mobile.

Timetable changes (a lesson moved or cancelled, an exceptional lesson added) are reflected: the teacher sees the up-to-date state each time the view is opened or reloaded.

Entering availability

If the school has enabled availability entry, the teacher opens their personal record via the  **Information** entry in the user menu; the **Availability** tab presents a grid where they declare their preferences for the next timetable construction:

- **Unavailable** slots — a blocking constraint: the algorithm never places a lesson there, and a manual planner would create a conflict by doing so,
- **Undesired** slots — the algorithm avoids placing a lesson there, without an absolute ban, and a manual planner sees a warning there,
- **Preferred** slots — the algorithm favors these slots.

Unmarked ranges remain neutral: available, with no preference.

See [Teacher availability](#) for the exact meaning of the colors and best practices for entering them.

The teacher can also add **free-text comments** (`no teaching on Fridays because of another professional commitment`) that help the administration understand the context.

Declaring an absence

The  button lets the teacher report their own unavailability:


- dates concerned;
- full day or specific time slots;
- reason;
- short comment.

The request appears in the **Absence management** module with a **pending** status until an administrator approves it. Substitutions are then handled in that same module; the portal form does not move lessons and does not attach a supporting document. See [Declaring an absence](#).

Viewing colleagues' and classes' timetables

By default, the teacher can view the timetables of other teachers (useful for scheduling meetings), of classes (to identify a free slot for a field trip) and of classrooms. A strict setting (see [Visibility restrictions](#)) hides the timetables of **other teachers**; it only applies to these timetables between colleagues — access to class and classroom timetables remains open.

Syncing with a personal calendar

As on the student side, the share icon  **Sharing** in the timetable view opens the **Sharing** window, whose **iCal** tab provides the personal subscription URL to paste into Google Calendar, Apple Calendar or Outlook. See [iCal — subscription and dynamic link](#).

Convenient for teachers who want to see their lessons in the same calendar as their meetings and personal life.

See also

[Student portal](#)

[Teacher availability](#)

[iCal — subscription and dynamic link](#)

[Declaring an absence](#)

[Visibility restrictions](#)

10.3 Guest portal (public links)

Source: help/en/portal/guest-portal.md · id: portal.guest-portal · Plan: standard · Updated: 2026-05-15

The **guest portal** refers here to the viewing of a timetable by someone who has no Omniscol account at the school. The common case is the signed public link: a URL restricted to a precise scope, open in read-only mode.

Access modes

Depending on the need, choose between:

- **Signed public link** — a URL containing a secret token gives access to a specific timetable (a student, a class, a classroom, a teacher) without authentication. Convenient for sharing with a parent: a single link, no password to remember. See [Public share links](#).
- **Restricted Omniscol account** — if the person needs lasting, authenticated or multi-scope access, create an account with the appropriate role and rights instead. Do not present this case as an anonymous public link.

What a guest sees

The guest sees the timetable in **strictly read-only** mode:

- no changes possible,
- no absence declaration,
- no availability entry,
- no access to timetables beyond the scope included in the link.

The link opens the intended viewing screen, in read-only mode. For a **timetable**, the display is automatically stripped down — without the top banner or the application menu — which gives a page focused on the calendar alone. This presentation corresponds to the `raw=true` URL parameter: set by default on timetable links, optional on other screens, and one an experienced user can add by hand. It changes neither the authentication nor the scope of the link.

Some share links serve a targeted action rather than a viewing: the teacher availability entry link allows only the intended form, until its expiration date.

Use case: student's parent (K-12)

The typical scenario in middle and high school:

1. The administrator (or the student themselves) generates the share link for the student's timetable.
2. The link is sent to the parent by email.
3. The parent opens the link: they see their child's timetable.
4. From that same link, they can subscribe via iCal and have the timetable directly in Google Calendar / Apple Calendar.

No account to create, no password to manage.

Use case: permanent observer

If an external educational advisor needs to follow a program throughout the year, avoid the anonymous public link by default. Create an Omniscol account restricted to the necessary scope, or set up a signed share with a short expiration and controlled renewal.

Security

- A signed public link remains usable until its expiration date. It is invalidated if the password of the account it is attached to changes (or if that account is deactivated); an iCal link is also invalidated by rotating the school's access key.
- A restricted Omniscol account can be deactivated like any other user account.
- Access through a public link must not be treated as an authenticated internal user.

See also

[Public share links](#)

[Student portal](#)

[iCal — subscription and dynamic link](#)

[Share a timetable via a public link](#)

10.4 Public share links

Source: <help/en/portal/share-links.md> · id: [portal.share-links](#) · Audience: *admin* · Plan: *standard* · Updated: 2026-06-13

A **public share link** is a signed URL that gives access to a viewing screen — most often a timetable — in read-only mode without login. It is the simplest channel for distributing a calendar to someone who does not have (and will not have) an Omniscol account: a parent, an external partner, an administrative department.


Available formats

Depending on the context and the rights of the user who shares, Omniscol can generate several signed URLs:

- **Responsive web page** — a browsable, portal-style display, with week / day / list views. This is what someone sees when opening the link in a browser.
- **iCal feed (.ics)** — a dynamic subscription for Google Calendar, Apple Calendar, Outlook. See [iCal — subscription and dynamic link](#).
- **JSON (API)** — for developers and machine-to-machine integrations, when this format is allowed.

These formats are not a universal promise: the dialog shows the tabs actually generated for the current selection.

What can be shared?

Sharing is not limited to the timetable viewing screen: the same share icon  **Sharing** appears on most Omniscol screens.


In read-only mode, you can share:

- the **timetable of an entity** — student, class, teacher, classroom or subject: the most common case (parents, colleagues, adjunct teachers, technical departments);
- the **full timetable screen** of the **Timetable** module;
- the six **dashboards** (teachers, classrooms, resources, subjects, classes, students);
- the **absence tracking** screens (teachers, classes, students, staff);
- the **duty rosters** of the **Staffing** module (grid, assignments, schedule, attendance sheet);
- a timetable's **visualization**, **reorganization** and **layout** screens;
- the details screen of a teacher or a staff member.

In read-write mode — the only two cases — you share the **availability entry** of a teacher or a staff member: the dedicated link lets the person concerned fill in and confirm their availability, and edit their profile, until the expiration date, **without an Omniscol account or password**. The link is tied to their identity and is only offered on their own record. Availability for a class, a classroom, a group or a subject remains an internal editing screen: it is not collected through a share link.

Each signed URL carries the requested scope and an expiration date.

Creating a share link

From the **Timetable** module, select the timetable to share (student, class, classroom...) then click the share icon . The **Sharing** window that opens offers:

- **Expiration date** — the date until which the generated URLs remain valid.
- **Web, iCal and JSON tabs** — visible depending on the formats generated for the share.

You then copy the URL you want. The dialog can display a QR code for links intended for human viewing.

Limiting or invalidating a link

Public sharing keeps no register of issued links: Omniscol provides neither a consolidated list of links already sent, nor an access counter, nor link-by-link revocation. Invalidation therefore goes through the account behind the link (password, deactivation) and through the expiration date.

To limit the risk:

- choose a short expiration date;
- share only the necessary scope;
- for team-wide distribution, use an **identified service account** rather than a personal account;
- if a link spreads beyond the intended audience, change or reset the password of the account behind the link, or deactivate that account if necessary.

Changing the password invalidates the links carried by that account, even if the new password is identical in plain text, because the stored hash changes. Deactivating or deleting the account behind the links also cuts off access.

Security

- The URL contains a signed token that is hard to guess.
- The link exposes **only the screens written into its token**, not the rest of the account. For an **entity timetable** (and for any **iCal** feed), the filter is frozen in the token: the recipient stays confined to that entity — a parent holding the link to their child's timetable cannot reach another student's timetable. A **full timetable screen** share, on the other hand, reflects the **rights of the account that generated it**: for targeted external distribution, share the timetable of the specific entity rather than the whole screen.
- Validity is bounded by the expiration date.
- Validity also depends on the account behind the link: a password change, deactivation or deletion invalidates the associated links.
- Anyone who has the URL can view the shared scope until expiration or invalidation.

Embedding a timetable in a page (iframe)

Today, this is not possible on another site. To protect against clickjacking, Omniscol refuses to be displayed in a frame (`iframe`) hosted on any domain other than its own. If you try to embed a timetable in your learning platform, your intranet or your public website, the browser **refuses** to display the frame. This security lock is set by Omniscol and cannot be changed from the school account.

The only exception, rare in practice, is a page served on exactly the same domain as your Omniscol instance.


To distribute a timetable on an external site anyway, three simple solutions:

- **a link** to the share URL: the reader opens the timetable in a new tab, without an embedded frame;

- an **iCal subscription**, so the timetable appears in the person's calendar (Google Calendar, Outlook...) — see [iCal — subscription and dynamic link](#);
- a **display panel**, for a public screen on the premises — see [Display panel](#).

How-to

Creating a parent link with an expiration

1. **To share a class timetable with parents** without creating accounts for them: a signed public link, limited in time.
2. **Open the target timetable** in the **Timetable** module: select the class concerned. The share icon  **Sharing** is located in the title banner of the view.
3. **Sharing window**: choose the **expiration date**. For a parent share, use at most the end of the school period concerned.
4. **Copy the format you need**. Web for direct viewing, iCal for a calendar subscription, JSON only if an API integration is planned. Omniscol can also display a QR code for printed distribution.
5. **Send the link** through the intended channel: parents' meeting, newsletter, learning platform, intranet or targeted message.
6. **If the link circulates beyond the intended audience**, create a new, more limited link, then invalidate the links carried by the old account if necessary: password change, deactivation or deletion of the account behind them.

See also

[Guest portal \(parents, observers\)](#)

[iCal — subscription and dynamic link](#)

[Share a timetable via a public link](#)



11. Display panels

11.1 Setting up a display panel for a lobby or a corridor

Source: <help/en/panels/lobby-panel.md> · id: `panels.lobby-panel` · Audience: `admin` · Plan: `standard` · Updated: 2026-05-12

A **display panel** is a web page to load full screen in a browser, on a TV set, a monitor or a tablet, to broadcast the day's lessons and events to the school's public. It is the module to use for a welcome screen, a screen in a corridor, outside a laboratory or facing a lecture hall: the content depends only on what you filter, not on where the screen is placed. You can create as many panels as you want, each with its own filter and its own access URL.

Creating a new panel

In the **Timetable** module, the  **Display panel** button opens the configuration screen. The list on the left shows the existing panels; the (green) **+ Add** button creates a new empty panel. You can also **duplicate** an existing panel with **Duplicate** , which is faster if you manage a fleet of screens sharing the same branding.

Each panel carries a **Label**, to find it in the list, and a **Title**, displayed at the top of the public screen. The two can differ: `Main lobby – campus A` on the administration side, `Welcome to School X – campus A` on the screen side.

Filtering the displayed lessons

The main filter — the **Show** property — chooses what goes on screen. Four complementary axes:

- **Levels** — all the classes of one level at once (for example all the Grade 6 classes).
- **Classes** — explicit selection of one or more classes.
- **Sites** — all the rooms of one site (useful for a screen placed in the lobby of a specific site).
- **Classrooms** — explicit selection of one or more rooms (useful for a panel covering a building wing).

An empty filter = all lessons. That is the right default for a central lobby screen that has to show what is happening everywhere in the school.

A second list — **Hide** — lets you exclude specific lessons from the retained stream:

- **Subjects** to hide (for example not displaying physical education if it takes place far from the main building),
- **Lesson types** to hide (for example hiding exams that must not be announced to the public).

Filter and mask combine: everything that matches the filter is kept, then everything that matches the mask is removed.

Setting the displayed time range

The **Time slot** property controls how far ahead the panel shows lessons:

- **End of the day** (default) — scrolls through today's current and upcoming lessons (the current lesson stays displayed until its end time). Appropriate for a screen that serves all day long.
- **sliding window** — in 30 min steps up to **+ 2 h**, then by full hour up to **+ 12 h**. The panel only shows what starts within the window chosen from the current time. Useful for a very busy lobby screen where you want to avoid a wall of information: showing only the next two hours, for example.

Refreshing is automatic on the browser side, without manual reloading; as the current time moves forward, past lessons leave the window and the next ones enter it.

Adding temporary messages

The **Information** section lets you display one or more text messages on the panel, each with its own time range.

Typical cases:

- announcing an event (`Parent-teacher meeting in room B204 at 5:30 pm`),

- warning about a disruption (`Building work in the south wing, access through the rear entrance`),
- broadcasting a welcome message (`Welcome to the open day`).

Each message carries free text, a start time and an end time. Outside this range, the message disappears automatically — you do not have to come back and remove it afterwards.

Getting the panel's public URL

Once saved, each panel has a **unique URL** to open full screen on the target device. The URL contains a secret hash that grants access without signing in — this is what allows an unauthenticated TV set to display the content. As long as the hash is not shared publicly, the panel is not accessible.

Good practice: open this URL in a browser in **full-screen kiosk** mode (Chrome `--kiosk`, Firefox F11), preferably on a mini PC or a signage box attached to the screen. The browser handles refreshing and scrolling on its own.

☰ How-to

Creating a display panel

1. **A display panel is a web page** to load full screen on a TV set or a monitor to broadcast the day's lessons.
2. **Click** `+ Add` to create a new empty panel. You can also duplicate an existing panel to start from a base — a time saver if you manage a fleet of screens with the same branding.
3. **Fill in the label** (**Label** — on the administration side, to find your way in the list) and the **title** (displayed at the top of the public screen). For example: `Main lobby – campus A` on the administration side, `Welcome to School X` on the screen side.
4. **Choose the lesson filter** in the **Show** property: levels, classes, sites or classrooms. An empty filter = all lessons. For a central lobby panel, it is usually empty. For a screen outside a room, select that room.
5. **Optionally add temporary messages** with their time range (meeting, disruption, welcome). Outside the range, the message automatically disappears from the display.
6. **Save**. You get the panel's unique URL, to open full screen on the target device (Chrome `--kiosk`, Firefox F11). Refreshing is automatic from then on.

🔗 See also

[Display panel](#)

[Panel outside a classroom](#)

[Panel customization](#)

11.2 Setting up a display panel outside a classroom


Source: [help/en/panels/room-panel.md](#) · id: `panels.room-panel` · Audience: `admin` · Plan: `standard` · Updated: `2026-06-13`

A **panel outside a classroom** is a special case of the display panel: the configuration is the same as for a lobby panel, but the filter targets a **single room** instead of a site or a level. The screen displayed next to a door shows only the occupancy of the corresponding room, which answers the question people in the corridor ask themselves: *"what is happening in this room now and next?"*

Typical cases:

- a lecture hall door,
- the entrance of a laboratory or a computer room,
- a multi-purpose room shared by several programs,
- a studio or a workshop (music, design, video).

Configuration

The **Timetable** module offers the same  **Display panel** button as for a lobby panel. The difference lies in the filter: in the **Show** section, choose **Classrooms** and select the single corresponding room.

The other settings are identical to the lobby panel:

- **Label** on the administration side (for example `Door B204`), **title** on the screen side (for example `Room B204 – Experimental sciences`).
- **Time slot**: prefer **+ 2 h** or **+ 4 h** rather than the whole day — for a room, the user mostly wants to know the current lesson and the next one.
- **Font size**: **Large** is often appropriate for a screen read from 1-2 meters away in a corridor, smaller if the screen is close.
- **Messages**: handy for announcing a one-off booking (`Room booked for the class council at 2 pm`) or an unavailability (`Room under maintenance until 3 pm`).

Several rooms on the same screen


If a screen covers several neighboring rooms (a building wing, the rooms of one laboratory), you can select several rooms in the filter. The lessons of all the selected rooms are then **merged into a single list sorted by time**; if it overflows the screen, the panel **paginates** it (page scrolling). It does not alternate room by room. For a truly separate display per room, create one panel per room — each has its own URL. A single mini PC can display several panels by rotating several full-screen tabs, via the browser's or the system's kiosk feature (this is not an Omniscol setting).

URL and deployment

Each panel has its own unique URL, with no sign-in, to open in full-screen kiosk mode on the device installed next to the door. Refreshing is automatic, by periodic polling: the display re-filters itself against the current time roughly every minute, and the data is reloaded every five minutes — no manual action, but the update is not instantaneous.

How-to

Creating a panel outside a classroom

1. **A panel outside a classroom** displays that room's occupancy over the next few hours. Difference from the lobby panel: single-room filter, short time range, font size suited to reading from the corridor.
2. Click  **+ Add** in the **Timetable** module to create a new panel (or duplicate an existing lobby panel and adjust it).
3. **Fill in the label and the title.** Label on the administration side (`Door B204`), title on the screen side (`Room B204 – Experimental sciences`). The title is what people in the corridor see.
4. **Filter: section Show** → **Classrooms**, select **the corresponding room**. For a screen covering a wing (several neighboring rooms), select them all — their lessons are merged into a single list sorted by time, paginated if needed.
5. **Settings specific to a classroom door:**
 - **Time slot**: prefer **+ 2 h** or **+ 4 h** rather than the entire day — the user wants to know the current lesson and the next one.
 - **Font size**: `Large` for a screen read from 1-2 m away.
 - **Messages**: useful for announcing a one-off booking or an unavailability.
6. **Save.** Get the panel's unique URL, to open in full-screen kiosk mode on the device installed next to the door. Automatic refresh by periodic polling (re-filtering roughly every minute, data reload every five minutes).

See also


[Display panel](#)[Display panel](#)[Panel customization](#)

11.3 Visual customization of display panels

Source: <help/en/panels/customization.md> · id: `panels.customization` · Audience: `admin` · Plan: `standard` · Updated: 2026-05-12

Omniscol display panels are designed to blend visually into your branding. Two levels of customization are available: (1) the settings **in the interface** (sufficient for most cases), (2) the **complete replacement** of the web page that renders the panel (for deployments with a strong visual identity).

Settings in the interface

In the **Timetable** module, the  **Display panel** button opens the panel configuration screen. This screen exposes the following rendering options, editable without touching any code:

- **Font size** — small, medium or large. The display uses units relative to the window size (`vmin`), so readability stays consistent whatever the screen format (`16:9`, `4:3`, `portrait`).
- **Colors** — background color, text color, colors of the even and odd rows of the lesson grid. All hexadecimal codes are accepted; you can copy the palette of your branding.
- **Header banner** — an ordered combination of: **title**, **logo**, **today's date**. You choose what appears in the header and in what order.
- **Grid columns** — selection and order among: **classroom**, **classes**, **teachers**, **subject**, **times**. Hiding a column (for example the teachers on a public panel at the entrance) or reordering it is immediate.
- **Teacher name format** — first name-last name, last name-first name, title + last name (Ms Durand), title + first name. The `Mr` and `Ms` titles automatically follow the interface language; what you set per panel is the **format**.
- **Logo** — the school logo uploaded in the school configuration can be displayed in the header banner.

These settings are saved in the panel configuration. Note: unlike lessons and messages, which refresh on their own, these **visual** options (colors, size, columns, banner, name format) are embedded in the page when it loads. After saving them, **reload the tabs** already open on the screens (F5 or a restart of the kiosk page) so they pick up the new rendering.

Complete template replacement (MIT license)

The template that renders the display page is explicitly under the **MIT license** and can be copied then adapted to customize a school's panel.

This opens up several scenarios for schools that want to go beyond the settings:

- **Complete visual redesign** — rewrite the template's HTML/CSS to fit a strong branding (proprietary font, specific animations, non-standard layout).
- **Integration into an existing signage system** — embed the content of an Omniscol panel in a global display page that multiplexes lessons, weather, school news, videos...
- **Specific business views** — create a display, for example, for supervisors (who has to do what), for a technical department (room occupancy with equipment needs), and so on.

To do this, two complementary approaches:

1. **Forking the template** — if you self-host Omniscol, duplicate `panel.dot`, modify it, and deploy your version. The MIT license explicitly allows modifications, redistributions and commercial uses.
2. **The panel's JSON API** — if you want to keep Omniscol as the data source but render the display on a third-party side, the panel URL also exposes a JSON variant (`GET /panels/:accesshash/lessons`). It returns the list of lessons filtered according to the panel configuration, ready to be consumed by a third-party signage system (Yodeck, Xibo, ScreenCloud, an in-house dashboard). This URL opens with the panel's access key,

without any additional token. For broader system-to-system integrations, the Omniscol REST API is described in [Omniscol API](#).


Practical recommendations

A few points that come up often in deployments:

- **Contrast** — a dark background + light text is easier on the eyes for a screen running 12 hours a day than a white background.
- **"Large" font size** as soon as the screen is read from more than 2 m away (typical of an entrance hall). Keep "Medium" for classroom door screens, viewed from closer up.
- **Logo in the header banner** — always useful for quick identification when several panels coexist on a large campus.
- **Date in the header banner** — useful in a lobby (visitors check the date), less useful outside a classroom (the context is obvious).
- **Hiding the teachers column** on public panels if your school has internal rules about displaying names.

☰ How-to

Matching your school's branding on a panel

1. **Two levels of customization:** (1) settings in the interface (sufficient 90% of the time), (2) complete replacement of the MIT template for strong branding.
2. **Open the panel configuration** in the **Timetable** module, via the  **Display panel** button. The screen shows the rendering options: colors, size, header banner, columns, name format.
3. **Align the colors with your branding:** background color, text color, colors of the even/odd rows. Hexadecimal codes accepted. For a screen running 12 hours a day, prefer a **dark background + light text** (easier on the eyes).
4. **Header banner:** choose the order of **logo, title, date**. The logo is useful as soon as there are several panels on a campus. The date is useful in a lobby, dispensable outside a classroom.
5. **Grid columns:** select and reorder among classroom, classes, teachers, subject, times. For a panel that is **public at the entrance** where your internal rules forbid displaying names, hide the **teachers** column.

Name format (if kept): first name-last name, last name-first name, title+last name, title+first name.

6. **Font size:** large as soon as the screen is read from more than 2 m away (lobby), medium for a classroom door (viewed up close). Relative units in `vw` — readability stays consistent whatever the format of the screen.
7. **Save, then reload the screens.** These visual settings are embedded in the page when it loads: reload (F5) the tabs already open on the panels so they pick up the new rendering.

To go further (complete redesign, third-party signage integration Yodeck/Xibo/ScreenCloud): the `panel.dot` template is under the **MIT license**, forking allowed. The JSON variant of the panel URL (`GET /panels/:accesshash/lessons`) exposes the data ready to be consumed by an external system, with the panel's access key. For the broader REST API, see [Omniscol API](#).

🔗 See also

[Display panel](#)

[Panel outside a classroom](#)

[Display panel](#)

[Omniscol API](#)

12. Migrating from another program

12.1 Migration from another program — Overview

Source: <help/en/migration/overview.md> · id: [migration.overview](#) · Audience: [admin](#) · Updated: 2026-05-10

Are you coming to Omniscol from another timetable management program? Several paths exist depending on the source program and how rich the available export is.

Available approaches

1. Native format importers

For two programs, Omniscol reads the native export file directly, without going through an intermediate spreadsheet: you select the file or files from the Import/Export screen (the UnDeuxTEMPS import, made of several `.DBF` files, sits in the French formats section).

Source program	Format read directly	Dedicated page
ASC Timetables	aSc XML file	From aSc Timetables
UnDeuxTEMPS	UDT files (<code>.DBF</code>)	migration.from-undeuxtemps

For the other programs (Hyperplanning, EDT / Pronote, ADE), the migration goes through the spreadsheet **mass import** described in point 2: you export your data from the source program, you shape it in a spreadsheet, then you import it. Each page below details the correspondences and the pitfalls specific to the program concerned:

- [From Hyperplanning \(Index Education\)](#)
- [From EDT / PRONOTE \(Index Education\)](#)
- [From ADE / ADE Campus](#)

Aurion and Auriga are **ERPs** (school administrative management), not scheduling programs. Omniscol interfaces with them but does not replace them. See [integrations.aurion](#) and [integrations.auriga](#) for the integration modes.

2. Generic spreadsheet import

If your source program has no native importer, or if you prefer to control the formatting yourself: extract your data into a spreadsheet (Excel, Google Sheets, Numbers, Calc) and use the Omniscol **mass import**. See [Mass import of courses from a spreadsheet](#).

3. Direct API

For an **automated** migration (for example if you want to synchronize during a transition period between two programs), you can write a script that pushes the data through the Omniscol API. See [Omniscol API](#).

4. Manual re-entry

For small institutions or partial migrations, manual entry remains possible. Allow a few hours to a few days depending on the size.

Recommended migration strategy

1. **Map your data** — before anything else, take stock of what you want to transfer: teachers, students, rooms, subjects, class structure, current timetable, availabilities, absences.
2. **Decide on the scope** — often, not everything is transferred. Students can stay in the SIS and be synchronized afterwards. Historical timetables can remain archived separately.
3. **Run a partial test import** — on a subset (one class, one cohort) to validate the quality of the transfer.

4. **Set the timeline** — often, the switch happens at the **start of the next school year** rather than mid-year, to avoid managing continuity.
5. **Train the users** — Omniscol has conventions that differ from competing programs. Plan for training time, especially on the domain concepts ([Class division](#), [Group alignment](#), [Group of groups](#)).

Terminology specifics by source program

Competitor program term	Omniscol equivalent
"resource" in the Hyperplanning sense (teacher, room, class...)	Several entities depending on the case
"cours" in the EDT/Pronote sense (the unit placed on the timetable)	Lesson / Session
"cours" in the pedagogical sense (a subject taught to a class)	Course
"building"	Tag or building on the Classroom
"formation" in higher education	Class or a set of classes
"UE / EC / ECUE"	Subject with a naming convention

See also

[Preparing your data for a mass import](#)

[Mass import of courses from a spreadsheet](#)

[Omniscol API](#)

[FAQ — higher education use cases](#)

[FAQ — special cases and advanced configurations](#)

12.2 Migrating from Hyperplanning (Index Education)

Source: [help/en/migration/from-hyperplanning.md](#) · id: [migration.from-hyperplanning](#) · Audience: admin · Updated: 2026-06-30

Hyperplanning (Index Education) is the most widespread timetable management program in French higher education. Migrating to Omniscol means bringing over your existing data (users, structures, current timetables) and adapting it to Omniscol concepts. This page summarizes the correspondences and the known pitfalls.

What maps directly

Hyperplanning	Omniscol
Promotion (common core)	Class
Partition of a promotion	Disjoint groups; a division if those groups share the same time slot (simultaneous tutorials/labs)
Tutorial/lab subgroup (from a partition)	Group in a division
Option	Group
Regroupement (grouping)	Group of groups (or an alignment across different classes/promotions)
Course	Course (and its lessons)
Subject / module (UE, EC)	Subject (often customized)

Hyperplanning	Omniscol
Calendar / periods	Calendar mode (dated timetable, Premium); date windows to include/exclude periods
Room	Room (specialization to carry over)
External instructor / adjunct	Teacher marked as "external" (Premium option)

What needs attention

- **Alignments:** in Hyperplanning, having tutorial or option groups from several promotions attend the same course at the same time is done through a **shared course** (often built with a grouping). When this shared course forces the same time slot, the same room and the same teacher onto groups from different classes, that is the Omniscol **alignment**. See [Group alignments](#).
- **Groupings:** a Hyperplanning grouping brings several groups together to attend the same course; in Omniscol this is a **group of groups** (available on all plans). See [Groups of groups](#). Depending on the case, a grouping across different classes or promotions may instead be the **alignment** described above.
- **Associated lessons** (alternating half-groups): Omniscol handles this natively through **associated lessons**. See [Complex lessons](#).
- **Teaching service:** pedagogical continuity (the same resources from one lesson to the next) is reproduced by keeping the same teacher and the same room on the lessons of a course — there is no dedicated entity.
- **Calendar-based courses:** if you use Hyperplanning in calendar mode for modules with precise dates, Omniscol covers the same needs with its [calendar mode](#), included in the Premium plan.
- **Hyperplanning vs Aurion:** if your institution uses Hyperplanning **and** Aurion, the migration only concerns the scheduling side; Aurion keeps feeding the administrative structure. See [Synchronization with external systems](#).

☰ How-to

1. **Retrieve the data** from Hyperplanning into a spreadsheet — lists of instructors, students, rooms and courses. Depending on your version, this is done by copy-pasting a list or through its export; refer to the Hyperplanning documentation.
2. **Prepare the Omniscol files** in the import format: users CSV, courses CSV. See [Preparing your data for a mass import](#).
3. **Create a sandbox Omniscol account** to test the import risk-free.
4. **Import in several passes:**
 - users (teachers, students),
 - classes and groups,
 - subjects,
 - courses via the [spreadsheet mass import](#).
5. **Rebuild the complex courses** (alignments, associated, alternating) by hand if the import flattened them.
6. **Check the diagnostic** and fix the detected inconsistencies.
7. **Run a test generation** to validate feasibility.
8. **Snapshot** before switching over to the production account.

Migrating users smoothly

Good practice: import users into Omniscol as **inactive** first. You validate the data, you switch the configuration over (SSO if applicable), and you activate the accounts in a single pass when everything is ready. This prevents users from receiving a premature invitation.

What does not migrate

- **Hyperplanning history** beyond the current timetables — to keep the history, keep Hyperplanning as a read-only archive rather than migrating everything.
- **Hyperplanning visual customizations** — rebuild them in Omniscol according to your visual identity. See [Panel customization](#).

☰ How-to

Hyperplanning → Omniscol migration

1. **Migrating from Hyperplanning:** the 8-pass sequence follows the order export → prepare → sandbox → multi-pass import → fixing complex courses → diagnostic → test generation → switchover.
2. **Pass 1 — Retrieval from Hyperplanning:** get the lists (instructors, students, rooms, courses) out into a spreadsheet. Depending on the version, by copy-pasting a list or through its export — see the Hyperplanning documentation. The cleaner the source side, the faster the rest goes.
3. **Passes 2-3 — Prepare + sandbox:** adapt the columns to the Omniscol template (see [Preparing your data](#)). Create a **sandbox Omniscol account** to test the import with no risk to production.
4. **Pass 4 — Multi-pass import** in this order: **users** → **classes and groups** → **subjects** → **courses** via the [mass import](#). The order matters: courses reference the classes and teachers, which must already exist.
5. **Pass 5 — Complex courses:** alignments (courses shared by several promotions on the Hyperplanning side → Omniscol alignments), associated lessons (alternating half-groups), A/B alternating weeks. The import often flattens them — rebuild them by hand via the complex icons on the course card.
6. **Passes 6-7 — Diagnostic + test generation:** let the diagnostic run, fix the inconsistencies. Run an **automatic generation** to check overall feasibility before migrating to production.
7. **Pass 8 — Switchover:** create a **snapshot** before switching. Activate the user accounts (imported as inactive until now). Publish. Keep Hyperplanning as a read-only archive — the migration does not carry over history beyond the current timetables.

If you also have **Aurion**: the migration only concerns the scheduling side, Aurion remains the administrative source. See [Synchronization with external systems](#).

🔗 See also

[Migration overview](#)

[Preparing your data for a mass import](#)

[Mass import of courses from a spreadsheet](#)

[Synchronization with external systems](#)

12.3 Migrating from EDT / PRONOTE (Index Education)

Source: [help/en/migration/from-edt.md](#) · id: [migration.from-edt](#) · Audience: [admin](#) · Updated: 2026-06-30

EDT and **PRONOTE** (Index Education) are the most widespread tools in French secondary education — lower and upper secondary schools, public and private alike. The pair covers the timetable (EDT) and school life — grades, report cards, homework diary (PRONOTE). Migrating to Omniscol primarily concerns the timetable side; for advanced school life (grades, report cards, parent communications), PRONOTE can stay in parallel.

The most direct path: the STSweb export

EDT builds its structures from **STSweb**: the **services** (a teacher delivers a set number of hours of a subject in a class) come from STSweb and go back up to it at the end of preparation. That is good news for the migration,

because **Omniscol imports an STSweb export directly** (STS file) from the [Import and export](#) screen.

This import rebuilds the structural data in one go: institution (UAI code), school year, **subjects, levels** (MEF) and their statutory time grid, **classes** (divisions), **groups, teachers**, and the **services** — which become Omniscol **courses**. The detail of the supported French formats is described on [admin.french-formats](#).

The spreadsheet remains useful for what STSweb does not carry (rooms, special groupings, fine-grained wishes): see the mass import below.

What is carried over differently

- **Rooms, remote sites**: STSweb does not describe rooms in detail. Bring them in with the spreadsheet or re-enter them, then rebuild the sites. For a set of interchangeable rooms (the equivalent of an EDT "room group"), use a shared **specialisation** that the solver will respect. See [Classroom specialisations](#).
- **Teacher wishes and unavailability**: the granularity differs from EDT's; plan for a re-entry (sending the entry links before the school year starts) or an approximate import to adjust.
- **Holidays and school calendar**: they are defined in the school-year settings (country template or manual entry), not through the export.

What stays in PRONOTE

Omniscol does not take over (and you can leave them in PRONOTE):

- grades, assessments, report cards, competencies;
- homework diary, assignments;
- school life in the broad sense (parent communications, surveys);
- absences can stay in PRONOTE, or be tracked in Omniscol if you prefer to move them there.

Main correspondences

EDT / STSweb	Omniscol
Class (division)	Class
Class part	Group; a division for the mutual-exclusion relation on the same time slot
Group (needs-based groups, specialty tracks, electives, languages)	Group (within a class); an alignment or a group of groups if across classes
Service (STSweb)	Course (its lessons follow from it at placement time)
Subject	Subject
Room / room group	Room; interchangeable pool = room left unset (auto-assignment) or shared specialisation
Remote site	Site (with travel time)
Teacher wish / unavailability	Availability (wishes); the "impossible" level (black) for an unavailability
Q1 / Q2 alternation (fortnights)	Alternating weeks A / B
Study hall / school library (CDI)	Staff duty (supervision) + a "study" course for the group with no lesson

☰ How-to

1. **Retrieve the STSweb export** (STS file) for the institution — it is the official source of structures and services.
2. **Import the STS file** into a sandbox Omniscol account, from [Import and export](#). The structure (institution, year, subjects, levels, classes, groups, teachers, services → courses) is rebuilt in one go. See [admin.french-formats](#).
3. **Complete with the spreadsheet** what STSweb does not carry (rooms, specialisations, special groupings) through the mass import. See [Mass import of courses from a spreadsheet](#) and [Preparing your data for a mass import](#).
4. **Check the week alternation**: EDT's Q1/Q2 convention must map to Omniscol's **A/B** alternating weeks.
5. **Carry over teacher wishes and unavailability** — re-entry or approximate import to adjust.
6. **Run the Omniscol diagnostic**, then a test generation.
7. **Snapshot** before switching over to the production account.

EDT + Omniscol coexistence during the transition

During a transition period (often one term), it is common to keep PRONOTE for school life and to move the timetable to Omniscol. PRONOTE normally receives its timetable from EDT; if you want it to keep displaying timetables managed elsewhere, check with Index Education what PRONOTE can consume — do not assume an external feed will be picked up as-is.

Typical school migration

Standard case of a lower or upper secondary school moving from EDT to Omniscol:

- STSweb export in June,
- Omniscol import and tests over the summer,
- switchover at the start of the school year in September,
- PRONOTE kept for school life if the institution wishes.

☰ How-to

Typical school migration, June → September

1. **The typical case of an institution** moving from EDT/PRONOTE to Omniscol: export in June, tests over the summer, switchover at the start of the school year. PRONOTE kept for school life if desired.
2. **June — STSweb export**: retrieve the institution's STS file (the official source of structures and services). Failing that, export the EDT lists (courses, services, classes) to a spreadsheet — depending on your version, through list exports or copy-paste; see the EDT documentation.
3. **July — Omniscol sandbox account**: import the STS file from [Import and export](#) (see [admin.french-formats](#)). The structure is rebuilt in one go. You test with no pressure: if something is not right, you iterate calmly.
4. **August — Complete with the spreadsheet** what STSweb does not carry (rooms, room specialisations, groupings) through the mass import (see [Mass import of courses from a spreadsheet](#)). Align the **week alternation** (EDT's Q1/Q2 → Omniscol's A/B).
5. **August — Teacher wishes and availability**: the granularity differs. Plan either a **re-entry** (sending the entry links before the school year starts) or an **approximate import** to adjust afterwards.
6. **Late August — Diagnostic** in Omniscol to spot the inconsistencies (missing subjects, conflicts, availability not carried over). Run a **test generation** on the first weeks of term to validate feasibility.
7. **September — Switchover**: snapshot first, then publish the timetables. **PRONOTE kept** for school life if you wish.

📌 See also

[Overview](#)[From Hyperplanning \(Index Education\)](#)[admin.french-formats](#)[Mass import of courses from a spreadsheet](#)

12.4 Migrating from aSc Timetables

Source: [help/en/migration/from-asc.md](#) · id: [migration.from-asc](#) · Audience: [admin](#) · Updated: 2026-06-30

aSc Timetables (Slovakia) is a long-established and very widespread timetable generation program, notably in international schools and several Central European countries. Good news: Omniscol reads and writes the native aSc Timetables file directly, with no intermediate spreadsheet and no manual remapping of structures.

One file, both ways

aSc Timetables saves a timetable in an **XML** file ("aSc Timetables 2012" format). Omniscol can:

- **read** this file to rebuild a complete timetable (classes, groups, teachers, subjects, rooms, courses);
- **write back** a file of the same type from an Omniscol timetable, to reopen it in aSc or Edupage (the school-life solution from the same publisher).

Everything happens from the [Import and export](#) screen, with no technical manipulation: to export, the [Export data in aSc \(.xml\) format](#) button; to import, you simply select the XML file produced by aSc. The round trip is **reversible**: exporting then re-importing yields the same structure on the Omniscol side, apart from naming conventions.

What carries over

Picked up automatically when reading the file:

- **Classes** and their **groups** (including divisions for lab work).
- **Subjects** with their codes.
- **Rooms** with their capacity.
- **Teachers** (last name, first name, contact details).
- **Courses** (subject, volume, teacher) and their placed **lessons** (day, time slot, room).
- **Breaks and pauses** during the day.
- **Alternating weeks** (A/B) if they are configured in aSc.

What needs a look after import

- **Complex constraints** specific to aSc (sequence constraints, conditional rooms, very specific anti-chaining): Omniscol converts them as well as it can, but some rules are expressed differently and deserve a review. Conditional rooms are better covered by [Classroom specialisations](#); the others by [Teacher availability](#) and [Time constraints \(general system\)](#).
- **Courses shared by several classes** (one subject common to several classes): converted into [Groups of groups](#). Check that the result matches your intention.
- **Renaming and round trips**: Omniscol does not keep aSc's internal identifiers; when it writes an aSc file back, it rebuilds the correspondences from the entities, in particular from their **name**. In practice, if you plan to re-export to aSc, avoid renaming classes, teachers and subjects in the meantime, otherwise the correspondences drift.

How-to

1. **In aSc**: save your timetable in the **XML** format (2012 version, or the closest one your aSc offers).
2. **Create a sandbox Omniscol account** to test risk-free.
3. **Import the file** from the [Import and export](#) screen: select the aSc XML file, then confirm.
4. **Read the Omniscol diagnostic** to spot the inconsistencies (courses without a room, availability not carried over, etc.).
5. **Adjust the complex constraints** that could not be converted automatically.
6. **Run a test generation** to check that the constraints produce a result matching what you had in aSc.
7. **Take a snapshot** before switching over to the production account.

Continuing to feed aSc in parallel

If you keep aSc for the duration of a transition, the reverse export lets you keep feeding it from Omniscol — useful, for example, when partner institutions still consult aSc. The generated file reopens in aSc Desktop (2012 version or later).

What needs attention

- **Accented characters**: if some accents come out wrong after a round trip, it is a file-encoding matter. The Omniscol export uses the encoding expected by aSc Desktop by default; simply check that an accented name survives the round trip.
- **Different generation engines**: aSc and Omniscol do not have the same solver profile. A timetable that generated easily in aSc may need a few adjustments in Omniscol, and vice versa.
- **aSc versions**: the import targets the 2012 format. Older versions may not be directly compatible — in that case, save in the 2012 format from aSc Desktop first.

☰ How-to

aSc ↔ Omniscol XML round trip

1. **Omniscol reads and writes the native aSc Timetables 2012 file:** a direct round trip, with no intermediate spreadsheet.
2. **On the aSc side:** save the timetable in the **XML 2012** format (or the closest version) and keep the file on your computer.
3. **On the Omniscol sandbox side:** open [Import and export](#) and select the aSc XML file. The import rebuilds classes, groups, subjects, rooms, teachers, courses, alternating weeks and breaks.
4. **Check the diagnostic:** courses without a room, availability not carried over, multi-class courses converted into [Groups of groups](#). Reminder: for a clean round trip, do not rename classes, teachers and subjects if you plan to re-export — Omniscol relies on their name to rebuild the correspondences.
5. **Adjust the complex constraints** specific to aSc (sequences, very specific anti-chaining): the conversion does its best, some rules are expressed differently in Omniscol. See [Teacher availability](#).
6. **Run a test generation** in Omniscol to check that the result matches. ⚠ aSc and Omniscol do not have the same solver profile; a timetable that is easy in aSc may need adjustments in Omniscol, and conversely.
7. **Reverse export** (if you keep aSc in parallel): the [Export data in aSc \(.xml\) format](#) button in [Import and export](#). The file reopens in aSc Desktop 2012+; the round trip preserves the structure (minor losses on naming conventions only).

🔗 See also

[Overview](#)[Import and export](#)[Teacher availability](#)[Groups of groups](#)

12.5 Migrating from ADE / ADE Campus

Source: [help/en/migration/from-ade.md](#) · id: [migration.from-ade](#) · Audience: [admin](#) · Updated: 2026-06-30

ADE (published by Adesoft), often used in its web version **ADE Campus**, is widespread in higher education, particularly universities and grandes écoles. ADE is **activity-centric** (the "course session") and frequently synchronizes with the institution's information system (Apogée, Pegase, Aurion, UNIT4...). Moving to Omniscol primarily concerns scheduling; the synchronization with your information system still has to be reconfigured on the Omniscol side.

Scope

What carries over:

- **programs** (PGE, CPGE, BBA...) and their groups (utility groups, half-promotions);
- **instructors** (permanent, non-permanent, adjunct);
- **activities** (course sessions) with their type, duration and number of repetitions;
- **rooms and equipment** (by building and floor, with their capacity);
- **availability and unavailability** of resources.

ADE and Omniscol have broadly compatible models, but the naming conventions differ. Plan for an initial mapping.

Correspondences

ADE (ADE Campus)	Omniscol
Program (PGE, CPGE, BBA...)	Several classes (one promotion /year = one class)
Utility group / half-promotion	Group; a division if two half-promotions attend different lessons on the same time slot
Instructor	Teacher
Activity (course session)	Course (and its lessons)
Activity typology (lecture, tutorial, lab...)	Course type
Modality (in-person, remote...)	Modality (Premium attribute)
Cap / enrolled	Theoretical headcount — no separate cap (the seat limit comes from the room capacity)
Room / equipment (building, floor, capacity)	Room (site, capacity); only movable equipment becomes a resource
Availability / unavailability (color scale)	Availability (wishes): 4 levels — impossible / undesirable / preferred / neutral
Activity association	Depending on the link: alignment, concatenation or associated lessons
CURSUS / UNIT4 code	External identifier (trace of the information-system synchronization)

☰ How-to

1. **Retrieve the data** from ADE — programs, instructors, rooms, activities. ADE presents its data as **configurable lists** (Listing views, choice of columns) that you can print or export to a spreadsheet via the **Print** menu.
2. **Prepare the Omniscol files** from these exports (see [Preparing your data for a mass import](#)).
3. **Import from a spreadsheet** into a test timetable (see [Mass import of courses from a spreadsheet](#)).
4. **Rebuild the links**: alignments (the activity associations on the ADE side), alternations, half-promotion divisions — ADE and Omniscol do not use exactly the same primitives.
5. **Run a test generation**.

What needs attention

- **ADE schedules on precise dates**: the **Placement** view (week, day, start, end) and the weeks bar make ADE a calendar-type tool. To keep this behavior, use the Omniscol calendar mode, included in the Premium plan (see [Calendar mode](#)).
- **Half-promotions and utility groups**: a program is often split into utility groups and half-promotions. On the Omniscol side, half-promotions correspond to a **division** of the class; think through the group granularity from the start.
- **Imposed or "free choice" rooms**: in ADE, a room can be imposed on an activity or left to the engine's choice. On the Omniscol side, an imposed room is kept as-is by the algorithm; for a "free choice" room, leave the lesson without a specific room — the algorithm picks one on the site, respecting the capacity and any required **specialisation** (a set of interchangeable rooms is modeled with a shared specialisation). See [Sites, classrooms, resources](#) and [Automatic classroom assignment](#).
- **Graded availability**: ADE handles availability on several levels (from green to red, plus a dynamic layer). Omniscol uses 4-level **availability (wishes)** (impossible, undesirable, preferred, neutral) — carry over the extremes first (impossible and preferred); the intermediate levels can be reworked afterwards.

- **Pedagogical lead:** ADE attaches a pedagogical lead to each activity. Omniscol does not carry this role at course level; keep the information through a naming convention or a memo if you find it useful.
- **iCal feeds:** ADE often publishes timetables as iCal feeds. The Omniscol import does not read iCal; an iCal export must first be converted into a spreadsheet table.
- **Synchronization with the information system:** if ADE was synchronized with your information system (Apogée, Pegase, Aurion, UNIT4...), you will have to reconfigure that synchronization on the Omniscol side — often via the API or the synchronization with external systems. ADE's **CURSUS / UNIT4 code** field is its trace on the source side. See [Synchronization with external systems](#).

☰ How-to

ADE → Omniscol migration (higher education)

1. **ADE (Adesoft), often in its ADE Campus version**, is widespread in higher education. The migration primarily concerns scheduling; the synchronization with the information system (Apogée, Pegase, Aurion, UNIT4) still has to be reconfigured on the Omniscol side.
2. **Retrieving the ADE data:** programs, instructors, rooms, activities (with type, duration, repetitions, cap). ADE presents this data as configurable lists (Listing views) that you print or export via the **Print** menu. An iCal feed must first be converted into a spreadsheet table (the import does not read iCal). If you schedule on precise dates, preserve that aspect too.
3. **Initial mapping:** ADE and Omniscol have compatible models but different naming conventions. Prepare the files according to the Omniscol template (see [Preparing your data for a mass import](#)). Think through the granularity of the **student groups** (half-promotions, utility groups) from the start — that is the tricky point in higher education.
4. **Import from a spreadsheet** into a test timetable: instructors, programs, subjects, rooms, then activities (see [Mass import of courses from a spreadsheet](#)). The course spreadsheet has no external-identifier column; for future reconciliation with your information system, plan on the external synchronization instead (see [Synchronization with external systems](#)).
5. **Rebuilding the links:** multi-program alignments (the activity **associations** on the ADE side), alternations, half-promotion divisions. Also revisit the rooms left as "free choice" by letting the algorithm assign them (through a shared specialisation if needed). ADE and Omniscol do not use exactly the same primitives — count on a manual finishing pass.

If ADE scheduled on precise dates, use the Omniscol **calendar mode** (included in Premium) to preserve that behavior (see [Calendar mode](#)).

6. **Test generation** in Omniscol to validate feasibility. Diagnostic, fixes, iteration. The more you test before the switchover, the less you fix afterwards.
7. **Reconfiguring the information system:** if ADE was synchronized with Apogée, Pegase, Aurion or UNIT4, you will have to reconfigure that synchronization on the Omniscol side via the API (see [Omniscol API](#)) or the synchronization with external systems (see [Synchronization with external systems](#)). For Aurion specifically, see [integrations.aurion](#) — 3 possible modes.

🔗 See also

[Overview](#)

[Omniscol API](#)

[Synchronization with external systems](#)

[Calendar mode](#)

12.6 Migrating from a homegrown Excel spreadsheet

Source: [help/en/migration/from-spreadsheet.md](#) · id: [migration.from-spreadsheet](#) · Audience: [admin](#) · Updated: 2026-05-12

Many institutions — especially small structures, private schools, short training programs — still manage their timetable in a homegrown Excel spreadsheet. Migrating to Omniscol is in that case particularly simple: no

proprietary format to decode, just a little discipline to structure the data.

Before starting: sort things out

A timetable spreadsheet often piles up several "tabs" of different kinds: the list of teachers, the list of classes, the list of subjects, the timetable grid, miscellaneous notes. Before the import, clarify:

- the **list of teachers** (a clean tab),
- the **list of students** (a clean tab, if it exists),
- the **list of classes** and their headcounts,
- the **list of rooms**,
- the **courses**: for each row, class, subject, teacher, room, day, time, duration.

If your spreadsheet does not separate this information, do it now — it is needed for the import anyway.

Expected import format

The import happens by **copy-paste** from your spreadsheet (Excel, Google Sheets, Numbers, Calc...), into a documented column template. Columns are recognized by their **position** (you rearrange the template so it matches your file), not by the header name. See [Preparing your data for a mass import](#) for the details (column order, date formats, subject codes, etc.).

For courses, the [Mass import of courses from a spreadsheet](#) is the dedicated tool: you **copy-paste** the rows of your spreadsheet directly into an editable area, with a one row = one course structure.

How-to

1. **Clean up** the source spreadsheet: remove unnecessary columns, harmonize the labels (a teacher must not appear under three different spellings).
2. **Prepare one tab per entity type**: teachers, students, classes, rooms, courses — you will paste each tab in turn.
3. **Omniscol test account** to validate the import risk-free.
4. **Successive imports**, starting with the reference data (teachers, rooms, subjects) before the courses that refer to them.
5. **Check the Omniscol diagnostic** to detect what was not interpreted correctly.
6. **Run a test generation** (even if you keep the manual placement from your spreadsheet) to validate feasibility.

Benefits of migrating

- **A single tool** brings together the spreadsheet, the timetable PDFs and the updates that circulated by e-mail.
- **Real-time updates**: everyone sees the latest version as soon as it changes.
- **Automatic distribution**: iCal, display panels, student and teacher portals.
- **Diagnostic**: Omniscol detects the conflicts your spreadsheet lets through.

Keep the spreadsheet as a read-only archive

No need to delete your historical spreadsheet. Keep it read-only as an archive (for example in a shared folder) — it can help if you want to look up an old configuration. Omniscol then becomes the operational source of truth; the spreadsheet remains a historical record.

☰ How-to

Migrating a homegrown Excel timetable

1. **The simplest case:** a timetable managed in a homegrown Excel file. No proprietary format to decode, just some discipline to structure the data before the import.
2. **Before the import — sort things out** in the source spreadsheet. If your file mixes teachers, classes, rooms and courses on the same tab, first separate them into thematic tabs: teachers, students, classes, rooms, courses.
3. **Clean up** each tab: no unnecessary columns, **harmonize the labels** (a teacher must not appear under three different spellings), remove merged rows. One row = one entity.
4. **Omniscol test account:** create a test account before touching production. You validate the imports risk-free.
5. **Successive imports**, in dependency order: **reference data first** (teachers, rooms, subjects), **courses next** (they refer to them). Use the [Mass import of courses from a spreadsheet](#) for the courses — you paste your rows into it directly from the spreadsheet.
6. **Diagnostic:** Omniscol detects the inconsistencies (courses without a room, unknown teachers, conflicts) that your spreadsheet does not see. Fix them before switching over.
7. **Test generation** (optional — if you want to keep manual placement) to validate feasibility.
Switchover: your old Excel file stays read-only as an archive; Omniscol becomes the source of truth, with automatic distribution (iCal, panels, portals) and real-time updates.

🔗 See also

[Overview](#)[Preparing your data for a mass import](#)[Mass import of courses from a spreadsheet](#)

13. Higher-education specifics

13.1 Higher education specifics — overview

Source: <help/en/higher-ed/overview.md> · id: [higher-ed.overview](#) · Updated: 2026-06-13

Higher education, continuing education and training centers have constraints that differ widely from primary and secondary schools. This section gathers the pages that cover these specifics.

Typical characteristics

- **Calendar mode** rather than weekly — teaching is rarely recurring all year long; lessons tend to be dated one by one.
- **Many external instructors** — adjuncts, visiting professors, experts. See [External instructors](#).
- **Parallel cohorts** rather than single classes — a master's program can have several tracks that share some courses, with specializations in common but projects of their own.
- **Co-teaching** is common (teacher pairs, dual expertise).
- **Multi-room exams** (split lecture halls). See [Multi-room exams](#).
- **Videoconference links per course** (hybrid courses, partial remote learning). See [Videoconference links per course](#).
- **Students repeating a year or off-track students** to manage.

Configuration recommendations

- **Premium account** — it includes calendar mode, dated availability, multiple active timetables and **one-off events** (thesis defenses, juries, open days). The remote / hybrid qualification of lessons — their **modality** — is also part of it; the **videoconference link** by itself is available on all accounts.
- **Calendar mode** for the main timetable or timetables, essential in most higher education cases.
- **Multiple active timetables in parallel** if you mix recurring and one-off courses, if your institution covers several independent faculties, or if it is a group of schools.
- **Groups of groups** for evolving groupings.
- **Availability entry in calendar mode** — availability entered date by date by the instructors, consolidated in real time.
- **API + integrations** with your SIS or ERP — often [synchronization with external systems](#) in business and engineering schools. It synchronizes the entities (teachers, classrooms) and the course catalog, and reports the lessons as interventions to the central system, the institution's source of truth.

Vocabulary

What higher education institutions call a **curriculum**, a **program**, a **course catalog** or a **syllabus** corresponds, on the Omniscol side, to the **set of courses** of a class or a program of study — each course being a typed subject, assigned to a class, with its hour volume and its constraints. Its **lessons** are the scheduled occurrences.

The other higher education terms refer to more generic Omniscol entities, documented in the glossary: **learner** and **participant** map to the [student](#); **instructor**, **adjunct**, **permanent teacher** and **expert** to the [teacher](#); **cohort**, **session** and **intake** to the [class](#).

Higher education use cases

- [FAQ — higher education use cases](#)
- [FAQ — special cases and advanced configurations](#)

See also

[Calendar mode](#)

[Groups of groups](#)

[External instructors](#)

[Multi-room exams](#)

[Videoconference links per course](#)

13.2 Sessions, cohorts, programs, tracks

Source: <help/en/higher-ed/sessions-and-tracks.md> · id: `higher-ed.sessions-and-tracks` · Updated: 2026-05-12

In higher education, the teaching structure is less linear than in schools. A student belongs to a **cohort** (their year group), but also to **tracks** or **programs** (their specialization) and to tutorial or lab **groups**. They sometimes follow several cohorts at the same time (minor, double curriculum). This page summarizes how Omniscol models these situations.

Cohort = Omniscol class

The **cohort** (informally, the "class of" a given year) corresponds to the Omniscol class in the administrative sense, for example "*L3 Computer Science 2026*" or "*M2 Finance 2025-2027*". It is the entity that gathers the group of students who entered at the same session.

Classes are named unambiguously, with the entry date for large curricula:

- `L1 SI 2026` — first-year bachelor's degree in Engineering Sciences, 2026 intake,
- `M1 Marketing 2026-2028` — first-year master's degree, two years.

Tracks and specializations

A **track** is a study orientation within one cohort or across several: "*Data option*", "*Entrepreneurship minor*", "*XYZ double degree*". Omniscol modeling depends on the complexity:

- **Simple track** within one cohort — a **group** inside the class (for example `L3-Info 2026-Parcours Data`).
- **Cross-cohort track** spanning several cohorts — a **group of groups** that gathers the matching groups from several classes. See [Groups of groups](#).
- **Double curriculum** (students following two full tracks) — a special case where the same student receives **two class assignments** over the year. This is the exception to the general rule that two classes do not share learners.

Sessions

A **session** is a defined teaching period: a semester, a quarter, an intensive 2-week module. Omniscol handles sessions through:

- the **publication ranges** of timetables (one timetable per session, published on the corresponding weeks),
- the **school years** (a school year can cover several sessions).

For short or non-recurring programs (seminars, continuing education modules), prefer the **calendar mode** (included in the Premium plan) over the classic weekly grid. See [Calendar mode](#).

Students with atypical paths

Frequent cases in higher education:

- **Students repeating a year** who have passed some course units and not others.
- **Mobility students** (Erasmus, exchange) — assigned to a host class with track adjustments.
- **Work-study students** — alternate school weeks and company weeks; the rhythm is modeled with alternate weeks or with a dedicated calendar timetable.

See also

[Overview](#)[Class, group, subgroup](#)[Groups of groups](#)[Calendar mode](#)


13.3 External instructors (adjuncts, visiting faculty)

Source: <help/en/higher-ed/external-faculty.md> · id: [higher-ed.external-faculty](#) · Audience: [admin](#) · Updated: 2026-06-13

Higher education relies heavily on **external instructors**: sector-specific adjuncts, professionals who teach occasionally, visiting professors from other institutions. Managing them differs from managing permanent teachers: few hours, a schedule the school has to accommodate rather than set, sometimes contracts paid by the hour.

Marking an instructor as external

In the teacher profile ([Managing teachers](#)), the **External teacher** switch (available with the Premium plan) distinguishes adjuncts from permanent teachers. The marker has two concrete effects:

- an **icon**  in front of the instructor's name in lists and in lesson tooltips,
- a **dedicated column** in table views and spreadsheet exports, to single out adjuncts (for example to track the hours taught by permanent vs external teachers).

Entering availability

An adjunct's availability is often **date-based** rather than weekly: they agree to teach on Tuesday 12 March and Tuesday 19 March, but not every Tuesday. For this case, entering [availability in calendar mode](#), included in the Premium plan, is the natural solution.

Without the Premium plan, enter approximate **weekly availability** (Tuesday afternoons) then lock the lessons one by one by hand.

Service hours and contracting

Adjuncts generally have an **hours allocation** defined at recruitment (`30 hours over the year`, a `15-hour module`). The **Service hours** field in the profile tracks the allocation; the diagnostic raises an alert if the planned hours exceed it.

For hourly contracting (payment per lesson rather than per month), the per-teacher statistics export feeds the external payroll / invoicing process.

Communication


Adjuncts sign in to the application less often than a permanent teacher. Three use cases:

- **Full account** — the adjunct receives credentials and views their timetable, enters their availability, reports their absences from the teacher portal.
- **iCal link only** — no account, just an iCal subscription link to sync their lessons with their personal calendar.
- **E-mail reminder (outside the application)** — the administrator notifies the adjunct by e-mail before their teaching slots; the adjunct never signs in. This reminder is manual: Omniscol does not send an automatic notification per lesson. Such a reminder is easy to build outside Omniscol with the schedule-retrieval API, filtering on teachers. A weekly job, for example, builds a summary injected into the school's own template, then sends it through the school's mail service. Going through that service preserves the sender reputation and limits anti-spam false positives.

Calibrate according to the profile (a senior consultant has no time to manage yet another account; a doctoral student who teaches a few hours will adapt easily).

☰ How-to

Onboarding an adjunct

1. **An adjunct who teaches 15 hours over the year:** external marker, date-based availability, hours allocation, suitable communication mode.
2. **Create the teacher profile in Teachers.** Fill in first name, last name, e-mail, and enable **External teacher**. The marker adds the  icon in front of the name and a dedicated column in table views and exports.
3. **Hours allocation:** in the **Service hours** field, enter the contractual allocation (15). The diagnostic raises an alert if the planned hours exceed it. Handy for hourly payroll and contract tracking.
4. **Availability:** on a Premium account, the adjunct enters their availability **date by date** via [availability in calendar mode](#) — a perfect fit for a schedule they do not control. Otherwise, **approximate weekly availability** + locking the lessons one by one by hand.
5. **Communication mode** — choose according to the profile:
 - **Full account** — for a doctoral student who teaches a few hours; credentials, teacher portal, availability, absences;
 - **iCal link only** — for a senior consultant too busy for yet another account; just the calendar subscription;
 - **E-mail reminder (outside the application)** — the administrator notifies the adjunct by e-mail, by hand (Omniscol does not send an automatic reminder per lesson); the adjunct never signs in.
6. **Statistics export** at the end of the month or semester for the **external payroll/invoicing:** hours taught per lesson, subject, class. See [Print and share](#). The External column in the exports singles out the adjuncts.

🔗 See also

[Managing teachers](#)[Availability in calendar mode](#)[External teacher](#)[Overview](#)

13.4 Doubled-up classrooms and multi-room exams

Source: [help/en/higher-ed/multi-room-exams.md](#) · id: [higher-ed.multi-room-exams](#) · Audience: [admin](#) · Updated: 2026-06-13

In higher education, some lessons occupy **several classrooms simultaneously**. A midterm exam gathers a cohort of 200 students spread across 5 lecture halls. A doubled-up lecture runs in 2 lecture halls synchronized by video link. A jury uses 3 rooms in parallel. Omniscol handles these cases through the **multi-room** concept.

Multi-room: the principle

A lesson (or an event) can carry **several classrooms**. All these rooms are then occupied on the time slot, and all the constraints (capacity, specialization, availability) are checked in parallel.

Multi-room is available in **all timetable modes** (weekly, cyclic, calendar). See [Sites](#), [classrooms](#), [resources](#).

PREMIUM

The exam and jury scenarios below rely on **one-off events** (dated lessons laid on top of the grid), available on **Premium** accounts. **Multi-room** on a regular lesson remains available on all accounts.

Typical use case: an exam across several rooms

A cohort `L1 Info 2026` (200 students) sits an algorithmics midterm. No lecture hall in the school seats 200, so the exam takes place simultaneously in:

- `Amphi A` (80 seats),
- `Amphi B` (70 seats),
- `Amphi C` (50 seats).

Omniscol modelling:

- create an **exam event** on the date and time concerned (see [One-off events](#)),
- as participants: the class `L1 Info 2026`,
- as rooms: `Amphi A`, `Amphi B`, `Amphi C` (multi-room),
- for supervision: 1 to 3 supervisors per room via the **Staffing** module (see [Overview of the Staffing module](#)).

Check that the combined capacity of the lecture halls ($80 + 70 + 50 = 200$) covers the headcount. For an exam modelled as an **event**, Omniscol does not compute this total automatically: the capacity diagnostic only covers regular lessons placed on the grid.

Doubled-up lecture by video link

A very popular lecture can be streamed by video link from a main lecture hall to an overflow lecture hall. Modelling:

- a single lesson with two rooms,
- the teacher is physically in the main lecture hall,
- the lesson's **videoconference link** carries the streaming URL for the secondary lecture hall (a memo can clarify the arrangement).

Defence jury

A jury hearing 10 doctoral candidates in parallel in 3 rooms (rotating in thirds):

- one event per jury session,
- multi-room for each session,
- **Staffing** for the jury members assigned to each room.

Automatic distribution of students

Omniscol does not automatically assign which student goes to which lecture hall. The distribution remains a decision of the institution (alphabetical order, level, type of exam, internal rules). It is done by hand, or via a spreadsheet export sent to the supervisors.

☰ How-to

Scheduling a midterm across 3 lecture halls

1. **The typical higher-education case:** a midterm for 200 students across 3 simultaneous lecture halls. Multi-room, an event and the **Staffing** module are all you need.
2. **Create the exam event:** in the **Timetable** module, show the **Events** filter (reorganization mode) and **click and drag** on the agenda to draw the dated slot. Title: `Partiel algorithmique L1 Info`, then start and end dates and times. See [One-off events](#).
3. **Participants:** add the class `L1 Info 2026` (200 students). All the students concerned automatically end up on the event, whichever lecture hall they will physically sit in.
4. **Rooms — this is multi-room:** add the 3 lecture halls (`Amphi A 80`, `Amphi B 70`, `Amphi C 50`). The 3 lecture halls are then occupied simultaneously on the slot. Check yourself that the combined capacity (200) covers the headcount (200): on an event, this total is not diagnosed automatically.
5. **Supervisors:** open the **Staffing** module and create the supervision tasks for each lecture hall (1 to 3 supervisors per room depending on your policy). See [Overview of the Staffing module](#).
6. **Distributing students across lecture halls:** Omniscol does not do this automatically. That choice follows your internal rules (alphabetical order, level, balancing). Export the student list in spreadsheet format, do the distribution by hand or per your rules, then send it to the supervisors. See [Print and share](#).

📌 See also

[Multi-room](#)[Modality](#)[Sites, classrooms, resources](#)[Overview of the Staffing module](#)[One-off events](#)

13.5 Co-teaching and rotating instructors


Source: [help/en/higher-ed/co-teaching.md](#) · id: [higher-ed.co-teaching](#) · Audience: [admin](#) · Updated: 2026-06-13

Two common situations in higher education:

- **co-teaching** — two teachers (or more) lead the same lesson together. Typical case: a theorist + a practitioner on a strategy course, a permanent teacher + an expert adjunct on a case study.
- **rotating instructors** — a course over a period (a module, a semester) is taught in turn by several teachers who take over from one another. Typical case: a sector-specific module broken down into thematic lessons, each entrusted to a different expert.

Both mechanisms are natively supported by Omniscol.

Co-teaching (joint delivery)

A lesson can carry **several teachers simultaneously**. In the **Timetable management** module, the teacher selector  **Assign teachers** accepts several entries on a single lesson. The individual timetables of the teachers involved show the lesson as co-taught.

Consequences:

- The **availability** of all co-teaching teachers must be compatible on the time slot.
- For **service hours**, the lesson appears **in full** in each co-teacher's timetable. If your policy splits the hour pro rata between co-teachers, that calculation happens at export time (payroll, invoicing).

See [Complex lessons](#) for the technical details of co-taught courses.

Rotating instructors on a module

More complex: 12 lessons in a module, each taught by a different teacher. Two approaches:

Approach A — One subject, several courses

Create **one course per instructor** in the module, each with its own assigned teacher. Data entry takes longer, but the result stays clear for the instructors: each sees their lessons in their timetable and the module appears as a subject with several courses.


Approach B — Calendar mode with specific dates

If the lessons have known specific dates, use [calendar mode](#), included in the Premium plan:

- one subject `Module Stratégie d'entreprise`,
- N dated lessons with one instructor per lesson,
- the algorithm respects each instructor's availability.


This approach is more compact and natural for intensive modules.

Memo to describe the sequence

You can add a **memo**  `Comment` on each lesson explaining the lesson's place within the module (`Séance 3 / 12 - Stratégie financière`). The memo appears in the timetables and helps students find their way.

How-to

Module with 12 rotating instructors

1. **The sector-specific module case:** 12 lessons, each entrusted to a different expert. Two mechanisms are combined here: the **calendar approach** to date the rotating lessons, and **co-teaching** on only those lessons delivered by a pair.
2. **For this variant with specific dates:** a timetable in **calendar mode**. All external instructors are marked **External** on their teacher profiles. See [External instructors](#).
3. **Create the single subject** `Module Stratégie d'entreprise`. It is the pedagogical envelope of the module. Pick a suitable **Type of course**: often a mix of lectures + case studies.
4. **Enter the 12 lessons** on specific dates, each with **its instructor**. Put the lesson title + its theme (`Séance 3/12 - Stratégie financière`) in the memo to help students keep track of the progression.
5. **For lessons taught by a pair** (theorist + practitioner on a case), use **co-teaching**: on the lesson concerned, select **several instructors** in  `Assign teachers`. The individual timetables of the teachers involved show the lesson as co-taught.
6. **Service hours:** the lesson appears **in full** in the timetable of each co-teacher. A pro-rata split between co-teachers is handled at export time (payroll, invoicing), not in Omniscop.
7. **Verification:** open the student portal to check how the module is displayed — the 12 lessons appear in the timetable with their respective instructors. The progression memos (`Séance N/12`) are visible on hover.

See also

[Co-teaching](#)

[Complex lessons](#)

[External instructors](#)

[Calendar mode](#)

13.6 Calendar mode for non-recurring programmes

Source: help/en/higher-ed/calendar-mode.md · id: [higher-ed.calendar-mode](#) · Audience: admin · Plan: premium · Updated: 2026-06-13

👑 PREMIUM

PREMIUM

Higher education does not always follow a **weekly recurrence**: intensive modules over 2 weeks, 3-day seminars, defence sessions spread over a month, one-off workshops led by external instructors. For these cases, the Omniscol **calendar mode** is more natural than a weekly grid riddled with exceptions.

When to prefer calendar mode

Indicators:

- lessons have **specific dates** rather than a repeated weekly time slot,
- the **instructors vary** from one lesson to the next,
- the **locations vary** (field trip, company visit, videoconference),
- the **pace** is not weekly (sometimes 3 lessons in 2 days, sometimes nothing for 3 weeks).

If you would describe the module with a calendar rather than a weekly grid, calendar mode is made for you.

Typical use cases in higher education

- **Intensive modules** — a seminar of 3 full days or an intensive module over 2 weeks.
- **Continuing education** — one-off sessions that cannot be carried over into an annual grid.
- **Research seminars** — one-off lessons with different guests each week.
- **Defences and juries** — spread over a few weeks, specific dates, specific rooms.
- **Fieldwork / projects** — short phases, varying locations.

Modelling

You define an **ordered list of dated lessons**, without going through a weekly grid:

- a specific **date and time**,
- a **duration** (free-form),
- a **location**,
- an **instructor** (may vary from one lesson to the next),
- an **audience**: class(es), group(s), groups of groups.

Calendar mode uses the same class / group structure as weekly mode — you lose none of the organizational tools.

The [automatic generation](#) also works in calendar mode: the solver places lessons within a target date window and can compact teaching days at the start or end of the period. See [Calendar mode](#).

Combining with a weekly timetable

You can have a **weekly timetable** for the recurring common core in the morning and a **calendar timetable** for the one-off lessons in the afternoon, both published in parallel for the same classes; or the first years on a weekly timetable, and the final-year specialization classes in calendar mode (see [Multiple active timetables in parallel](#)).

Instructor availability in calendar mode

Entering [availability in calendar mode](#) is particularly useful here: adjuncts enter their availability on exact dates, not on a weekly pattern.

See also

[Calendar mode — product view](#)

[Choosing the right timetable type](#)

[External instructors](#)

[Multiple active timetables in parallel](#)

[Availability in calendar mode](#)

13.7 Videoconference links per course


Source: [help/en/higher-ed/videoconference-links.md](#) · id: [higher-ed.videoconference-links](#) · Audience: [admin](#) · Plan: [premium](#) · Updated: 2026-06-13

PREMIUM

PREMIUM

In higher education, many courses are **remote** or **hybrid** (some students in person, others connected). Each lesson then has a **videoconference link** (Zoom, Teams, Google Meet, Jitsi) that must appear in the timetable so that everyone knows where to click.

How to attach a link

On a lesson or an event, the [Videoconference link](#) field accepts a URL. For a lesson, you enter it in the classroom selection window. Once set, the timetable displays the link with a  icon; the link opens in a new tab.

The link appears:

- in the student and teacher portal on the lesson,
- in the iCal export through the conference or URL properties when the link is present,
- in the JSON API.

Remote or hybrid courses

The lesson's [Modality](#) field describes its format: **In person**, **Remote**, **Hybrid** or **Self study**. You set it in the same window as the classroom and the videoconference link.

For a **fully remote** course, choose the **Remote** modality, enter the videoconference link and do not force a physical classroom if the course does not use one. For a **hybrid** course (in-person + videoconference in parallel), choose **Hybrid**, keep the physical classroom and enter the link as well.

Permanent links vs per-lesson links

Depending on your videoconference tool:

- **Permanent link** — personal Zoom room, recurring Teams meeting. The same link is used for all the lessons of a course. Enter it once on the course (all its lessons inherit it), or directly on the class to apply a default link to all its lessons.
- **Per-lesson link** — each lesson has its own URL, entered on the lesson (in the same window as the classroom). Reserve this case for lessons whose link changes every time (a Zoom meeting generated on the fly, for example).

Security

A few good practices:

- **No public link on sensitive courses** — for a private course or a proctored exam, configure the videoconference room with authenticated access (not direct-link access).
- **Link in iCal** — the iCal feed can be read by anyone who has the share link. A videoconference link placed on the lesson is therefore distributed through iCal — check that this is what you want.

- **Renewal** — if a permanent link is compromised, renew it on the Zoom / Teams side and propagate the update in Omniscol (the course or the class, not each lesson).

See also

[Modality](#)[Types of course](#)[iCal — subscription and dynamic link](#)[FAQ — higher education use cases](#)

13.8 Multi-site in higher education

Source: [help/en/higher-ed/multi-site.md](#) · id: [higher-ed.multi-site](#) · Audience: [admin](#) · Updated: 2026-06-13

Multi-site is a capability **available from the Standard plan upwards** (disabled only in Lite); the general principles are described in [Sites](#), [classrooms](#), [resources](#), [multi-room](#). This page lists the **specifics frequently seen in higher education**, where multi-site is almost the norm: engineering schools across several campuses, business schools with regional locations, multi-site universities.

Note that in Omniscol, the [campus](#) is an organizational notion distinct from the [site](#). The two often overlap (one campus per site, with its dedicated classes). When you need to manage a time grid, classrooms or travel times, model the location as a site.

These scenarios also apply in primary and secondary education when the school is geographically distributed (a separate middle school + high school, a school with a preschool annex, etc.).

Typical case: a main site + regional branches

A frequent case in engineering and business schools:

- 1 **main site** (typically the headquarters),
- 1 to N **regional branches** where off-site cohorts follow part of the curriculum,
- **instructors** who shuttle between the sites.

Omniscol modelling: one site per location, travel times declared per pair (the high-speed trains / flights / car journeys between the sites). See [Multi-site policy](#) for the exact mechanics (blocking constraints to the minute, lunch break reduced by the travel time, etc.).

Separate timetable per site or global timetable

Two philosophies depending on the internal organization:

- **Consolidated global timetable** — a single timetable covers all sites. Suited when the planning teams are centralized and the instructors move around a lot. The diagnostic cross-checks everything in a single pass.
- **One timetable per site** — thanks to [multiple active timetables in parallel](#), included in the Premium plan. Each site has its local team, its own timetable active over the same weeks; when viewing, the views are merged dynamically for the students or instructors involved with several sites.

Virtual sites and remote lessons

Special case: a lesson **streamed by video link** from a main site to a fully remote cohort. Possible modelling:

- create a **virtual site** “Remote” with no travel time to the broadcasting site, which lifts any travel-related time constraint,
- place one “Video” room per remote cohort in it,
- attach that room to the lesson in multi-room together with the original lecture hall.

This modelling stays limited if you have **many remote cohorts**: multiplying the virtual “Video” rooms becomes heavy to maintain and managing their capacity does not really make sense. In that case, document the need as specific and prefer the videoconference link directly on the lesson (or, on Premium, as a default on the class), without creating one virtual room per cohort.

See [Multi-room exams](#) for the multi-room mechanics, and [Videoconference links per course](#) for the associated Zoom / Teams / Meet links.

Filters and statistics per site

Omniscol screens expose **per-site** filters, to isolate a physical location. Useful for:

- displaying the timetable of one specific location,
- monitoring a branch's classroom occupancy separately,
- generating per-site statistics (hours taught, occupancy rates).

Display panels per site

A [display panel](#) can filter on a specific site, which generally corresponds to a panel installed **in the lobby** of one site. See [Configuring a panel for a lobby or corridor](#).

See also

[Sites, classrooms, resources](#)

[Site](#)

[Multiple active timetables in parallel](#)

[Multi-room exams](#)

[Videoconference links per course](#)

[Modality](#)

[Configuring a display panel](#)

[Higher education overview](#)

14. Primary / secondary specifics (school)

14.1 Primary and secondary specifics — overview

Source: <help/en/k12/overview.md> · id: [k12.overview](#) · Updated: 2026-06-13

This section gathers the pages that cover the specifics of primary and secondary education (elementary school, middle school, high school).

Typical characteristics

- **Weekly timetable** — the recurring standard week is the norm, with or without [alternate weeks](#) (A/B).
- **Dedicated rooms per class** — many schools operate with a room attached to each class; it is the teachers who move from room to room.
- **Half classes / elective groups** — handled through [class divisions](#) within a single class.
- **Latin students / Greek students / high-school specialties** — handled through cross-class [alignments](#).
- **Automatic generation** — the solver does the job, and this is the heart of the value for schools.
- **Study halls** — periods when students are at school with no assigned lesson. See [Study halls and supervised study](#).
- **Multi-grade classes** — typical of small rural schools. See [Multi-grade classes](#).

Configuration recommendations

- A **standard account** is enough for the majority of cases (the Premium tier is not necessary for a standard school).
- **Weekly mode** by default.
- **Enable teacher availability entry** in weekly mode; the entry can be made or reviewed by the administration.
- **One dedicated room per class** where applicable — entered on the class.
- **Careful setup of levels** in the **Administration** module, via [+ Create](#) (Year 1, Year 2, ..., Grade 6, Grade 7, ..., Grade 10, Grade 11, Grade 12, etc.).
- **Official holidays of the country** — when creating the school year, Omniscol offers to import the country's common holidays. See [School year and holidays](#).

School use cases

Omniscol's historical base is secondary education (middle and high school): many default conventions come from there. The step-by-step scenarios applicable to schools are gathered in the [use-case scenarios](#) and the [timetable creation FAQ](#).

See also

[What is Omniscol](#)[Choosing the right timetable type](#)[Half classes and electives in class divisions](#)[Study halls and supervised study](#)[Multi-grade classes](#)

14.2 Half classes and electives in class divisions

Source: <help/en/k12/half-classes-and-options.md> · id: [k12.half-classes-and-options](#) · Audience: [admin](#) · Updated: 2026-06-13

In primary and secondary schools, managing a class quickly gets more complicated with **half classes** (labs / practical work in reduced numbers) and **electives** that split the students of one class into several simultaneous subgroups. The Omniscol concept that structures this is the **class division**: subgroups that must have lessons at the same time but with different content.

Half classes for labs

A typical case in the first years of secondary school: 28 students as a whole class most of the time, split into 2 groups of 14 for science labs. The two half groups must have their lab at the same time (otherwise they could not have another lesson in parallel), each with its own teacher and room.

Modeling in Omniscol:

- create two **groups** `6A-TP1` and `6A-TP2` in class `6A`,
- declare them as a **class division**: Omniscol guarantees they are busy simultaneously,
- create **two science lab courses**, each with its own group, teacher and room, on the same weekly time slot.

The algorithm honors the class division: `6A-TP1` can never have science while `6A-TP2` is in French.

Electives in a class division

A typical case in the last years of lower secondary school: on one time slot, some students take Latin, others advanced German, others take nothing (study hall). Three simultaneous groups within one class.

Modeling:

- groups `Year9A-Latin`, `Year9A-French`, `Year9A-Study-Hall` in class `Year9A`,
- declared as a **class division** (all three are simultaneous),
- one course per group on the same time slot, each with its own teacher / room.

Students who change electives during the year

An elective can evolve during the year (a student drops Latin to join the study hall group, for example). Group membership is managed **week by week** in the student's record: the weeks already past keep the old assignment, the following weeks take the new one.

Class division + alignment

For languages taught **across several classes** (for example Latin open to 4A, 4B and 4C), combine:

- a **class division** per class (`Year9A-Latin / Year9A-Other-Option`, `Year9B-Latin / Year9B-Other-Option`, etc.),
- an **alignment** between the three Latin groups so that they share the same time slot, in the same room, with the same teacher.

See [Group alignments](#).

☰ How-to

Create a Latin/French/study hall slot for a class

1. **The typical late-lower-secondary case:** on one time slot, some students take Latin, others German, others nothing (study hall). Three simultaneous groups. The Omniscol class division handles this painlessly.
2. **Create three groups** in class `Year9A`: `Year9A-Latin`, `Year9A-French`, `Year9A-Study-Hall`. Assign each student to their group according to their elective choice.
3. **Declare the class division** (`Year9A-Latin`, `Year9A-French`, `Year9A-Study-Hall`): the three groups are **exclusive and simultaneous**. Omniscol will guarantee they are busy on the same time slot.
4. **Create three courses** on the same weekly time slot:
 - Latin for `Year9A-Latin` with the Latin teacher and a suitable room;
 - French for `Year9A-French` with the French teacher;
 - study hall for `Year9A-Study-Hall` with a supervisor (or leave it without a teacher, depending on your policy).

The solver respects the class division: all three will necessarily share the same time slot.

5. **For electives open to several classes** (Latin in Year9A + Year9B + Year9C with a single teacher and a single room): combine a **class division per class** (`Year9A-Latin` exclusive of the other Year9A groups, same for Year9B and Year9C) **+ an alignment** between the three Latin groups. See [Group alignments](#).
6. **Changes during the year** (a student drops Latin to join the study hall group): change their group membership in their student record, starting from the week of the change. Past weeks keep the old assignment.

🔍 See also

[Class divisions](#)[Class, group, subgroup](#)[Group alignments](#)[Multi-grade classes](#)

14.3 Study halls and supervised study (primary and secondary)

Source: [help/en/k12/study-halls.md](#) · id: `k12.study-halls` · Audience: `admin/staff` · Updated: 2026-06-13

Study halls (or **supervised study**) are periods during which students who have no lesson are hosted in a study room under supervision. They are a typical institution of French middle and high schools: students who do not take the elective running on a slot, those exempt from PE, those whose teacher is absent.

Why not just a regular course?

A study hall is not managed exactly like a course:

- **no subject** in the usual sense,
- **a changing audience** from one time slot to the next (whoever has no lesson),
- **supervision rather than teaching** — the supervisor does not teach, they supervise,
- **a strong demand for fairness:** the supervision load must be distributed among the teachers (and pastoral staff / monitors) according to a fairness rule.

This is why Omniscol offers the dedicated **Staffing** module for this type of management. See [Overview of the Staffing module](#). This module is an account **option**: without it, model study halls with a **study group** (section below); the supervision grid itself belongs to the module.

Modeling with the Staffing module

Study halls translate into duty grids:

- a **weekly grid** of the time slots where students can be in study hall,
- for each time slot, one or more supervision **tasks** (depending on the number of students expected and the desired monitor / student ratio),
- **assignment** of the monitors (education assistants, or teachers with the **Staff** role) to each task.

See [Building a service grid](#) and [Assigning staff](#).

Link with student timetables

When a student has no lesson on a time slot (their class is on an elective or half of the class is in a lab), their individual timetable simply shows a **free slot**: Omniscol does not automatically display a "Study hall" lesson there.

To make the study hall appear in the timetable of the students concerned, create a **study hall course** for the group that has no lesson on that slot, with the study room — this is the `4A-Étude` group pattern described in [Half classes and electives in class divisions](#). Supervision coverage, for its part, is handled on the **Staffing** side — see [Overview of the Staffing module](#).

Supervised study vs free study

A useful distinction:

- **Supervised study** — a supervisor is present, students work in silence. The classic case for **Staffing**.
- **Free study** (common room, self-directed work space) — no dedicated supervisor, simply a place where students can wait. No need for **Staffing**: model it as a dedicated room; a **classroom specialisation** can, if needed, prevent ordinary courses from being placed there (without reserving the room for the study hall itself, which has no subject). See [Classroom specialisations](#).

See also

[Overview of the Staffing module](#)

[Building a service grid](#)

[Assigning staff](#)

[Half classes and electives in class divisions](#)

14.4 Multi-grade classes

Source: [help/en/k12/multi-grade-classes.md](#) · id: `k12.multi-grade-classes` · Audience: `admin` · Updated: 2026-06-13

Multi-grade classes bring together students from several grade levels in the same room, under the responsibility of a single teacher. This is the reality of **small rural schools** at primary level, and sometimes of **specialized schools** or **international schools** with small enrollments.

Why Omniscol handles these cases

The Omniscol model — which separates the class (administrative) from groups (teaching subsets) — natively covers multi-grade setups. The same administrative class `Grade 2/3` can carry two groups `Grade 2` and `Grade 3`, sometimes with a shared course, sometimes with courses split by grade level.

Modeling

Two approaches, depending on the nature of the courses taught:

Approach A — one administrative class, two teaching groups

- Administrative class: `Grade 2/3` (for student management, the parent list, the room).
- Teaching groups: `Grade 2` and `Grade 3`.
- **Shared courses** (music, sports, homeroom) are assigned to the whole class `Grade 2/3`.

- **Separate courses** (reading / math at each group's level) use the groups `Grade 2` and `Grade 3`, declared as a **class division** so that they take place simultaneously (the single teacher teaches one group while the other works independently).

Approach B — two administrative classes, a single teacher

- Separate administrative classes: `Grade 2` and `Grade 3`.
- The single teacher is assigned to the courses of both classes.
- Since a teacher cannot run two lessons at the same time, Omniscol reports a conflict if their `Grade 2` and `Grade 3` lessons are placed on the same time slot.

Approach B is rarer because it complicates administrative management (two distinct classes for what is in practice a single set of students) and it offers no mechanism as direct as the class division to organize the alternation between grade levels.

Three grade levels and beyond

The mechanism extends to three or more grade levels (the case of very small schools: Grades 1 through 5 in the same room). Omniscol imposes no numerical limit — the complexity comes from the teaching organization, not from the modeling.

Study halls and independent work

When the single teacher is with one grade level, the other grade level works independently in the same room. There is no need to model this as a study hall — it is intrinsic to how a multi-grade class operates and it is not a separate supervision duty.

How-to

Model a multi-grade `Grade 2/3` class

1. **The typical rural school case:** a `Grade 2/3` class with a single teacher who alternates between the two grade levels. Approach A (recommended): one administrative class, two teaching groups.
2. **Create the administrative class** `Grade 2/3` in **Classes**. It carries the student list, the parents, the room, the communications. Level: assign the most representative level or create a dedicated level `Multi-grade primary`.
3. **Create the two teaching groups** `Grade 2` and `Grade 3` within the class. Assign the students to their respective group. The two groups together = the whole class.
4. **Declare the groups as a class division** (`Grade 2`, `Grade 3`): the division **allows** their **simultaneous** placement (it lifts the conflict that would otherwise forbid it) and the solver **seeks** to place the `Grade 2` and `Grade 3` courses on the same time slots. The room and the teacher remain assigned course by course. See [Class divisions](#).
5. **Shared courses** (music, sports, homeroom): assign them to the whole class `Grade 2/3`. All students attend them together.
6. **Courses split by grade level** (reading, math): create one course per group `Grade 2` and `Grade 3`, on the same time slot (as the class division requires). Assign the teacher to the course they lead and leave the other course without a teacher (independent work): the same teacher assigned to two simultaneous lessons would trigger a conflict. No need to model a study hall; independent work is intrinsic to how a multi-grade class operates.
7. **For very small schools** (Grades 1 through 5 in the same room), the mechanism extends without limit. You create as many groups as there are grade levels, declared as a class division. The complexity comes from the teaching organization, not from the Omniscol modeling.

 **See also**

[Class, group, subgroup](#)

[Class divisions](#)

[Half classes and electives in class divisions](#)

15. Use-case scenarios

15.1 Use-case scenarios — overview

Source: <help/en/use-cases/overview.md> · id: [use-cases.overview](#) · Updated: 2026-06-13

This section answers a recurring request from large institutions: having a **list of operational scenarios** in *user story* format, with a concrete explanation of "how it is done in Omniscol". Each available scenario links to a reference page or an existing guided tour; scenarios still under review are tracked separately and do not appear in this list.

The scenarios come from real feedback from customers and prospects (business schools, engineering schools, universities, continuing education), anonymized and phrased in generic terms.

Format

Each scenario follows the classic Agile format:

As a (*user role*) **I want** (*action*) **so that** (*business goal*)

Followed by a **solution in Omniscol**: a short text, a link to the detailed reference page, and an interactive tour only when a reliable guided tour already exists.

Statuses

Each scenario carries an implementation status:

Status	Meaning
✓ Available	Feature in production, tested and validated.
● Partial	Incomplete coverage; a workaround exists or the feature applies to some modes only.
△ Roadmap	Not available in the current interface; identified need or separate product plan.
— Out of scope	Handled by another tool in the ecosystem, or outside Omniscol's functional scope.

Contents by theme

- [Timetable creation and generation](#)
- [Day-to-day placement and modification](#)
- [Multi-group, multi-room, multi-instructor](#)
- [Mass operations](#)
- [Exams and events](#)
- [Absences and substitutions \(scenarios\)](#)
- [Reporting and statistics](#)
- [Distribution and sharing](#)

Tours and videos

When a scenario contains a validated guided tour, the same content can feed several surfaces:

- a **popover in the help panel** (interactive mode, the user clicks step by step),
- a **subtitle embedded in the video** (passive mode, linear playback),
- an associated **WebVTT file** if a video is produced.

The subtitles are synchronized with the sequence of steps: a single source produces all three formats.

A scenario without verified steps remains a plain reference page, with no guided tour or associated video.

How to read a scenario

A typical scenario page contains:

```
## Short scenario name

> As a <role>, I want <action>, so that
> <goal>.

Status: ✓ Available

Solution in Omniscol. Short, operational explanatory text.

[Link to the detailed reference page](id:timetables.publication)

`<optional guided tour for the interactive and video versions>`
```

Scenarios with status ✓ or ● are **usable today**. Scenarios with status △ Roadmap are tracked separately and are not listed here.

See also

[FAQ — higher education use cases](#)

[FAQ — special cases and advanced configurations](#)

[Higher education specifics — overview](#)

15.2 Scenarios — Timetable creation and generation

Source: <help/en/use-cases/creation-and-generation.md> · id: [use-cases.creation-and-generation](#) · Updated: 2026-06-13

Initial creation, automatic generation, sandbox, and timetable duplication for tests and comparisons.

For the reading format and the status legend, see [Scenarios — overview](#).

Run an automatic timetable generation

As a planner, I want to run an automatic timetable generation, **so that** a first usable draft is produced quickly.

Status: ✓ Available

Solution in Omniscol. Automatic generation is available for all three timetable types — weekly, cyclic and calendar. The solver places the lessons that were created but not yet positioned, and can reposition lessons already placed if this improves the result, except for locked lessons. If no complete solution exists, Omniscol returns the best computed timetable and leaves the remaining lessons in the list of unpositioned sticky notes.

[Automatic generation](#)

Rerun a generation after changing constraints

As a planner, I want to rerun a generation after changing constraints, **so that** different configurations can be tested without starting from scratch.

Status: ✓ Available

Solution in Omniscol. Adjust the availabilities, the constraints or the locks, then rerun the generation from the same screen. Locked lessons keep their position; the others can be repositioned if this improves the result. To compare several configurations, work on copies of the timetable (“Work on a copy (sandbox) of a timetable”).

[Diagnosing a failed generation](#)

Work on a copy (sandbox) of a timetable

As a planner, I want to work on a copy (sandbox) of a timetable, **so that** I can experiment without affecting the official version.

Status: ✓ Available

Solution in Omniscol. Duplicate the timetable from the **Timetable management** module and work on the copy: as long as it is not activated, the official version remains the one users see. Activate the copy when it is ready.

[Visualize, duplicate, reorganize, Publishing \(activating\) a timetable](#)

See also

[Scenarios — overview](#)

[Overview of the Timetable management module](#)

[Automatic generation](#)

[Solver](#)

15.3 Scenarios — Day-to-day placement and modification

Source: help/en/use-cases/placement-and-modification.md · id: use-cases.placement-and-modification · Updated: 2026-06-13

Finding available classrooms and teachers, positioning and modifying lessons, detecting conflicts and omissions.

For the reading format and the status legend, see [Scenarios — overview](#).

Find an available classroom suited to a lesson (capacity, type, site) and assign it

As a planner, I want to find an available classroom suited to a lesson (capacity, type, site) and assign it to a session, **so that** I can quickly position a lesson without conflicts.

Status: ✓ Available

Solution in Omniscol. The classroom selector accepts a text search and capacity filters (>20 , ≤ 50 , $=30$, $20-30$). Several criteria separated by a space are interpreted as a logical AND; several values separated by a comma as a logical OR. The same search accepts "ok" (vs "ko") to keep only the classrooms free on the time slot.

Find an available teacher who teaches the subject

As a planner, I want to find an available teacher who teaches the subject, **so that** I can easily replace or complete an assignment.

Status: ✓ Available

Solution in Omniscol. Teacher selection takes into account the subject taught and the teacher's availability. Depending on the screen, the replacement is made directly from the lesson or from the course list.

Identify the available time slots for a class or a group

As a planner, I want to identify the available time slots for a class or a group, **so that** I can place a lesson

manually and efficiently.


Status: Partial

Solution in Omniscol. Overlay the timetables of the class or group in the multi-entity views to spot the free time slots (see "View timetables").

Quickly modify a lesson (classroom, teacher, time)

As a planner, I want to quickly modify a lesson (classroom, teacher, time), **so that** I can adjust the schedule in real time.

Status: Available

Solution in Omniscol. Open the lesson to change the classroom, the teacher or the time in its edit form, or reposition it with its pin button  (click a colored time slot) in the schedule view. A **multiple selection** (Shift+click on several lessons) opens the same form in bulk mode: reposition, unfix or delete all the selected lessons in a single operation.

Ad-hoc changes

Detect lessons without a teacher or a classroom

As a planner, I want to detect lessons without a teacher or a classroom, **so that** I can immediately correct critical omissions.

Status: Available

Solution in Omniscol. The timetable diagnostic explicitly lists placed lessons without a classroom and lessons without a teacher; the schedule views flag them as well.

Conflicts and diagnostic

Be alerted in case of inconsistency (missing classroom, missing teacher, conflict)

As a planner, I want to be alerted in case of inconsistency (missing classroom, missing teacher, conflict), **so that** I can safeguard the quality of the schedule.

Status: Partial

Solution in Omniscol. Inconsistencies are flagged in the diagnostics and the schedule views: lesson without a classroom, without a teacher, classroom, class or teacher conflicts. The diagnostic panel can be **finely filtered** — by severity, by alert type, by entity (class, teacher, classroom), by level, site or campus, and by date range in calendar mode (Premium feature; see [Conflicts and diagnostic](#)).

Workaround (without Premium). Use the chronological list of alerts and the available viewing filters to isolate the lessons concerned. Fix blocking conflicts before distribution or before feeding the data back into a third-party system.

Quickly view the remaining conflicts

As a planner, I want to quickly view the remaining conflicts, **so that** I can finalize a reliable schedule before distribution.

Status: Available

Solution in Omniscol. The timetable diagnostic and the schedule views flag the remaining conflicts. Deal with blocking conflicts first, before publication.

Conflicts and diagnostic

Collect instructor availability / preferences

As a planner, I want to collect instructor availability / preferences.

Status: ✓ Available

Solution in Omniscol. Teachers enter their availability and unavailability from their own space; the planner can also enter it for them. This availability is visible during manual placement and taken into account by automatic generation.

Teacher availability

Manage time slots / time grids / off-grid

As a planner, I want to manage time slots / time grids / off-grid.

Status: ✓ Available

Solution in Omniscol. Time grids define, site by site, the time slots open for placement, breaks included.

PREMIUM

The [off-grid lessons](#) cover the lessons that do not match any time slot of the grid.

Time grid, time slots and durations

Manage breaks / off-grid

As a planner, I want to manage breaks / off-grid.

Status: ✓ Available

Solution in Omniscol. See "Manage time slots / time grids / off-grid": breaks are defined in the time grid of each site, and off-grid covers lessons outside the time slots.

View timetables (multiple selection, custom views, overlay)

As a planner, I want to view timetables (multiple selection, custom views, overlay, visibility, export).

Status: ✓ Available

Solution in Omniscol. The viewing modes cover the week, day, month, list and side-by-side views and several comparison modes. Multi-entity viewing displays several timetables next to each other, by column or as a grid depending on the context.

Not available in the current interface: enriched tooltip with the subject code, automatic hiding of weeks without lessons, single merged view without a distinguishing column.

Show only the available classrooms

As a planner, I want to show only the available classrooms, **so that** I can save time when assigning classrooms.

Status: ✓ Available

Solution in Omniscol. Use the availability filter in the classroom selector to hide the classrooms already occupied on the time slot (see also "Find an available classroom suited to a lesson").

See also

[Scenarios — overview](#)

[Viewing and filtering](#)

[Ad-hoc changes](#)

[Conflicts and diagnostic](#)

15.4 Scenarios — Multi-group, multi-room, multi-instructor

Source: <help/en/use-cases/multi-entities.md> · id: [use-cases.multi-entities](#) · Updated: 2026-06-13

Sharing lessons across several groups, several classrooms, several instructors: co-teaching, alignments, groups of groups.

For the reading format and the status legend, see [Scenarios — overview](#).

Share lessons (1 lesson for 2 groups at the same date/time in the same classroom)

As a planner, I want to share lessons (1 lesson for 2 groups at the same date/time in the same classroom).


Status: ✓ Available

Solution in Omniscol. Model the shared lesson with an [alignment](#) or a [group of groups](#): the shared lesson brings the groups together on the same time slot, in the same classroom, with the same teacher.

Place a lesson with several instructors (co-teaching)

As a planner, I want to place a lesson with several instructors (co-teaching).

Status: ✓ Available


Solution in Omniscol. A lesson can carry several teachers: add the co-instructors in the teacher selector  [Assign teachers](#) of the lesson form.

[Co-teaching](#)

Place a lesson with several classrooms

As a planner, I want to place a lesson with several classrooms.

Status: ✓ Available

Solution in Omniscol. Assign several classrooms to the same lesson from the classroom selector  [Assign a classroom](#).

[Multi-room exams](#)

Place a lesson with several groups

As a planner, I want to place a lesson with several groups.

Status: ✓ Available

Solution in Omniscol. You can **directly attach several groups** — from the same class or from different classes — to a single lesson. For lasting or recurring groupings, structure them instead with a [group of groups](#) or an [alignment](#) as appropriate.

Place lessons by group in bulk

As a planner, I want to be able to place lessons by group in bulk, so that I can assign groups in bulk, in one click.

Status: ✓ Available

Solution in Omniscol. Two levers cover bulk placement.

- **Multiple selection, then the pin.** Select several lessons at once — with **Shift+click**, or through the **Extend the selection** menu (by day, week, month, day of the week, subject or group division). The **pin** then shows the time slots compatible with the whole selection: a single click places them together on the same time slot.
- **Automatic generation.** To place all the lessons on the grid at once, let the solver distribute them while respecting the constraints (see [Automatic generation](#)).

To bring several groups together on a single lesson, see "Place a lesson with several groups" above; to track the placed volumes, rely on the [hours distribution](#).

See also

[Scenarios — overview](#)

[Multi-room](#)

[Co-teaching](#)

[Class divisions](#)

[Group alignments](#)

[Groups of groups](#)

15.5 Scenarios — Mass operations

Source: [help/en/use-cases/mass-operations.md](#) · id: [use-cases.mass-operations](#) · Updated: 2026-06-28

Mass assignments, advanced filters and automations to quickly manage classrooms, capacities and programs.

For the reading format and the status legend, see [Scenarios — overview](#).

Mass-assign only the unassigned classrooms

As a planner, I want to mass-assign only the unassigned classrooms.

Status: ✓ Available

Solution in Omniscol. In the list view of the **Timetable management** screen, sort or filter the columns to isolate the lessons **awaiting classroom**, tick them, then run the automatic classroom assignment on the selection.

Automatic generation also offers a classrooms-only mode: run while ignoring classrooms already assigned, it only completes the lessons that have none, without recalculating the placement.

[Automatic classroom assignment](#)

Mass-assign all classrooms with >115 seats for events

As a planner, I want to mass-assign all classrooms with >115 seats for events, so that classroom management is optimized.

Status: ✓ Available

Solution in Omniscol. Spot the large classrooms in the classroom statistics, where the search accepts capacity comparators (for example >115). Then select the lessons concerned in the list view of the **Timetable management** screen and run the automatic classroom assignment: in the selector of candidate classrooms, tick only the classrooms you want; each one's capacity is displayed to guide the choice.

Restrict classroom assignment to a building or a floor

As a planner, I want to restrict mass classroom assignment to a given building or floor, so that lessons are grouped in the same place and travel is limited.

Status: ✓ Available

Solution in Omniscol. In the automatic assignment, the selector of candidate classrooms groups classrooms by **site** then by **building**: tick a whole site or building to allow only its classrooms, and the algorithm picks among them. See [Automatic classroom assignment](#).

Reassign classrooms based on actual headcounts

As a planner, I want to mass-reassign classrooms based on actual headcounts (electives, languages, groups), so that classroom size is adjusted to the number of enrolled students.

Status: ✓ Available

Solution in Omniscol. Isolate the lessons concerned in the list view (per-column sorting and filters), then run the automatic assignment: the algorithm compares each classroom's **capacity** with the lesson's **headcount** and keeps the smallest classroom that fits. See [Automatic classroom assignment](#).

See also

[Scenarios — overview](#)

[Automatic classroom assignment](#)

[Classroom statistics](#)

[Distribute the hours and create the lessons](#)

[Ad-hoc changes](#)

15.6 Scenarios — Exams and events

Source: [help/en/use-cases/exams-and-events.md](#) · id: [use-cases.exams-and-events](#) · Updated: 2026-06-13

Creating specific lessons: exams (in-person or remote, special configurations), competitive exams, maintenance operations.

For the reading format and the status legend, see [Scenarios — overview](#).

Create an event (competitive exam)

As a planner, I want to create an event (competitive exam), so that rooms are booked and statistics are available.

Status: ✓ Available

Solution in Omniscol. Create a dated [one-off event](#), with its rooms, participants and resources: competing occupancies of a room, a class or a teacher are flagged as conflicts as soon as the event involves the participants concerned. One-off events require a **Premium** account.

Create a specific exam lesson (every other seat, extra-time arrangements, etc.)

As a planner, I want to create a specific exam lesson (every other seat, extra-time arrangements, etc.), so that rooms are booked and statistics are available.

Status: ✓ Available

Solution in Omniscol. Create a dedicated lesson or event with the rooms and duration suited to the exam's configuration.

[Multi-room exams](#)

Create an event (maintenance)

As a planner, I want to create an event (maintenance), **so that** rooms are booked and statistics are available.

Status: ✓ Available

Solution in Omniscol. Create an [event](#) on the time slot concerned and add the room or resource to take out of service: it is then occupied on that slot and removed from the availabilities.

Create an exam (remote or in-person, 1h30 or 3h and other durations, elective or mandatory course)

As a planner, I want to create an exam (remote or in-person, 1h30 or 3h and other durations, elective or mandatory course), **so that** the exam sessions are scheduled.

Status: ✓ Available

Solution in Omniscol. An exam is created as a lesson of type **exam** (see [Types of course](#)): free duration (1h30, 3h...), in person or remote depending on its [modality](#), in one or [several rooms](#). The elective nature is modeled with a [group](#) rather than the whole class.

See also

[Scenarios — overview](#)

[One-off events](#)

[Off-grid lessons](#)

15.7 Scenarios — Absences and substitutions

Source: help/en/use-cases/absences-and-substitutions.md · id: use-cases.absences-and-substitutions · Updated: 2026-06-13

Declaring absences and managing teacher substitutions over short or long periods.

For the reading format and the status legend, see [Scenarios — overview](#).

Declare a teacher's absence

As the school administration office, **I want** to declare a teacher's absence, **so that** the affected lessons are flagged and the necessary substitutions can be organized.

Status: ✓ Available

Solution in Omniscol. Declare the absence from the **Absence management** module (the teacher can also declare it themselves, subject to validation). Omniscol identifies the affected lessons and lets you organize the substitutions.

[Declaring an absence](#)

Replace one teacher with another over a given period

As the school administration office, **I want** to replace one teacher with another over a given period, **so that** structural changes (illness, departure...) are handled.

Status: ✓ Available

Solution in Omniscol. Declare a multi-day absence with a start date and an end date: Omniscol computes the affected lessons, which can be left without a substitute, covered by a substitution policy or adjusted with one-off substitutions.

[Multi-day absences](#), [Substitution policies](#)

See also

[Scenarios — overview](#)

[Overview of the Absences module](#)

[Declaring an absence](#)

[Substitution policies](#)

15.8 Scenarios — Reporting and statistics

Source: [help/en/use-cases/reporting-and-stats.md](#) · id: [use-cases.reporting-and-stats](#) · Updated: 2026-06-13

Occupancy statistics, teaching load control, remote/in-person ratios, hour volumes per group and subject.

For the reading format and the status legend, see [Scenarios — overview](#).

View classroom occupancy pressure

As a planner, I want to view classroom occupancy pressure, so that I can anticipate saturation points.

Status: ✓ Available

Solution in Omniscol. The **classroom statistics** of the **Dashboard** (see [Classroom statistics](#)) give the occupancy rates per classroom and per period; the **Timetable** module shows the occupancy schedule of each classroom (occupied time slots).

Check the number of hours taught by a teacher

As an academic coordinator, I want to check the number of hours taught by a teacher, so that I can control workload and compliance.

Status: ⦿ Partial

Solution in Omniscol. While editing a timetable in the **Timetable management** module, the teacher's total hours are displayed and updated live; the **Dashboard** provides the total hours over the period.

Workaround. To check a specific semester, select the corresponding date range in the **Dashboard**: the total hours are recalculated over the chosen period, with no manual subtraction.

Extract activity statistics (hours, occupancy, distribution)

As a quality manager, I want to extract activity statistics (hours, occupancy, distribution), so that I can meet the requirements of a quality or certification audit.

Status: ⦿ Partial

Solution in Omniscol. The exports of the **Dashboard** and of the schedule views provide the available volumes. On their own, they do not constitute a complete certification report: the business and documentary framing remains the institution's responsibility.

COUNTRY: FR

In France, these exports notably feed the **Qualiopi** follow-up.

Analyze classroom and resource occupancy

As school management, I want to analyze classroom and resource occupancy, so that I can optimize the use of the facilities.

Status: ✓ Available

Solution in Omniscol. The **Dashboard** gives the occupancy of classrooms and resources; the exports include these volumes.

View the remote / in-person ratio per program/cohort

As a planner, I want to view the remote / in-person ratio per program/cohort, so that I can apply the rules (max 20% remote, min 20% undesirable Saturday time slots).

Status: 🚫 Partial

PREMIUM

Solution in Omniscol. The **Dashboard** provides breakdowns by **Modality** on classes and subjects when lessons carry a modality (see [Editing a lesson](#)) — **In person**, **Remote**, **Hybrid** or **Self study**. Omniscol does not provide automatic regulatory checking of thresholds (for example "20% remote maximum").

Track the planned hour volume and what remains to be planned

As an academic coordinator, I want to track the planned hour volume and what remains to be planned, so that I can check consistency with the curriculum plan.

Status: ✓ Available

Solution in Omniscol. During editing, the [hours distribution](#) shows live, subject by subject, the volume already placed and the **time left to place**. The **Dashboard** provides the follow-up afterwards, per class, subject and group.

🔗 See also

[Scenarios — overview](#)

[Dashboard](#)

[Overview of the Dashboard module](#)

[Distribute the hours and create the lessons](#)

15.9 Scenarios — Distribution and sharing

Source: [help/en/use-cases/diffusion-and-sharing.md](#) · id: [use-cases.diffusion-and-sharing](#) · Updated: 2026-06-13

Data export, iCal subscription for instructors and learners, read-only public link, resource booking.

For the reading format and the status legend, see [Scenarios — overview](#).

Export timetable data

As the school administration office, I want to export timetable data, so that it can be reused in other tools (ERP, reporting...).

Status: ✓ Available

Solution in Omniscol. The timetable views and the Dashboard offer exports; the Omniscol API lets you automate data retrieval.

[Print and share, Omniscol API](#)

View your timetable via a secure web link without authentication

As a teacher, I want to view my timetable via a secure web link without authentication, **so that** I can easily access my lessons.

Status: ✓ Available

Solution in Omniscol. The share link is signed and grants read-only access, with no sign-in; it can carry an expiration date and remains tied to the account that created it (deactivating that account or changing its password invalidates the link).

[Share a timetable via a public link](#)

Subscribe to your timetable via an iCal feed

As a teacher or student, I want to subscribe to my timetable via an iCal feed, **so that** the schedule is automatically integrated into my personal calendar.

Status: ✓ Available

Solution in Omniscol. Every user can subscribe to their timetable via a dynamic iCal link, compatible with the usual calendar apps (Google Calendar, Outlook, Apple Calendar); the calendar updates at the pace of its own synchronizations.

[iCal — subscription and dynamic link](#)

Export a teacher's or a group's timetable in Excel format

As the school administration office, I want to export a teacher's or a group's timetable in Excel format, **so that** the data can be shared or reprocessed.

Status: ✓ Available

Solution in Omniscol. See [Print and share](#).

Distribute timetables via a controlled public link

As an institution, I want to distribute timetables via a controlled public link, **so that** communication with instructors and learners is simple.

Status: ✓ Available

Solution in Omniscol. Public share links are signed, read-only and **carry an expiration date** (by default the end of the school year, or +90 days, adjustable). For institutional distribution, create the links from an identified service account, responsible for the sharing, rather than from a personal account.

[Public share links](#)

Book a resource (room, instructor, lesson, or other)

As a user, I want to book a resource (room, instructor, lesson, or other), **so that** rooms are booked and statistics are available.

Status: 🕒 Partial

Solution in Omniscol. No dedicated booking module exists today.

Workaround (Premium accounts). To block a resource on a specific time slot, create an [event](#); to make it unavailable over a whole **period**, use a [date window](#).

🔗 See also

[Scenarios — overview](#)

[iCal — subscription and dynamic link](#)

[Sharing link](#)

[Public share links](#)

[Print and share](#)

16. FAQ — frequently asked questions

16.1 FAQ — general questions

Source: <help/en/faq/general.md> · id: [faq.general](#) · Updated: 2026-06-13

The general questions most frequently asked by prospects and new users. For more specific questions, see the specialized FAQs: [Timetable creation](#), [Data import](#), [Algorithm behavior](#), [Display and UX](#), [Security and hosting](#), [Pricing and licenses](#).

Does Omniscol work with a limited internet connection?

Yes, provided you have a sufficient connection at the time of loads and saves. The web application caches already-consulted data in the browser, which makes consultation more tolerant of short outages. Changes, imports, exports and administration operations, on the other hand, require network access available at the time of the action.

Which browsers are supported?

Chrome is recommended. Firefox, Safari and Edge also work. Internet Explorer is not supported.

Do I need to install a client?

No. Everything runs in the browser. No local installation is required.

Is my data backed up?

Yes. Omniscol relies on backups and redundancy mechanisms described in the contractual framework and the account's security documents. See [Security and hosting](#).

For manual backups, a **full export in JSON format** is available from Import/Export. In case of a major problem, this file can be sent back to the Omniscol team for re-injection.

Can I have several timetables at the same time?

Yes. On a Standard account, over disjoint week ranges (for example semester 1 vs semester 2). On a [Premium](#) account, several timetables can be published **simultaneously** over the same weeks, with dynamic merging of the views — this is included by default. This capability can also be enabled contractually on some Standard accounts (exceptional case, with an adapted framework). See [Multiple active timetables in parallel](#).

How many administrators can work at the same time?

The number of simultaneous sessions depends on your contract and your usage. With the real-time Collaboration option, administrators see the presence of the other connected users (a red disc signals a risk of simultaneous modification). The working views are not always refreshed automatically after a save made by a colleague. See [Collaboration between administrators](#).

How to share a timetable with someone who has no account?

The  Sharing button on the timetable screen generates a read-only [share link](#), with a mandatory expiration date.

For a dynamic calendar subscription (automatic synchronization), the iCal option is available in the share. See [iCal — subscription and dynamic link](#).

Which languages are supported?

To date: French, English, German, Spanish, Italian, Portuguese, Dutch, Polish, Russian, Simplified Chinese, Arabic, Hebrew, Kazakh, Vietnamese. Right-to-left languages (Arabic, Hebrew) benefit from an adapted interface (RTL).

My school is in a country that is not listed. Can I use Omniscol?

Yes. The settings prefilled per country are a convenience, not a constraint. In the least favorable case, you manually customize the levels, the time grid, the holidays and the subjects.

How do I retrieve my data if I leave Omniscol?

Full JSON export from Import/Export. It includes all the school's data, in a reusable format. You remain the owner of your data, in accordance with the GDPR.

What is the difference between the Timetable and Timetable management modules?

- **Timetable management** is used to **create, configure, generate and publish** timetables. Occasional use (before the school year starts, structural changes).
- **Timetable** is used to **consult and occasionally modify** the published timetables. Daily use.

See [Overview of the Timetable management module](#) and [Viewing and filtering](#).

See also

[Timetable creation](#)[Data import](#)[Algorithm behavior](#)[Security and hosting](#)[FAQ — higher education use cases](#)[FAQ — special cases and advanced configurations](#)

16.2 FAQ — Timetable creation

Source: [help/en/faq/timetables.md](#) · id: [faq.timetables](#) · Updated: 2026-06-25

Frequently asked questions about creating and configuring timetables. For the details of each step, see [Overview of the Timetable management module](#).

Which timetable type to choose: weekly, cyclic or calendar?

The choice depends on your school's rhythm:

- **Weekly** — recurring lessons over a typical week. Suited to primary and secondary education.
- **Cyclic** — recurring lessons over an N-day cycle, different from the 5- or 7-day week (typical of North American systems).
- **Calendar** — individually dated lessons, with no recurrence (typical of higher education and continuing education). Available on Premium accounts.

Automatic generation works for all timetable types.

See [Choosing the right timetable type](#).

Can several timetables be created in parallel?

Yes. Several timetables can coexist in the same account (drafts, versions, semesters, periods). On a Standard account, publications must cover disjoint week ranges. On a Premium account, several timetables can be published **simultaneously** on the same weeks, with dynamic merging — the same capability can be enabled by contract on some Standard accounts. See [Multiple active timetables in parallel](#).

What is the recommended order for configuring a timetable?

The order of the **steps**: General, Sites, Teachers, Classes (with their groups), Group alignment, Hours distribution, Generation, then Publication. The first seven are the tabs of the timetable editor (from left to right); **publication** happens from the timetable management screen. Each step is documented in detail in the "Creating a timetable" section of the table of contents.

Does automatic generation have limits?

The solver is powerful but strictly respects the constraints it is given. If a configuration is structurally impossible (contradictory availability, insufficient capacity, inconsistent alignments), Omniscol returns the best computed timetable, leaves the missing lessons in the list of unplaced sticky notes and provides an explicit diagnostic. See [Diagnosing a failed generation](#).

How long does a generation take?

The duration depends on the size of the timetable, the number of lessons and the constraints. Each generation starts a dedicated, parallelized computing environment; initialization often takes around ten seconds before the actual computation. The window can be closed during the computation; a notification appears at the end. For the orders of magnitude, see [Algorithm behavior](#).

Is a generated timetable published automatically?

No. Publication ("timetable allocation") is a separate step that must be performed explicitly after review. Without publication, the timetable remains a draft that end users cannot see. See [Publishing \(activating\) a timetable](#).

Can a timetable be restored to an earlier state?

Yes, in two forms:

- **Duplicating** a timetable — each structural change can be made on a copy, with the original serving as a reference (available on all tiers).
- **Snapshots** — restorable backup points of the entire account, automated or manual (Snapshots option, depending on the contract). See [Backup points](#).

How to manage alternate weeks (A/B)?

The feature is enabled in the [General settings](#), under the **Alternate weeks** section (letters A/B/C or digits 1/2/3, with any cadence over as many weeks as needed). On a course, [+ Add week](#) adds a range for a new alternate week. See [Alternate lessons](#).

See also

[Overview of the Timetable management module](#)[Diagnosing a failed generation](#)[Choosing the right timetable type](#)[Algorithm behavior](#)[FAQ — special cases and advanced configurations](#)

16.3 FAQ — Data import

Source: [help/en/faq/data-import.md](#) · id: [faq.data-import](#) · Updated: 2026-05-10

Which import formats are supported?

CSV and TSV, by copy-paste from Excel, Google Sheets, Numbers or Calc. The full **JSON export** of the account is available to the administrator (backup); **re-importing** a full JSON account, on the other hand, is a restore operation carried out by Omniscol, not a routine action of the administration screen.

For dedicated migrations, see [Overview](#).

Is the identifier generated by the system?

Yes. When importing students, teachers or classes, Omniscol generates a stable internal identifier: for people it derives from the first and last name, for a group it includes the parent class (`<classe>:<groupe>`). The pattern of the **login identifier** (first name-last name, last name-first name or identification number) is configured in [General settings](#), **Identifier** field.

You can also record your own reference via the **external identifier / registration number** field — useful for traceability with your student information system and for telling apart people who share the same name.

For groups, must the class always be referenced?

Yes. A group is always a subdivision of a class. A group's ID includes the parent class. In lesson import files, the "class" and "group" columns work together.

See [Class, group, subgroup](#).

Does the column order of the table matter?

The import screen opens a **column template** typed by position (the subject, the teacher, the room... each occupies an expected column). You **rearrange the template's columns** so that they match the order of your source spreadsheet: the copy-paste then stays clean, without rewriting your file. See [Mass import of courses from a spreadsheet](#).

Which separators for lists (multi-teacher, multi-group, etc.)?

Common separators are recognized inside a single field: comma, semicolon, slash, vertical bar, plus sign, ampersand or line break. The import engine detects the convention used in the field.

What happens if the table contains unknown subjects?

At step 4 of the import, Omniscol offers to **create the unknown entities**. For custom subjects and some teachers, you can request creation on the **Administration** side to make them available throughout the school. Labels are corrected at step 3, during disambiguation.

Are complex lessons imported correctly?

Partially. The import engine handles well:

- simple lessons (subject + class + teacher + room + time slot),
- basic alternate lessons (\bar{A}/B),
- multi-teacher lessons (co-teaching),
- multi-room lessons.

It handles the following less well, so they require manual finishing:

- associated lessons (alternating half-groups),
- complex concatenations,
- groups of groups,
- off-grid lessons with precise times.

What to do if the import fails on a large file?

- Check for empty rows or merged cells.
- Split the file into subsets (by class, by campus, by semester).
- Prefer a **full export** of the source account, often cleaner than a manual partial export.

See also

[Preparing your data for a mass import](#)

[Mass import of courses from a spreadsheet](#)

[Overview](#)

[Higher education use cases](#)

16.4 FAQ — Generation algorithm behavior

Source: <help/en/faq/solver-behavior.md> · id: [faq.solver-behavior](#) · Updated: 2026-05-15


Frequently asked questions about the behavior of the automatic generation engine. For a description of the solver itself, see [Solver](#); for the operational details, see [Automatic generation](#).

Which timetable types does the solver work on?

On [weekly](#), [cyclic](#) and [calendar](#) timetables. In calendar mode, lessons are placed on real dates and the generation options include date bounds.

How does the algorithm work?

Omniscol runs a solving process in the background. The solver evaluates the lessons to place, the declared constraints and the generation preferences. The result depends above all on data quality: hour volumes, teachers, rooms, groups, availability, incompatibilities and inter-site constraints.

Each generation starts a dedicated, parallelized computing environment. There is no queue to manage on the school side; initialization often takes around ten seconds before the actual computation begins. In the **Timetable management** module, the  button starts the computation, tracks its status and lets you inspect the result.

The engine is a **neuro-symbolic Monte-Carlo metaheuristic** optimization AI: stochastic search, symbolic constraints and progressive score optimization. It does not promise success on an impossible set of constraints; in that case, Omniscol returns a diagnostic.

How long does a generation take?

The duration depends on the size of the timetable, the number of lessons, the number of constraints and the chosen options. As an order of magnitude, a middle school with around 550 students, 45 teachers and 16 classes, with plenty of availability, can obtain a complete timetable in under a minute, with very few or no gaps, when the data is consistent. A denser, highly constrained case, or one with costly optimizations, can take significantly longer.

Does the solver respect all constraints?

Hard constraints are strictly respected:

- a teacher in only one place at a time;
- a class without double placement, except for groups in a division;
- availability and time slots marked **Unavailable**;
- a standard room occupied by only one lesson at a time;
- a large room used only within its capacity and class-count limits;
- compatible specialized rooms, capacities, material resources and inter-site travel;
- locked lessons kept in place.

Soft constraints (undesirable availability, preferences, pedagogical weighting, day balance, number of attendance days, gaps in timetables) are optimized as well as possible. They create penalties: the solver looks for the solution that accumulates the fewest, but may keep some if that is the best compromise compatible with the hard constraints.

What if the generation fails?

When no complete solution is found, Omniscol returns the best computed timetable and leaves the unplaced lessons in the sticky notes bar. The partial timetable remains available to understand what was placed, then fix the constraints or reposition some lessons manually. The most frequent causes are described on the page [Diagnosing a failed generation](#).

Can the position of a lesson be forced manually?

Yes, by **locking** a lesson after manual placement. The solver does not move a locked lesson during a new generation, but adjusts the others around it. Useful for anchoring immovable lessons (external speakers on fixed dates, dated exams).

Why doesn't the solver offer several solutions?


The solver optimizes according to the declared constraints and preferences and returns **the best solution found**. To obtain several solutions to compare, duplicate the timetable before each generation and slightly adjust the constraints or weightings between runs.

Does the solver take declared absences into account?

The solver works on the **structural** timetable, independently of absences. Absences are handled **afterwards** in the **Absence management** module (see [Overview of the Absences module](#)), affecting the display of the published timetable and the statistics of the [Dashboard](#) (not the underlying structure).

Deliberate conflicts and blocking conflicts

Not all conflicts are equivalent:

- **Non-blocking conflicts** — can be left as they are if this is a deliberate choice (for example, a room whose displayed headcount exceeds its capacity because you know that not all enrolled students will attend). The alert stays visible but does not block.
- **Blocking conflicts** — prevent the automatic generation from succeeding (no compatible room, inconsistent alignment, etc.). The  button stays disabled as long as these conflicts remain.

See [Conflict](#).

See also

[Solver](#)[Automatic generation](#)[Diagnosing a failed generation](#)[Conflict](#)[Diagnostic](#)

16.5 FAQ — Display and interface

Source: [help/en/faq/display-and-ux.md](#) · id: [faq.display-and-ux](#) · Updated: 2026-06-25

Frequently asked questions about display, views, and the user experience of the Omniscol interface.

Which views are available to consult a timetable?

The display component combines:

- **rendering modes:** **grid** (standard calendar), **list**, spreadsheet-style **table**;
- **reading variants:** **day**, **month**, **schedule** (and **hourly schedule**), **side-by-side**.

The schedule view crosses several entities or several days in a single view; the hourly schedule does the same on an hourly scale. The table is a tabular rendering of the current view.

See [Timetable display](#) and [Viewing and filtering](#).

Is a vertical timetable display available?

In the **Timetable** module, the timetable display component offers no vertical view. An equivalent vertical view exists in the **Staffing** module (see [Overview of the Staffing module](#)) for related contexts (supervision grids).

Is the interface responsive (mobile, tablet)?

Yes, the interface adapts to screens of all sizes. On a smartphone, a horizontal swipe scrolls through the displayed period (the week by default, the day in the Day view).

Which name display settings are available?

Names go through a formatting function driven by the school's configuration:

- First name/last name order (following the country's conventions).
- Bolding of the main name (varies by country — handling of the *middle name* for American, Vietnamese and Arabic conventions).
- Capitalization of the first name, of the middle name if enabled, and/or of the last name via the name capitalization setting.


Is the interface accessible to people with disabilities?

An accessibility audit (RGAA, WCAG) is in place. A conformance report can be provided on request. Identified improvements are handled continuously. For specific needs (screen reader, high contrast, keyboard navigation), contact support.



Are right-to-left languages supported?

Yes. Arabic and Hebrew have an adapted right-to-left interface. The 14 supported languages are listed in [General questions](#).

How to print a timetable?

The  **Print** button in the **Timetable** module starts printing. The rendering is optimized for paper (no interactive elements, adapted layout).


How to export a timetable to Excel or an external calendar?

- Excel / CSV export via  **Table**.
- iCal export / subscription via  **Download** (see [iCal — subscription and dynamic link](#)).
- Public read-only share link (see [Sharing link](#)).

Is dark mode available?

Not in the current interface: Omniscol offers no dark theme and does not switch automatically with the operating system's dark setting. The interface does, however, take into account the system's accessibility preferences such as high contrast.

Can I customize subject colors?

Yes. Each subject (see [Subject](#)) can be given a specific color via  **Colorpicker**. Course types remain plain text labels, with no associated color or icon.

See also

[Viewing and filtering](#)

[FAQ — special cases and advanced configurations](#)

[Display panel](#)

[Print and share](#)

16.6 **FAQ — higher education use cases**

Source: [help/en/faq/higher-ed-cases.md](#) · id: [faq.higher-ed-cases](#) · Updated: 2026-06-13

Frequently asked questions from higher-education institutions (business schools, engineering schools, art and design schools, universities, continuing-education centers). The answers apply across these institution types; for the domain specifics see [Higher education specifics](#).

Multiple timetables, each user sees only their own

Q. How can several timetables be managed in parallel so that each user sees, in their calendar, only the lessons that concern them?

A. Three mechanisms work together:

- On the administrator side, several timetables can be published in parallel over the same weeks, with dynamic merging — included by default with a Premium account (see [Multiple active timetables in parallel](#)).
- By default, each user only sees the lessons that concern them — their class, their groups, or their own lessons for a teacher. Filters are applied automatically at login according to the user's role.
- To view another scope (another class, another teacher...), the user applies a filter manually, provided their role allows it. The [visibility restrictions](#) make it possible, for example, to prevent teachers from seeing their colleagues' timetables.

iCal synchronization after a timetable change

Q. When a lesson is changed in Omniscol, is the iCal feed updated automatically?

A. Yes, provided you use an **iCal subscription link** (not a static download). The subscription link is dynamic; the calendar application re-fetches the feed at regular intervals, with a delay ranging from a few minutes to several hours depending on the application and its configuration. The calendar application decides the refresh rate, not Omniscol.

See [iCal](#) and [iCal — subscription and dynamic link](#).

Notification on every change

Q. Can a notification be sent for every timetable change?

A. No automatic notification on every change: the volume would be unmanageable for recipients. Current approach: publish changes in batches via a summary email sent to a selection of the affected users ([✉ Send email](#)).

Room within a macro-type / room category

Q. How can you indicate that a course must take place in a particular room belonging to a macro-type — depending on whether you want to force a specific room or let the engine choose within the category?

A. Via [classroom specializations](#), which are free-form fields:

- Create a specialization matching the macro-type (for example "WORKSHOP", "IT-ROOM", "CHEM-LAB").
- Assign this specialization to the rooms concerned.
- On the course, indicate the required specialization. Omniscol suggests the specialized rooms first and the solver picks one of those available.
- To **force a specific room**, indicate it directly (in addition to the specialization, or instead of it). The solver honors the forced room.

For bulk imports, you can indicate only the specialization; the solver takes care of the final assignment.

Checking room occupancy before generation

Q. Is there a report to check, before generation, whether the hours assigned to a room or to a room macro-type fit within the total available hours?

A. The [Generation](#) tab runs a preliminary diagnostic: capacity checks (number of lessons to place compared with the capacity of the time grid) and compatibility checks on classroom specializations. A warning is raised in case of over-allocation. The post-publication [dashboard](#) gives the actual detail (taking absences and substitutions into account).

Duration of each individual course at import

Q. In the import file, the "duration" column contains the total hours of an activity, which may combine several courses. Where should the individual duration of each course be indicated?

A. An **import row corresponds to one course**. The duration entered is that of this course, not of the module. For a module containing several courses of different durations, you therefore need one import row per course.

If the source aggregates ("Algebra = 30 h total"), you must break it down before importing — or use [automatic distribution](#) to split a volume of hours into individual courses.

Courses on certain weeks only

Q. How can you indicate that a course only takes place on certain specific weeks of the academic year?

A. It depends on the mode:

- In [weekly](#) mode, [alternate weeks](#) cover the case where a course only recurs one week out of N. For a course that only falls on a few specific weeks of the year, calendar mode is more appropriate.
- In [calendar](#) mode, each lesson is placed individually on its date: this is built in.
- At import, enter the **date** of each lesson in the dedicated column (format `YYYY-MM-DD`), or leave it empty for a lesson to be placed manually later.

Cross-class / cross-cohort courses

Q. How do you model courses where, in the same time slot and the same room, several groups of one class or of different classes take part, sometimes for different subjects, sometimes on different campuses?

A. Depending on the case:

- **Several groups of the same class in the same slot, same subject, same teacher, same room** — a single course assigned to a super-group. Fixed composition → [class division](#) if applicable. Evolving composition → [group of groups](#), available on all accounts and all timetable types.
- **Several different classes in the same slot, same subject, same teacher, same room** — an [alignment](#) when the composition is fixed, or a [group of groups](#) (available on all accounts, all timetable types) when the grouping must remain flexible.
- **Different subjects in the same slot and the same room** (rare case) — two separate courses are needed. Each student sees, on their timetable, the subject of their own group.
- **Across two sites** (two classes in different locations sharing a course) — supported, provided the shared room or the videoconference is modeled correctly (see [virtual sites](#)).

Alternating instructors on the same time slot

Q. How can you indicate that two teachers alternate on the same time slot, the same day of the week, with the same class / subject / group — one teacher one week, the other the next?

A. This is an **instructor alternation**, not [co-teaching](#) in the strict sense (which implies two teachers simultaneously on the same lesson). Two implementations depending on the cadence:

- **Regular alternation in weekly mode** — use [alternate weeks](#) with a different teacher on each week.
- **Alternation on specific dates in calendar mode** — create one course per lesson with the corresponding teacher. More verbose but more flexible: each lesson can be adjusted individually.

One-off co-teaching (visiting professor)

Q. How can you indicate that, for one or a few lessons only, a visiting professor co-teaches a course with the main teacher?

A. In [calendar](#) mode, create the ordinary lessons with the main teacher only, then for the lessons where the visiting professor takes part, add them as a [co-teacher](#) on the specific lesson. Each teacher is credited in their own statistics.

Videoconference link per course

Q. Can the videoconference link only be attached to a whole class, or can it be attached to a specific course or lesson?

A. A videoconference link can be defined at three levels: on the **class** (it then applies by default to all its lessons), on a specific **course**, or on an **individual lesson** in calendar mode. This suits hybrid sessions where only some courses are delivered remotely. See [Videoconference links per course](#).

Students repeating a year and off-track students

Q. How do you manage a student who must attend a subject from another year (repeating a year, catch-up, *fuori corso*, etc.)?

A. A student can be assigned to several classes simultaneously, and restricted to certain groups of a class. Four typical cases:

1. **Another entire class + a specific group of a third one** — assign to both and pick the group.
2. **Several groups of another class without the whole class** — assign directly to the groups.
3. **A single subject from another class via a specific group** — create a dedicated group in the other class and assign the student to that group only.
4. **Off-track student retaking an isolated subject** — same logic with a dedicated group.

SIS ID as unique identifier

Q. Can the student or teacher identifier coming from our SIS be used as the unique identifier in Omniscol, to avoid confusion between people with the same name?

A. Yes, via the **Registration number / external identifier** field on the user record — that is exactly what it is for. Fill it in at import and Omniscol will use it as the identification key. Particularly useful for two-way synchronizations with your SIS (see [synchronization with external systems](#) for Aurion / Auriga or [Omniscol API](#) for a custom integration).

Several email addresses per user

Q. Can two email addresses be imported for the same teacher (institutional and personal)?

A. Not in the current interface: one main email address per user. If a second address must be kept for information purposes, use the Comment field of the user record. This field is not used as a sending address.

Learners, students, participants — which entity?

Q. Our reference model distinguishes **learner** (general term), **student** (initial education), **participant** (continuing education). How do these nuances map?

A. All map to the Omniscol **student** entity (role `student`). The distinction between initial and continuing education is handled by an optional attribute on the record (SIS registration number, status), not by separate entities.

Instructors: permanent, adjunct, expert, visiting

Q. Our reference model distinguishes **permanent teacher** (permanent contract, distributed time), **adjunct instructor** (one-off contract), **expert** (occasional external provider), **visiting professor** (invited academic). Do they all correspond to the same Omniscol entity?

A. Yes, all map to the **teacher** entity (role `teacher`). The **per-teacher "external / adjunct" marker** that distinguishes them from permanent staff is a **Premium** option (see [external teacher](#)); without Premium, the four profiles remain a single teacher entity, without a dedicated marker.

The "course leader" and "program director" roles are **business roles** internal to your institution. Omniscol does not model them as such, but can represent them through the [custom roles](#).

Programs, curricula, study tracks

Q. Our offering is organized into **programs** (PGE, MBA, MSc, PhD, etc.), **management courses** (UE / EC / ECUE), **teaching activities** (sessions). How does this map into Omniscol?

A. Recommended mapping:

Business concept	Omniscol entity
Program	A class (or a set of classes for parallel cohorts)
Course / module / UE / EC	A subject (with or without a lesson type)
Session / teaching activity	A lesson — an individually placed occurrence of a course

For hierarchical nomenclatures (UE → EC → ECUE), a naming convention on subjects ("UE3-EC2-ECUE1 Linear algebra") or the use of **subject groups** can represent the grouping.

Mobility, exchanges, double degrees

Q. How do you manage incoming and outgoing mobility (academic exchanges, double degrees, *free movers*, *study abroad*)?

A. On the timetable side, these learners are **ordinary students assigned to specific classes**. The administrative management (agreements, credits, ECTS, status) belongs to the institution's SIS; Omniscol focuses on the question of "who attends which course, when, where".

Configuration recommendations for higher education

For a higher-education account, consider:

- **Premium account:** calendar mode, calendar availability and multiple active timetables are included by default, as is MCP (connecting a compatible AI assistant). Complement as needed with the contractual options: custom roles, snapshots, real-time collaboration. (**Groups of groups**, by contrast, are available on all accounts.)
- **Calendar mode** for most timetables (individually dated lessons, instructors varying from session to session).
- **Multiple active timetables** in parallel per program (one timetable per master's program, merged dynamically when courses are shared).
- **Groups of groups** for evolving cross-program groupings (multi-track common core).
- **Multi-room** for exams split across several lecture halls or for broadcast lectures.
- **Custom roles** to model course or program leaders with permissions targeted to their scope.
- **SIS integration** ([synchronization with external systems](#) for Aurion / Auriga, or [Omniscol API](#) for a custom integration) to synchronize learners.
- **OIDC / SSO** with the institution's identity provider (Google Workspace, Microsoft Entra ID, Keycloak, etc.).

See also

[Higher education specifics — overview](#)

[Multi-room](#)

[Modality](#)

[Co-teaching](#)

[Calendar mode](#)

[Groups of groups](#)

[FAQ — Timetable creation](#)

[FAQ — Data import](#)


16.7 FAQ — special cases and advanced configurations

Source: [help/en/faq/edge-cases.md](#) · id: [faq.edge-cases](#) · Updated: 2026-06-13

Questions about atypical configurations or special cases: lessons that do not follow the time grid, fine-grained management of rooms and their specialisations, non-blocking conflicts, specific displays, traceability of inconsistencies.

Lessons with a custom time

Q. Part of the lessons in our school (about 20%) do not follow the standard time grid: precise exam durations, shifted times for some practicals. How to place a lesson that must start at 1:10 pm on a grid with a 10-minute step, and get a consistent end time?

A. Yes. A lesson can be entered with its own start and end time, without exactly matching the boundaries of the standard grid. In the placed lesson, use the  Custom time button (Custom time) then fill in the start and the end.

The lesson is still taken into account in conflicts: teacher, room, class, group and resource are considered busy as soon as the custom time overlaps a time slot. For automatic generation, this position is locked.

The **Off-grid** configuration of a class is a different case: it is meant for calendar-mode classes configured off-grid whose every lesson must be stored with precise times.

See [Off-grid lessons](#) for the details.

Creating subjects with an existing parent and family

Q. When creating a custom subject, is it possible to select an already created **parent subject** and an existing **family**, without having to go through an external re-import?

A. Yes. The creation or edit window of a custom subject lets you directly select an existing parent subject (Parent subject field) and an existing family (field Family), without an external re-import. See [Managing subjects](#).

Editing or deleting a room specialisation

Q. Why is directly editing or deleting a room specialisation not available (for example to fix a spelling mistake)?

A. The list of specialisations is computed **dynamically** from the rooms. To make a specialisation disappear:

- Remove it from all the rooms that carry it. On the next save, it disappears from the repository automatically.
- To **rename** it, create the new specialisation with the right label, replace it on the rooms concerned; the old one then disappears on its own.

The interface therefore offers no button for directly editing or deleting a specialisation. See [Classroom specialisation](#).

Timetable filter by subject: plain subject vs subject with type

Q. When I filter timetables by subject, some lessons do not appear even though they are clearly visible when I filter by class. What is the cause?

A. The **(subject, course type)** pair is treated as an entry independent of the subject alone. Consequences:

- "Algebra" without a type and "Algebra — Tutorial" correspond to **two different entries** in the timetable.
- To filter on both at once, do not specify a type in the filter.
- If you need to audit all the types of a single subject, use the filters available in the view then check the results in the course list.


Displaying the last name in uppercase

Q. Is it possible to display last names in uppercase? Useful in particular when the last name is also a common first name (for example "Martin Pierre" vs "Martin PIERRE").

A. Yes. The name capitalization setting is configured in the school settings. It can apply to the first name, to the middle name if the option is enabled, and/or to the last name according to the name order configured for the school.


Lesson lock that does not appear in every view

Q. The lock on a lesson's position is not displayed systematically when switching views (sticky notes ↔ list).

A. The lock marker depends on the view and on the nature of the lesson. In the distribution views and some course rows, the lock is displayed as a  icon. If a view does not show it, check the lesson's details before concluding that the algorithm can modify it.

Room assignment: favorite site and priority specialisations

Q. When assigning a room, the rooms of another site appear first even though the favorite site is checked. Why?

A. The sort uses by default the **site attached to the class**, not the user's favorite site. This is a deliberate behavior: the class drives the choice of the room. If the proposed order is not the expected one, check that the class is attached to the intended site in  `Edit`.

For **specialisations**, review the proposed rooms and their compatibility warnings in the selector. Do not assume a sort that prioritizes best compatibility.

Total capacity = sum of the rooms (multi-room)

Q. When several rooms are assigned to the same lesson, does Omniscol compute the **total capacity** as the sum of the individual capacities, thereby avoiding false over-capacity alerts?

A. Yes. The total capacity is computed as the **sum** of the capacities of the assigned rooms. If the group fits within the sum, no alert is raised. This feature is available **on all timetable types**: weekly, cyclic, calendar.

If the sum remains below the group, the conflict is reported. The administrator can then:

- add an extra room,
- reduce the group,
- accept the conflict if it is deliberate (typical case: you know that not all enrolled students will attend).

See [Multi-room](#).

Detecting inconsistencies of groups in a class division


Q. Omniscol reports "Different number of lessons for groups in class division" without specifying in which subjects the inconsistencies are. How to investigate in detail?

A. The **advanced statistics** panel of the hours distribution screen allows detailed inspection, notably by isolating the groups that interact on the affected time slots.

For schools where imbalances are numerous and intentional, a heuristic limits the display of alerts beyond a threshold to avoid an unmanageable list. It remains possible to export the courses and filter them in a spreadsheet for an exhaustive diagnosis.

Several comments with different restrictions

Q. On the same lesson, is it possible to add two comments with distinct publication restrictions (one for administrators only, the other for teachers)?

A. Yes. Each comment has its own publication restriction: administrators, teachers, or all users. Use the  `Comment` button to add several comments and set their visibility separately.

Vertical timetable display (weekly load)

Q. To visualize the load of a group or the occupancy of a type of rooms over a full week, the horizontal display forces you to scroll. Is a vertical display available?

A. In the **Timetable** module, timetables are displayed horizontally. An equivalent vertical view exists in the **Staffing** module for related contexts (supervision grids). See [Overview of the Staffing module](#) and [Display and UX](#).

Statistics: placed lessons vs created courses

Q. The counter on the "hours distribution" screen shows how many courses are created. Where can I see how many hours are actually **placed** in the timetable (vs created without placement)?


A. Two screens give two complementary measures:

- The **Dashboard** works on the **operational timetable** (lessons actually placed, possible merge of several published timetables, absences taken into account). The displayed percentages are relative ("93% of this class's hours"), not a placement rate.
- The **hours distribution** screen measures declared courses and the created volumes. If you need the placement rate, review the courses in the schedule view or export the list to filter the lessons without a position.

Adjunct / external teacher visible on the timetable

Q. Does the icon that distinguishes adjunct teachers from permanent ones appear next to the teacher's name in the timetable views?

A. Depending on the view:

- **Grid view** — space is limited and names may be abbreviated or truncated there as plain text; the icon does not appear there systematically.
- **Lists, tooltips and selectors** — wherever the full name is rendered, a  icon precedes the name of the teachers whose record carries the External teacher marker.

This "external" marker is a **Premium** option; without it, no icon distinguishes adjunct teachers. See [External teacher](#).

PREMIUM

One-off events: frequently requested details

A few details about the events screen (see [One-off events](#)):

- **Proposed times** — the time range of an event follows the Opening hours of the site; if it is too short (an evening event for example), widen these hours in the site's record.
- **Changing the time in place** — the time of an event is changed directly, with a click on the event.
- **Participants** — the participant list offers classes and groups, as well as teachers and students individually.

See also

[Off-grid lessons](#)

[One-off events](#)

[Multi-room](#)

[Classroom specialisation](#)

[Display and UX](#)

[Algorithm behavior](#)

16.8 FAQ — Security and hosting

Source: [help/en/faq/security-and-hosting.md](#) · id: [faq.security-and-hosting](#) · Updated: 2026-05-10

This page summarizes the most frequent security and hosting questions. For an exhaustive description, Omniscol provides the official documents on request (or under NDA depending on the case):

- [SECURITY_OVERVIEW.md](#) — public overview,
- [SECURITY_ONE_PAGER.md](#) — summary for CISOs,
- [SECURITY_FULL.md](#) — complete documentation (governance, infrastructure, controls, compliance, incident response),
- [ISO27001_MAPPING.md](#) — mapping to ISO 27001 controls,
- [SSI_QUESTIONNAIRE_STANDARD.md](#) — answers to the standard information security questionnaire.

For any security question: security@omniscol.com.

Where is my data hosted?

The legal notice states that hosting is located in the European Union, on AWS Paris and Scaleway. The precise commitments on redundancy, backup, retention and availability depend on the contractual framework applicable to your account.

Is Omniscol GDPR compliant?

Omniscol is designed for GDPR-compliant use. Practical consequences:

- Hosting exclusively in the EU (France).
- **Logical isolation** per customer: no cross-account access is possible.
- No **profiling**, no **resale** of data, no **secondary use** of your school's content.
- **Reversibility**: complete JSON export via [Download](#).
- Data subject requests handled according to the applicable contractual and regulatory framework.

Is Omniscol ISO 27001 certified?

Not certified to date, but **practices are aligned** with the principles of the standard and follow a risk-based, continually reviewed approach. Details of the controls can be provided under the appropriate contractual or security framework.

How are passwords protected?

Omniscol protects passwords in the following way:

- **Client-side pre-hashing** with **script** — your plaintext password never reaches the servers.
- Server-side storage as a salted hash.
- No plaintext password is ever transmitted or stored.
- No Omniscol administrator can read, retrieve or recover a user password.

If you suspect an account has been compromised, changing the password invalidates any derived access that depends on this secret, in particular share links tied to the creator's password.

Are communications encrypted?

Public application access goes through HTTPS. The network exposure configuration depends on the chosen deployment and must be checked within the security or contractual framework of the account concerned.

How does authentication work?

Omniscol sessions use signed tokens with a limited lifetime; access is renewed while the user works in the interface.

Share links and API tokens follow separate rules:

- a share link can have an expiry date and also depends on the account that created it; changing that account's password or deactivating it invalidates the related access;
- an API token depends on a key created in the interface; deleting that key or reaching its expiry date invalidates the associated tokens.

See also [Sharing link](#) and [Omniscol API](#).

API: keys and tokens

API access is managed from the interface with two visible objects:

1. **Key** — created with a label and, if needed, an expiry date. This expiry can be changed later.
2. **Token** — generated from a key, with a list of authorized API endpoints and its own expiry. This expiry cannot be changed after generation; a new token must be created.

Best practice: one key per integration, with an explicit label and an appropriate expiry. When in doubt, delete the key and create a new one.

See [Omniscol API](#).

Can my users use their corporate account (SSO)?

Yes, via [OIDC / SSO](#), when the configuration is enabled on the account. Omniscol supports Google Workspace, Microsoft Entra ID and generic OIDC providers.

Each user then signs in with their corporate identity and keeps the permissions configured in Omniscol.

What protections exist against abusive sign-in attempts?

- Failed authentication responses are deliberately slowed down.
- Administration screens have stronger protections.
- Omniscol does not reveal whether the username exists: the message stays generic.

User account lifecycle

- **Deactivation without deletion** — a deactivated account can no longer sign in, but its associated data is preserved.
- **Mandatory change on first access** — when a password is set manually, the user must change it on first sign-in.
- **Reset** — an administrator can trigger a password reset without knowing the existing password.

Network security and operations

The security documents provided on request detail the hosting, the network separation, the data access rules and the operational procedures applicable to the account concerned.

Is there an audit log?

Yes, when the logs option and the corresponding retention are enabled for the account. The visible scope depends on the logged operations and the depth configured by Omniscol. See [Activity log \(logs\)](#).


OWASP Top 10

Omniscol applies controls aligned with the OWASP Top 10 risks: access control, encryption, input validation, secure configuration, dependency monitoring, session protection and logging. The exact scope can be detailed in the security documents provided on request.

Automated security testing

Automated checks are integrated into Omniscol's development and deployment process. Their scope evolves with the product; the detailed elements can be provided under the appropriate security framework.

How to perform a manual backup?

The  **Download** button downloads a complete export of the account in JSON format. The file must be stored with care: it contains the password hashes and all personal data.

In the event of a major incident, this file can be sent back to the Omniscol team for re-import.

If the **Snapshots** option is enabled on the account, you can also automate the creation of restorable backup points. See [Backup points](#).

Incident history

As of the most recent review (February 2026):

- **0 customer data breaches** (cumulative history).
- **0 security incidents** in 2025.

These figures are published in `SECURITY_ONE_PAGER.md`, a document reviewed quarterly.

See also

[Architecture and roles](#)

[Sharing link](#)

[API token](#)

[Omniscol API](#)

[OIDC / SSO](#)

[Backups and snapshots](#)

[Activity log \(logs\)](#)

16.9 FAQ — Pricing and licenses

Source: `help/en/faq/pricing-and-licenses.md` · id: `faq.pricing-and-licenses` · Updated: 2026-06-13

Frequently asked questions about subscription plans, included features and à-la-carte options, as well as billing. For feature details by plan / option, see [Omniscol plans and options](#).

Which subscription plans exist?

Three main plans, plus a catalog of à-la-carte options:

Plan	Target audience
Lite	Very small institutions, freelancers, solo trainers, trials — timetable generation and printing only (no publication or student management)
Standard	Typical school (primary, middle or high school), single planning team, weekly timetable
Premium	Higher education, continuing education, large volumes, complex multi-site setups

Added to these are **Standard Plus** (Standard extended with the **Staffing** module) and the standalone **Staffing** offer (supervision management without the rest of Omniscol).

See [Omniscol plans and options](#) for the feature details.

Which features are included in Premium?

By default on a Premium account:

- Calendar mode — timetables dated lesson by lesson;
- Calendar availability — availability entered date by date;
- Multiple active timetables — simultaneous publications over the same weeks.

(**Groups of groups** are not a Premium feature: they are available on all accounts, whatever the plan.)

Which options can be added à la carte?

The remaining contractual options can be enabled according to the account's needs, in addition to the chosen plan:

- Multiple active timetables on a Standard account — exceptional contractual activation, with tailored scoping and billing (the feature is included by default in Premium);
- Linked accounts — linked accounts and shared resource occupancy, enabled on request by Omniscol;
- Custom roles — custom roles with a permission matrix;
- Backup points — restorable account states;
- Collaboration — real-time collaborative editing;
- **Staffing** — supervision / monitoring module (also available as a standalone offer without the rest of Omniscol);
- Synchronization with external systems — synchronization with an external ERP / IT system (Aurion, Auriga, or a connector added on a project basis);
- Activity logs — access to the logs when the retention depth is configured on the account;
- Add-on modules — additional optional modules, depending on module and contract.

Is there a "Staffing only" offer?

Yes. For institutions that only need supervision / monitoring management, the **Staffing** module is also sold without the rest of Omniscol. The account then provides only the **Staffing** module and the minimal user screens.

Where can I find the prices?

Prices are not shown in this documentation: they depend on the contract with your institution (size, features, duration, support). Contact contact@omniscol.com for a quote.

How do I know my account's plan?

The active plan and subscribed options are not visible in the regular administration interface (the corresponding section of **Settings** is reserved for Omniscol). To find out an account's plan, contact the Omniscol team.

Can I switch plans or add options during the year?

Yes. Adding an option takes effect immediately, without service interruption. Moving from Standard to Premium follows the same principle. Deactivating an option requires that the corresponding features are no longer used on the account (for example no resource sharing between accounts if linked accounts are removed, or no more simultaneous publications if multiple active timetables are removed from a Standard account that has the option).

Is there a trial period?

Yes. An evaluation period is generally offered for new accounts, with support from the Omniscol team for the initial setup. Conditions on request.

How does billing work?

Billing is annual by default, with a commitment over the chosen duration. Specific conditions (billing frequency, payment methods) are defined in the contract.

What happens at the end of the contract?

Your data remains accessible during a contractually defined transition period, during which the full JSON export remains available. Reversibility is guaranteed by contract (GDPR).

See also

[Omniscol plans and options](#)

[What is Omniscol?](#)

[Security and hosting](#)

Glossary

Absence	External teacher / Adjunct instructor	School year
Alternate lessons / Weeks A-B	Free group	Sharing link
API token	Grade	Site
Associated lessons (alternating half-groups)	Group	Solver / Automatic generation algorithm
Calendar mode	Group alignment	Staffing
Campus	Group of groups	Student / Pupil / Learner
Class	Holidays	Subject
Class division	iCal (calendar export)	Substitution / Replacement
Classroom	Incompatibility (between subjects)	Teacher / Instructor
Classroom specialisation	Large room (several simultaneous lessons)	Teachers' availability (wishes)
Co-teaching	Lesson / Session	Timeline
Concatenated lessons	Lesson modality	Timetable
Conflict	Lesson status	Timetable display mode
Course	Multi-room (one lesson in several rooms)	Transverse course
Cyclic timetable	Publication / Activation of a timetable	Type of course
Dashboard	Resource	Weekly timetable
Date window		
Diagnostic		
Display panel		

Absence

Source: <help/en/glossary/absence.md> · id: `glossary.absence` · Updated: 2026-05-14

Also known as : *unavailability · leave · time off · one-off absence*

An **absence** in Omniscol is the declaration that an entity will not be present over a given period. The managed entities are teachers, classes, students and, if the Staffing module is active, staff members.

An accepted absence can change how lessons are displayed:

- **absent teacher without a substitute**: the lesson is marked as affected, with the teacher struck through;
- **absent teacher with a substitute**: the substitute is displayed next to the struck-through regular teacher;
- **absent class**: the class's lessons are removed from the display over the period;
- **absent student**: tracking attached to the student, without changing the lessons of their class.

Statuses of an absence

- **accepted**: the absence is validated and taken into account.
- **pending**: request declared by the user concerned, to be validated by an administrator.
- **rejected**: rejected request.
- **aborted**: request canceled by its declarant.

Only absences with the **accepted** status affect how timetables are displayed.

Fields of an absence

- **Entity**: teacher, class, student or staff if the Staffing module is active.
- **Date range**: start and end. A genuinely open-ended absence must be tracked explicitly; the end of the school year can serve as a technical bound if no end is provided.
- **Time range**: full day or specific time slots.
- **Filters**: classes, subjects or assignments depending on the entity type.
- **Reason**: chosen from the list provided; an administrator can enter a custom reason.

- **Comment:** short free text.
- **Status:** accepted, pending, rejected or aborted.

See also

[Substitution / substitute teacher](#)

[Absences module — overview](#)

[Declaring an absence](#)

Alternate lessons / Weeks A-B

Source: <help/en/glossary/alternate-lessons.md> · id: [glossary.alternate-lessons](#) · Updated: 2026-05-10

Also known as : *alternate weeks · alternating weeks · weeks A/B · odd and even weeks · weekly alternation*

Alternate lessons are lessons that do not recur every week, but alternate with other lessons on the same slot. The classic case: one lesson in week A, another in week B, on the same day at the same time.

Configuring the cycle

The alternation is configured in [Save](#). Three options:

- **Disabled** — no alternation.
- **Letters** — week A, B (and C, D... if you add more).
- **Numbers** — week 1, 2 (and 3, 4... if you extend).

Omniscol does not impose a 2-week cycle — you can alternate over 3, 4 or more weeks.

Creating an alternate lesson

When configuring the hours distribution:

1. Hover over the lesson to alternate. A **+** icon appears at the top right; click it.
2. A new free slot is added for the alternate week.
3. Create the lesson corresponding to this new slot.
4. Position it with its pin button **📌**, then click the desired day/time slot among the colored placeholders.

You can repeat this to add more alternate weeks to the same course.

Combining with other complexities

An alternate lesson can be:

- **simple** — pure alternation between two lessons,
- **concatenated** — a lesson that alternates and is also a double slot,
- **associated** — combined with associated lessons (group swap: A in biology then physics, B in physics then biology). The group swap is a complexity **distinct** from the A/B alternation, but it can be combined with it.

Offset caused by holidays

If you were in week A just before the holidays and want to resume in week B afterwards, create a **virtual offset** on the timeline of the year's weeks (see [school year](#)). Without an offset, the alternation simply resumes its normal cycle.

See also

[Lesson / Session](#)

[Complex lessons](#)

[School year and holidays](#)

API token

Source: <help/en/glossary/api-token.md> · id: [glossary.api-token](#) · Updated: 2026-05-15

Also known as : [api key](#) · [bearer token](#) · [authentication token](#)

An **API token** allows third-party software to call Omniscol's API endpoints on behalf of the school, for reading and, if write API endpoints are selected, for writing.

It is powerful — so handle it with care.

Generation

A token is generated in [Sharing](#) (Import/Export > Sharing). You provide:

- a key (generated in Omniscol, with a label and an optional expiration; its expiration date can be changed later),
- a scope: the list of API endpoints this token will be able to call (grant only what is strictly necessary),
- optionally a **token expiration**. This date is written into the generated JWT and cannot be changed afterwards.

Deleting the key, or reaching its expiration, immediately revokes the derived tokens. To change the expiration of an individual token, generate a new JWT.

Usage

The token is sent in the HTTP header `Authorization: Bearer <token>`, or (less recommended for production) in the URL via `?auth=<token>`.

Best practices

- **One key per integration** — makes independent revocation easier.
- **Minimal scope** — only check the APIs you need, not everything by default.
- **Expiration at the right level** — use the key expiration to manage a long-lived integration; use the token expiration for temporary, non-modifiable access.
- **Periodic rotation** — regenerate the key every 6-12 months.
- **Not in Git** — the token must not end up versioned in a public repository. Use your server's environment variables.

Difference from OIDC / SSO

To sign in real users with their institutional identity, use [OIDC / SSO](#). The API token is meant for server-to-server technical integrations, not for users' daily sign-in.

See also

[Sharing link](#)[Omniscol API](#)[MCP — connect an external AI agent](#)

Associated lessons (alternating half-groups)

Source: <help/en/glossary/associated-lessons.md> · id: [glossary.associated-lessons](#) · Updated: 2026-05-10

Also known as : [group a/b rotation](#) · [swap](#)

Associated lessons are two simultaneous half-group lessons that **alternate consecutively**. The typical case:

```
Slot 1: group A in biology, group B in physics
Slot 2: group A in physics, group B in biology
```

Both groups have covered both subjects after the two slots, but in a different order. This structure is very common in experimental sciences (biology labs + physics labs with two specialist teachers who "rotate"), in the arts (one

half-group in practice, the other in theory, then they swap), etc.

Creation

1. First create two **concatenated** lessons (one under the other).
2. Hover over the boundary between the two: an "association" button appears.
3. Click it and designate the **two groups** that must alternate — ideally declared as a **class division**.

Consistency with groups

The groups used in an association must be declared as a **class division** in the class — otherwise the solver raises a consistency warning (a group cannot be in two places at the same time).

See also

[Concatenated lessons](#)

[Class division](#)

[Complex lessons](#)

Calendar mode

Source: <help/en/glossary/calendar-mode.md> · id: [glossary.calendar-mode](#) · Plan: [premium](#) · Updated: 2026-06-13

PREMIUM

Also known as : *agenda mode* · *dated schedule* · *non-recurring*

PREMIUM

A timetable in **calendar mode** organises lessons **by date**, with no recurring typical week. Each lesson is positioned individually on a specific date, as in a diary.

It is the preferred mode for:

- **higher education** (engineering schools, business schools, universities),
- **continuing education** (one-off sessions, short programmes),
- **training centres** where lessons do not repeat every week.

This mode is available on **Premium** accounts.

Differences from weekly mode

Aspect	Weekly	Calendar
Recurrence	Typical week	Date by date
Auto generation	Yes	Yes (with a date window and day compacting)
Availability	Validated in advance	Consolidated in real time
Editing a lesson	Affects all active weeks	Affects one specific date
Groups of groups	Yes	Yes
Publication	Over week ranges	Binary (published or not)
Default display	Week view	Week or month view
Clearing on holidays	Automatic	None (lessons are already dated)

Automatic generation in calendar mode

The solver also works in calendar mode: it places unpositioned lessons on the available dates, respecting the same constraints (a teacher or a class cannot be in two places at once, compatible classrooms, blocking availability constraints). The generation screen also exposes calendar-specific settings: a **target date window** to limit the computation to a period, and **day compacting** (grouping at the start or end of the window, or letting the solver decide). Manual placement assistance remains available in parallel: conflicts detected in real time, filtered classroom suggestions, availability taken into account instantly.

Availability in calendar mode

On a calendar timetable, the availability of external instructors can change over time. Omniscol consolidates it in real time and raises conflict alerts as availability changes. See [Availability in calendar mode](#).

Combining weekly + calendar in the same school

On Premium, you can **publish simultaneously** a weekly timetable (the recurring morning lessons, an integrated preparatory cycle) and a calendar timetable (the one-off afternoon masterclasses, a graduate cycle) over the same weeks, thanks to [multiple active timetables in parallel](#). If the two timetables share entities (teachers, classrooms), Omniscol dynamically merges the views and detects their cross-conflicts.

See also

[Timetable](#)[Choosing the right timetable type](#)[Calendar mode](#)[Groups of groups](#)

Campus

Source: [help/en/glossary/campus.md](#) · id: [glossary.campus](#) · Plan: [premium](#) · Updated: 2026-06-13

PREMIUM

Also known as : [branch](#) · [faculty](#) · [hub](#)

PREMIUM

A **campus** lets you distinguish several branches, faculties or hubs within a single Omniscol account.

The campus concept is **reserved for Premium accounts** and remains optional: only create campuses if they genuinely help you reflect your organisation.

The concept is **organisational**. It does not replace the [site](#), which describes a geographical or physical reality: location, time grid, classrooms, travel time.

A campus becomes particularly useful when the site concept does not match your logical organisation. Examples:

- several faculties share the same buildings;
- several schools of the same group share several sites, in a criss-cross fashion;
- a single physical site hosts several hubs or programmes.

Even when campus and site largely overlap, the concept can remain useful to make filters and groupings easier.

What a campus is for

A campus is mainly used to:

- group classes according to a shared organisation;
- make some filters and groupings easier, especially in the conflict diagnostics.

A campus on its own does not define a time grid, a travel time or a strict placement constraint. These behaviours remain carried by the sites, classrooms, resources and constraints of the timetable.

Where to configure it

Campuses are created in the [general settings](#), below the class levels. Each class can then be attached to a campus, which feeds the corresponding filters and groupings.

See also

[Site](#)[Classroom](#)[Class](#)[Grade](#)[Detect and resolve conflicts](#)

Class

Source: [help/en/glossary/class.md](#) · id: [glossary.class](#) · Updated: 2026-07-05

Also known as : *cohort · intake · session · year group*

In Omniscol, a **class** is the main entity that brings together students following the same curriculum in the same school, over a given period. It is what courses are addressed to: a course targets a class (in full, or via a [group](#) that subdivides it), not the other way round.

A class carries:

- a **name** (for example "Year 7A", "Grade 10 Homeroom", "Cohort 2028"),
- a **level** (levels are defined in [+ Create](#) — for example Grade 6/Grade 7/Grade 8 in school, Year 1/Year 2/Year 3 at bachelor level),
- optionally a **campus** (Premium option; branch, faculty, hub or other organisational grouping),
- a default **site** (physical location and classrooms used by default),
- optionally a **dedicated classroom** in which its lessons take place by default,
- a **theoretical headcount** (expected number of students; optional but useful for sizing classrooms).

Vocabulary depending on the context

The word "class" takes on very different nuances depending on the type of institution. All of them map to the same Omniscol entity:

- **Primary / secondary** — "class" in the ordinary school sense: one stable pupil group attached to a year, section or homeroom (for example "Year 7A", "Grade 10 Homeroom", "5th Grade Blue").
- **Higher education** — often "cohort", "intake", "session", "year group", sometimes "track" or "specialisation". An intake of 200 students is a class; tutorial and practical subgroups are [groups](#), not classes.
- **Continuing education** — "session", "cohort", "learner group".

COUNTRY: US

In a US context, "class" here refers to a fixed group of students who share the same schedule, close to a homeroom or cohort. It does not mean a subject or a course in the sense of "I'm taking a math class". Typical examples: "5th Grade Blue", "Grade 10 Homeroom", or "Cohort 2028".

COUNTRY: GB

In a UK context, "class" in this app refers to a stable group of students with a shared timetable, close to a form group, set or band. It does not mean an individual subject or lesson. Typical examples: "Year 8 Form A", "Year 10 Set 3", or "Band B".

COUNTRY: IE/AU/NZ/ZA

In Irish, Australian, New Zealand and South African school contexts, "class" here refers to a stable group of students with a shared timetable, close to a year group, form group or home group. It does not mean an individual subject or lesson. Typical examples: "Year 8 Form Class", "Year 10 Home Group", or "Year 9 Cohort".

PREMIUM

In higher education, "class" here means a stable group of students following the same overall program or timetable. It often corresponds to a cohort, a training group or sometimes a track. It is distinct from a course, a unit or a module.

When you read "class" in Omniscol, read "the cohort following this curriculum over this period", not "the physical room" (which is a [classroom](#)).

Class vs group

A classic pitfall: confusing a class and a group. Groups are **always** subdivisions of a class. If you need to mix students coming from several classes (for example the Latin students of Year 8A, Year 8B and Year 8C attending the same Latin course), you do not create a new class: you use a [group alignment](#) or a [group of groups](#).

See also

[Group](#)[Grade](#)[Class division](#)[Group alignment](#)[Campus](#)[Site](#)[Class, group, subgroup](#)

Class division

Source: [help/en/glossary/class-division.md](#) · id: [glossary.class-division](#) · Updated: 2026-05-10

Also known as : *group partition*

A **class division** brings together several [groups](#) of **one and the same class** that must **share the same time slot**.

In practice: since students are not duplicated (the same student cannot be in two places at the same time), if your half-groups Lab-A and Lab-B must do two different practicals simultaneously, the solver needs to know that they form a division. The engine then knows how to:

- place both practicals in the same time slot,
- with two different classrooms,
- with two different teachers,
- without raising a conflict alert.

When to create a class division

Whenever the same time slot must host different lessons for disjoint subsets of a class:

- practical half-groups (Lab-A vs Lab-B, Mathematics vs English half-classes, etc.),
- language groups (Spanish vs German),
- exclusive electives (Latin vs Classical Greek),
- computing half-groups (because the computer room only has 15 workstations).

The validity criterion: **no student may belong to two groups of the same division**. Otherwise, you are dealing with an alignment rather than a division (which is in any case not possible if the groups belong to the same class).

How to create a class division

On a class's [groups page](#), select the groups concerned and click [+ Add class division](#). Several divisions can coexist within the same class (one for science practicals, one for languages, one for philosophy electives).

Difference from an alignment

A **class division** = groups of a **single class** that must be in the same time slot. An **alignment** = groups from **different classes** that must be in the same time slot, **with the same teacher and the same classroom**.

See also

[Group](#)[Group alignment](#)[Full page on class divisions](#)

Classroom

Source: <help/en/glossary/classroom.md> · id: [glossary.classroom](#) · Updated: 2026-06-13

Also known as : *lecture room · teaching room · dedicated room · premises*

A **classroom** in Omniscol is a physical room in which a lesson can take place. Each classroom belongs to a [site](#). A classroom carries:

- a **name** ("A102", "Amphi Newton", "Lab. Chimie 3"),
- a **capacity** (number of students the room can hold; a critical field for the solver),
- optionally a **Maximum number of classes** setting when it is a **large room** able to host several lessons at the same time (exam room, theater, gym, swimming pool, outdoor area),
- optionally a **specialisation** (chemistry, computing, sport, multimedia, etc. — a single label per room),
- optionally free-form **tags** or comments (flip chart, fixed projector, power outlets for students, etc.),
- optionally a **building**,
- **opening hours** (useful for rooms shared with another school, or for rooms open only on certain days).

Assignment to lessons

A lesson can have one or [several rooms](#). When you assign a room manually, Omniscol:

- **pre-filters** unsuitable rooms (wrong specialisation, insufficient capacity, unavailability, wrong site) by displaying them semi-transparent,
- **sorts** compatible rooms by suitability (stars: optimal fill, etc.),
- lets you force an inconsistent assignment, but raises an alert.

Capacity and sizing

Capacity is a critical field. When in doubt, and if you are certain the room will be large enough, enter a high number. If you undersize it, the solver will reject valid configurations or raise alerts.

Large room: several simultaneous lessons

The **Maximum number of classes** field turns a room into a **large room**: it can then host several different lessons at the same moment (distinct teachers and groups), within the limit of that number and of its total capacity — typically an exam room, a theater, a gym, a swimming pool or an outdoor area. This setting only appears on a room that has a **specialisation**. Left empty, the room hosts only one lesson at a time.

Specialisation

The **specialisation** (chemistry, computing, sport, multimedia, supervised test room...) is a free label, unique per room. It is then used to state, when assigning a subject to a class, that a room carrying exactly this label is required. The engine strictly enforces this constraint.

Room dedicated to a class

Many primary/secondary setups associate a room with each class (lessons take place there by default, and only the teachers move around). You can configure this on the class (see [creating classes](#)).

Rooms in two virtual sites

If you have created two virtual [sites](#) for a single physical location (typically a middle school and a high school sharing the premises), a room can belong to only one of the two sites at a time. To make it usable on the middle school side **and** on the high school side, duplicate it in both sites and **enter mutually exclusive opening hours**.

Omniscol does not distinguish a single room shared between two sites from two rooms with the same name: it does not cross-check the two entries automatically, and it is up to you to avoid double-booking the real physical room.

See also

[Site](#)[Campus](#)[Classroom specialisation](#)[Multi-room](#)[Large room](#)[Resource](#)

Classroom specialisation

Source: <help/en/glossary/classroom-specialization.md> · id: [glossary.classroom-specialization](glossary/classroom-specialization) · Updated: 2026-05-10

Also known as : *room type · specialised room · subject-dedicated room*

A **classroom specialisation** indicates that a classroom is dedicated to a certain type of activity: chemistry laboratory, computer room, gym, multimedia room, supervised test room, art workshop, music studio, etc.

It is a **free-form** field — you create the specialisations you need, with your own nomenclature. No closed list imposed by Omniscol.

Solver behaviour

When a subject requires a specialisation, the solver assigns **only** the classrooms that carry that specialisation. A chemistry lab room will never be used for a literature lesson if you required "chemistry" on the practical subject.

Forcing an inconsistent assignment (manually giving a chemistry lesson a classroom that does not have the chemistry specialisation) **does not block** but displays an alert.

Tidying up or deleting a specialisation

A specialisation exists as long as it is used by at least one classroom. To delete it, remove it from all the classrooms that carry it — on the next save, it disappears from the reference list.

To **rename** one, create the new specialisation with the right name, replace it on the classrooms, and the old one will disappear automatically (provided it is no longer referenced anywhere, including on the subjects side).

See also

[Classroom](#)[Subject](#)[Classroom specialisations](#)

Co-teaching

Source: <help/en/glossary/co-teaching.md> · id: [glossary.co-teaching](glossary/co-teaching) · Updated: 2026-05-10

Also known as : *team teaching · dual instructor · main and secondary instructor*

Co-teaching refers to a lesson delivered simultaneously by **two or more teachers**, in the same room, with the same student audience.

Omniscol models this case simply: a course can have **several instructors**. Each of them is credited with the lesson in their statistics and their service hours.

Typical cases

- **Reinforced supervision** — a permanent teacher + a teaching assistant, or two teachers for a large group.
- **One-off visiting professor** — a course whose session of the day is co-taught by an external instructor (see [adjunct / external teacher](#)).
- **Co-leads** — a module taught by a pair of teachers throughout the semester.

- **Dual expertise** — a multidisciplinary module (for example "IT for healthcare" with a physician and a computer scientist).

Difference from alternation

Co-teaching means **two teachers at the same time on the same lesson**. **Alternation** ([alternate lessons](#)) means **two teachers on the same time slot, but on different weeks**.

If your two teachers alternate every week but are never together, those are alternate lessons, not co-teaching.

If one of them only takes part in a few one-off sessions, the simplest approach is to create a separate course for those specific dates with the visiting professor, in addition to the main course — especially in [calendar mode](#).

See also

[Lesson / Session](#)[External teacher](#)[Co-teaching in higher education](#)

Concatenated lessons

Source: [help/en/glossary/concatenated-lessons.md](#) · [id: glossary.concatenated-lessons](#) · Updated: 2026-05-10

Also known as : [back-to-back lessons](#) · [consecutive lessons](#) · [double lessons](#) · [chained lessons](#)

Concatenated lessons are two lessons that must be placed **consecutively** within the day. The solver guarantees that they follow each other with nothing in between.

Typical cases:

- a **2-hour practical** built from two 1-hour blocks,
- a **lecture followed by a tutorial** (an introductory lecture, then an application tutorial),
- a **2-hour double exam session** on a 1 h × 2 slot.

Creation

Drag and drop one lesson under another in the hours distribution view. The two lessons then appear **stuck together**, one below the other, forming a single block.

Detaching

The "scissors" button that appears on hover between two concatenated lessons **splits** the concatenation. The two lessons then become independent again.

Why not create a single, longer lesson?

Good question. Two main reasons to concatenate rather than lengthen:

1. The lessons can have **different types** (a lecture concatenated with a tutorial), or **different teachers**, or **different rooms**.
2. You want the statistics to count two separate lessons (two lesson-log entries) rather than one.

If all the attributes are identical, lengthening a single lesson is simpler. Concatenation brings flexibility where it is needed.

See also

[Associated lessons](#)[Complex lessons](#)

Conflict

Source: <help/en/glossary/conflict.md> · id: [glossary.conflict](#) · Updated: 2026-06-25

Also known as : *alert · detected incompatibility · overlap · collision*

A **conflict** in Omniscol is a situation where two constraints cannot be satisfied at the same time. Examples:

- a teacher assigned to two lessons on the same time slot,
- a room booked by two lessons at the same time,
- a group of students expected in two places at once,
- a lesson that does not respect the specialisation required for its subject,
- a lesson whose total capacity (sum of the assigned rooms) is lower than the group's headcount,
- an alignment whose aligned groups have different numbers of hours.

Real-time detection

Omniscol detects conflicts **as you enter data**, not only at generation time:

- a red banner at the top of the hours distribution screen,
- an **▲** icon on the affected lessons and classes,
- a clickable list of conflicts (the "magnifier" leads to the details).

Intentional conflicts

Not every conflict has to be resolved. Some are intentional:

- the same group on an exam + an extra-time room (the same cohort of students is expected in two rooms depending on conditions),
- headcount > room capacity when you know that not all enrolled students will attend.

You can **leave the conflict**; the alert stays displayed but does not block operation.

Blocking conflicts

Some inconsistencies prevent the **automatic generation** from succeeding:

- no room compatible with a subject + a number of hours,
- a teacher whose availability is entirely incompatible with their courses,
- a structurally impossible alignment.

These conflicts are flagged as **critical**; as long as they exist, the  **Generate timetable** button remains unavailable.

See also

[Diagnostic](#)

[Solver](#)

[Detecting and resolving conflicts](#)

Course

Source: <help/en/glossary/course.md> · id: [glossary.course](#) · Updated: 2026-06-26

Also known as : *class subject · subject of a class · course line*

A **course** is the teaching of a **subject** to a **class** (or to one of its **groups**), over a given period. It is the basic unit of your **course offering**: what must be taught, to whom and in what volume — independently of when it will land in the timetable.

A course carries:

- a **subject** (the discipline being taught),

- optionally a **course type** (lecture, tutorial, lab, exam...) — the (subject, type) pair forms a distinct course,
- a target **number of hours** (hours per week, or a number of lessons),
- one or more assigned **teachers**,
- optionally **constraints**: pedagogical weight, required specialised room, incompatibilities, placement preferences.

COUNTRY: US

In US school usage, this is often thought of as **periods per cycle** or **contact time**. Related but not identical: **Carnegie units** and **student hours** are time-based measures used for credits or workload. In Omniscol, the field here stores the teaching volume itself, not the credit value.

COUNTRY: GB

In UK school usage, this is often expressed as **contact hours**.

Course, subject, lesson: do not confuse them

- The **subject** is the pure discipline (mathematics), in the school's catalog.
- The **course** is that subject *applied to a class*, with a volume and attributes: "mathematics in Grade 9 A, 4 h per week, as a tutorial".
- The **lessons** are the concrete **occurrences** that implement the course in the grid (the four weekly slots actually placed).

A course therefore gives rise to one or more lessons; deleting a lesson does not delete the course.

A concept the interface keeps implicit

The interface does not say "course" at this point: you build one by **assigning a subject to a class** (the **Courses** tab of a class), then filling in the volume, the type and the teachers. The notion remains useful for reasoning — particularly in higher education, where the course (the definition) is clearly distinguished from its lessons (the occurrences).

All the courses together: the program

The set of courses of a class — or of a program of study — forms its **program**. Depending on the level and the school's vocabulary, it is also called a **curriculum**, a **course catalog**, a **degree plan** or a **syllabus**. Omniscol does not impose this vocabulary: it handles courses (class subjects, possibly typed); their sum *is* the curriculum.

See also

Subject

Lesson / Session

Type of course

Courses, lessons, course types

Data organization: subjects, classes, timetables

Configuring subjects per class

Cyclic timetable

Source: <help/en/glossary/cyclic-timetable.md> · id: [glossary.cyclic-timetable](#) · Updated: 2026-05-10

Also known as : *cycle* · *numbered days*

A **cyclic timetable** defines recurring lessons over an arbitrary number of days (Day 1, Day 2, ... Day 6, etc.), not necessarily aligned with the 5- or 7-day week.

It is typically used in **North American systems** (6- or 8-day cycles), in some international schools, and more rarely in Europe.

Difference from the weekly mode

Aspect	Weekly	Cyclic
Repetition unit	Week (5-7 days)	Cycle of N days
Numbering	Monday, Tuesday...	Day 1, Day 2...
At publication	You choose the active weeks	You choose the weekdays the cycle applies to
Auto generation	Yes	Yes

When to use it

If your school runs on a 6-day cycle (for example Day 1 on Monday, Day 2 on Tuesday, Day 3 on Wednesday, Day 4 on Thursday, Day 5 on Friday, Day 6 the following Monday, Day 1 the following Tuesday...), the weekly mode is not suitable — the cyclic mode is designed for this case.

If you are in Europe with a 5-day cycle strictly aligned with the week, stay in weekly mode — it is simpler to configure and publish.

See also

[Timetable](#)[Choosing the right timetable type](#)

Dashboard

Source: <help/en/glossary/dashboard.md> · id: [glossary.dashboard](#) · Updated: 2026-05-18

Also known as : [statistics](#) · [indicators](#) · [reporting](#)

The **Dashboard** module provides statistics calculated on the actual timetable (lessons placed, absences taken into account, substitutions applied) over a chosen period: week, month, school year, or custom date range.


Available indicators

Statistics are organized by analysis dimension: teachers, classrooms, resources, subjects, classes and students. The module's full page details the indicators available for each dimension.

Use cases

- **HR reporting** — record of hours taught for payroll, certifications, accreditations.
- **Classroom optimization** — detect underused or overloaded classrooms.
- **Resource control** — check the actual use of movable equipment.
- **Educational audit** — check that all the planned hour volumes were actually delivered.

Export

The  **Print** button opens a table that can be copy-pasted, printed, or exported as CSV for external processing (Excel, Power BI, etc.).

Calculation

Important: the Dashboard works on the **operational timetable** (the lessons actually placed and published, merging [multiple active timetables in parallel](#) where applicable) and **takes validated absences and substitutions into account**. The calculation reflects reality, not the planned configuration.

See also

[Overview of the Dashboard module](#)[Using tables and charts](#)[Teacher statistics](#)[Classroom statistics](#)[Subject statistics](#)[Class statistics](#)

Date window

Source: [help/en/glossary/date-windows.md](#) · id: [glossary.date-windows](#) · Plan: [premium](#) · Updated: 2026-06-20

PREMIUM

Also known as : [date windows](#) · [inclusion window](#) · [exclusion window](#) · [date overlay](#)

PREMIUM

A **date window** is a **reusable** date **inclusion** or **exclusion** period that restricts when certain lessons can be scheduled. Defined once on the edit screen of a calendar timetable, it then applies as an **overlay** on the time constraints of several entities — classes, groups, subjects or classrooms.

Reserved for **calendar-type timetables**, on **Premium** accounts.

Two types

- **Allowed lessons** — lessons can only be scheduled during these dates.
- **Excluded lessons** — lessons cannot be scheduled during them.

Multiple windows stack; where they overlap, an exclusion takes precedence over an allowed period.

See also

[Date windows](#)[Calendar mode](#)

Diagnostic

Source: [help/en/glossary/diagnostic.md](#) · id: [glossary.diagnostic](#) · Updated: 2026-06-26

Also known as : [check](#) · [verification](#) · [configuration alert](#) · [consistency alert](#)

A **diagnostic** in Omniscol is a consistency check run continuously by the system, which points out timetable configuration problems before they make the generation fail or produce an incorrect result.

Severity levels

- **Critical** (red, blocking) — prevents the generation from succeeding: structurally impossible alignment, insufficient total capacity, teacher without any free time slot...
- **Warning** (orange) — can be left as is but signals a risk or an inconsistency to review: imbalance in the aligned hour volumes, slightly undersized classroom, very restrictive availability.
- **Information** (blue) — non-blocking note: "some classrooms are never used", "a teacher has service hours well below those of their peers".

Where to see diagnostics

- In the **top banner** on the hours distribution screen and the timetable view.
- Next to the **name of the class** or teacher concerned in the lists.
- Detailed in the [Generation](#) tab (or checking depending on the timetable mode) with a contextual explanation for each diagnostic.

Frequent diagnostics

Diagnostic	Typical cause	How to resolve
No compatible classroom	Required specialisation with no matching classroom	Create the classroom or remove the specialisation
Inconsistent hour volume in an alignment	One class has 2h, the other 3h on the same aligned group	Align the volumes, or unalign
Insufficient capacity	Headcount > classroom capacity	Assign a larger classroom, or several classrooms (multi-room)
Availability with no possible time slot	All time slots are marked impossible for this teacher	Relax the availability or redistribute the lessons
Lesson during an absence period	Lesson placed in a range where the teacher is absent	Move the lesson or remove the absence

See also

[Conflict](#)[Solver](#)[Diagnosing a failed generation](#)

Display panel

Source: <help/en/glossary/panel.md> · id: [glossary.panel](#) · Updated: 2026-06-13

Also known as : [lobby screen](#) · [hall screen](#) · [entrance screen](#)

A **panel** in Omniscol is a public display screen, configured to continuously show a filtered timetable. Typical cases:

- **lobby panel** — a screen at the school entrance, cycling through the day's lessons for all classes,
- **panel outside a room** — a screen mounted next to a door, showing that room's occupancy over the day,
- **lecture hall / laboratory panel** — a display of the upcoming lessons in a specialized space.

A unique URL for each panel

Each panel you create has a **unique URL** to open in full screen in a browser. Scrolling and refreshing happen automatically.

Customization

When defining a panel, you configure:

- **filter of the displayed lessons**: which grade(s), class(es), site(s), room(s),
- **time range** displayed,
- **font size**,
- **information banner** (scrolling text, logo),
- **colors and theme**.

For deployments with a strong visual identity, it is also possible to entirely replace the web page that renders the panel, to integrate it into an existing digital-signage setup. See [Visual customization](#).

See also

[Lobby panel](#)[Panel outside a classroom](#)[Visual customization](#)

External teacher / Adjunct instructor

Source: <help/en/glossary/external-teacher.md> · id: [glossary.external-teacher](#) · Updated: 2026-06-13

Also known as : *visiting professor* · *guest instructor* · *sessional teacher*

An **external teacher** (or *adjunct*, *visiting professor*, *expert* depending on the context) is a teacher whose contractual relationship with the school is temporary or occasional. From the solver's point of view, an external teacher is a [teacher](#) like any other — same availability, same assignments.


The distinction is mainly **administrative**:

- a short contract (by the hour, by the lesson, by the assignment),
- a smaller volume of hours than permanent teachers,
- a user often not hosted on the same systems as permanent teachers (different e-mail, no access to the same internal tools).

PREMIUM

The dedicated External teacher marker, ticked on the user's profile, is reserved for Premium accounts. It explicitly identifies external teachers and drives the dedicated display described below.

Dedicated display

A  icon distinguishes external teachers from permanent ones on some screens (notably the lesson tooltip and the list view of timetables). On the main time grid, the default display makes no distinction for space reasons, but the information remains visible in the tooltip on hover.

Special case: a visiting professor for a few lessons

For a one-off lesson where a visiting professor co-teaches with the main teacher without alternating, see [co-teaching](#). For a weekly alternation of two teachers on the same time slot, see [alternate lessons](#).

See also

[Teacher](#)

[Co-teaching](#)

[External teachers in higher education](#)

Free group

Source: <help/en/glossary/free-group.md> · id: [glossary.free-group](#) · Updated: 2026-06-13

Also known as : *satellite group* · *semi-autonomous group* · *open group* · *open-enrollment group*

A **free group** is a **semi-autonomous working group**: part of the class works independently, usually **in the same room and with the same teacher** as the main lesson, in parallel with it — a kind of **satellite group** of the course. The planner wants to **show it on the timetable** without setting up a formal group division opposite it.

That is where the word **free** comes from: since this subgroup shares the room and the teacher of the course it comes from, a regular [group](#) (even in a division) would trigger a **room and teacher conflict**. The free group is the **wild card** that **disables all conflicts** — whether with the rest of the class, with the teacher or with the room. It therefore coexists with its originating lesson.

This need arises in particular in **art schools** and **social work programs**, where part of the time is spent in **supervised independent work**: a subgroup moves forward on its project in a corner of the room while the rest of the class follows the lesson. More broadly, the free group models any group whose membership is **not fixed**, without generating conflicts.

Difference from a regular group

A regular group is taken into account by the solver: it checks that its students are not double-booked (two lessons at the same time) and that both its **room** and its **teacher** are free. That is exactly what would block the satellite group: the room and the teacher of the main course are already taken.

A free group **lifts these checks**. Its lesson can therefore share the room and the teacher of the course it comes from; and a student can belong to several overlapping free groups without any alert being raised.

In practice: manual placement

Automatic generation **does support** free groups, but it is recommended **not** to use it in this case: **place and lock by hand** the main lesson and its satellite lesson on the desired time slot. The solver does not know which one is the main lesson and which one is the satellite — left to generation, they could land anywhere. Given its very occasional and specific nature, the free group calls for **manual use**.

Creation

When creating the group, check the "free group" option on [+ Add group](#). Visually, these groups are marked in the interface so they are not confused with constrained groups.

PREMIUM

Free groups are a capability of Premium accounts.

See also

[Group](#)[Free groups](#)

Grade

Source: <help/en/glossary/level.md> · id: [glossary.level](#) · Updated: 2026-06-21

Also known as : *class level* · *year group*

A **grade** orders [classes](#) by **educational progression**: Grade 6, Grade 7, Grade 8, Grade 9 in school; L1, L2, L3 in a bachelor's degree; Bachelor, Master, year 1, year 2... depending on the institution.

It is an **organizational label**, not a placement constraint: the grade imposes nothing on the solver. It is used to **sort and find** classes — in the class screens, in several filters and in some diagnostic views.

Where to configure it

Grades are part of the account's **general** settings. From the [general settings](#), under **Grades**, you can create them via [+ Create](#), delete them, and **drag and drop them to reorder** — the order reflects the progression.

Each class is then assigned a grade when it is created (see [Creating the classes](#)).

Grade, campus and site

Three neighboring notions not to be confused:

- the **grade** places the class in a **progression** (how far along the curriculum the class is);
- the **campus** attaches it to an **organizational entity** (faculty, division, location);
- the **site** gives its **physical location** and the rooms used by default.

The same class generally carries all three: a grade, possibly a campus, and a default site.

See also

[Class](#)[Campus](#)[Site](#)[Creating the classes and their groups](#)[General settings](#)

Group

Source: <help/en/glossary/group.md> · id: [glossary.group](#) · Updated: 2026-07-05

Also known as : [subgroup](#) · [half-group](#) · [elective group](#) · [lab-a](#) · [lab-b](#)

In Omniscol, a **group** is a **subdivision of a class**. All the students in a group are also students of the parent class.

You create a group whenever a subset of a class's students must attend a course different from the rest of the class — even temporarily, for a single lesson. Examples:

- lab half-groups ("Lab-A", "Lab-B"),
- elective groups ("Latinists", "Hellenists", "Spanish 2nd language", "German 2nd language"),
- level-based groups ("Advanced English", "Standard English").

COUNTRY: US

In US school usage, the closest equivalents are often **section**, **track** or simply **group**, depending on the purpose. Typical labels: "Honors", "ESL", "Section B".

COUNTRY: GB

In UK school usage, the closest equivalents are often **set** or **band** when students are split by level or option. In Omniscol, **group** remains the broader term: any subgroup of a class.

Strong recommendation: one group per course

Even if the Latinists and the Hellenists of a class are in practice the same students, **create two separate groups**. Otherwise the [alignments](#) with other classes become ambiguous and conflicts are hard to diagnose.

Conversely: no need to create a "Whole class" group. When a course is meant for the whole class, no group is attached; the course is simply assigned to the class.

Expected headcount

Enter the number of students expected in the group. This field is optional but recommended: the [solver](#) uses it to pick [classrooms](#) with adequate capacity.

Mixing students from several classes

The group concept stops at the boundary of one class. To mix the Latinists of 8A, 8B and 8C who have lessons together:

- create a "Latinists" group in each of the three classes,
- then create an [alignment](#) that ties the three together.

You can also use a [group of groups](#), which is easier to edit when the grouping needs to evolve.

Free groups

For open enrollment (evening workshops, electives not constrained by class), see the notion of a [free group](#) — membership is not fixed in advance, students can join or leave without generating conflicts.

See also

[Class](#)[Class division](#)[Group alignment](#)[Groups of groups](#)[Free group](#)

Group alignment

Source: <help/en/glossary/alignment.md> · id: [glossary.alignment](#) · Updated: 2026-06-26

Also known as : *cross-class grouping*

An **alignment** brings together several **groups** from **different classes** that must share **the same time slot, the same teacher and the same room**.

The classic example: the latinists of 8A, 8B and 8C attend a single Latin lesson together. Instead of creating three parallel lessons (which would compete for the teacher and the room), you create a **"Latinists" group** in each of the three classes, then **align** them. The solver then places a single lesson, linked to the three groups simultaneously.

Typical cases:

- cross-class language groups (English groups gathered from several classes),
- cross-class electives (Latin, Greek, political science),
- specialization subjects in French high schools (since the reform),
- co-modules in higher education (the same core-curriculum course shared by several programs).

Power and restrictions

Alignment is extremely powerful — it avoids duplication, guarantees the consistency of changes (a moved lesson is moved everywhere), and simplifies consultation. **But it is also very restrictive:**

- Once aligned, a group can no longer have an independent lesson on that specific slot: if one of the three classes has an exception lesson, you must **duplicate** its group and unalign the clone.
- All aligned groups must have, in their respective classes, exactly the same hourly volume and the same configuration. An imbalance produces a consistency warning (see the [Generation](#) tab).
- Changes (adding a lesson, changing the room) must be made once, on the alignment — Omniscop propagates them. But editing the lesson through a single one of the classes can produce side effects.

Important differences

Concept	Scope	Constraint
Class division	Groups of one class	Same slot, different teacher/room
Alignment	Groups from several classes	Same slot, same teacher, same room
Group of groups	Same as alignment	Same slot, single teacher/room, flexible editing afterwards

In calendar mode

In a timetable in **calendar** mode, the **group of groups** is preferred over the alignment: easier to edit, you can add or remove a group as you go without having to recreate the structure.

See also

[Group](#)[Class division](#)[Groups of groups](#)[Transverse course](#)[Group alignments](#)

Group of groups

Source: <help/en/glossary/groups-of-groups.md> · id: [glossary.groups-of-groups](#) · Updated: 2026-05-10

Also known as : *super-group · grouping of groups*

A **group of groups** is a grouping — more flexible than an **alignment** — of several **groups** coming from the same class or from different classes.

Available as a basic grouping mechanism in Omniscol.

Key difference from an alignment

An alignment is **frozen at creation**: the list of aligned groups is fixed, and taking a group out of it requires cloning the group.

A group of groups, by contrast, **can be modified after creation**. You can add or remove groups without having to recreate the structure. This is what makes it valuable in higher education, where the composition of groupings often evolves over the year (a new track joining a shared course, a subgroup splitting off for a project).

When to use it

- Courses shared by several tracks (engineering or business school),
- Cross-cohort lectures, transverse seminars,
- Core-curriculum courses in continuing education,
- Open educational events (a visit, an outside speaker).

A course assigned to a group of groups is visible (and editable) from each of the parent classes of the member groups.

Distinctive point

The group of groups is most useful when the composition of the audience evolves over the year or when you want to avoid the symmetry constraints of an alignment.

See also

[Group alignment](#)

[Transverse course](#)

[Calendar mode](#)

[Groups of groups](#)

Holidays

Source: [help/en/glossary/holidays.md](#) · id: [glossary.holidays](#) · Updated: 2026-06-26

Also known as : [breaks](#) · [public holidays](#) · [school holidays](#)

Holidays are school closure periods attached to a [school year](#): school breaks, bridge days and public holidays. Each has a name, a start date and an end date.

When a school year is created, Omniscol can offer to import the common holidays of the configured country, when data exists for the chosen range. The list can then be completed and adjusted manually.

Effect on the timetable

The effect of a holiday period depends on the type of timetable:

- **Recurring timetable (weekly or cyclic)**: at publication, the holidays cancel the lessons of the days concerned — recurring lessons do not take place on those days and the statistics take this into account.
- **Calendar timetable**: a lesson can be positioned on a holiday day, but this creates a conflict; automatic generation then reserves the right to reposition that lesson if it is not locked.

Holidays describe a **general** closure of the school. To neutralize a period on a narrower scope:

- a [class absence](#) removes one class over a given period;
- in a calendar timetable, [date windows](#) restrict the activity period of one specific course.

Where to configure them

Holidays are managed per school year, in the [school years](#), one by one or in a table.

See also

[School year](#)[Date window](#)[School years](#)

iCal (calendar export)

Source: [help/en/glossary/ical.md](#) · id: [glossary.ical](#) · Updated: 2026-05-15

Also known as : [ics](#) · [calendar subscription](#) · [ical link](#) · [ical export](#)

iCal (`.ics` extension) is the standard calendar exchange format, read by every calendar application on the market: Apple Calendar, Google Calendar, Outlook, Thunderbird, Fastmail, etc.

Omniscol offers two types of iCal export:

1. Static download

The [Download](#) button on a selected timetable. You get an `.ics` file to import into your calendar application. A **snapshot** at download time: it is not updated if the timetable changes afterwards.

Useful for sharing a one-off schedule by email.

2. Dynamic subscription link

The [Download](#) button on a timetable (or a teacher screen, etc.) generates an iCal subscription URL. Once copied into your calendar application as a **subscription**, the calendar updates automatically as the timetable evolves in Omniscol.

This is how each teacher and each student syncs their Omniscol timetable with their personal calendar, without having to re-download a file after every change.

Omniscol iCal links are signed, can expire, and are tied to the account that generated them. The expiration is embedded in the URL's token; to change it, generate a new link. A password change, a deactivation or a deletion of the holding account invalidates the associated links.

Synchronization and updates

The subscription iCal is **refreshed by the calendar application**, at a pace that varies by client:

- Apple Calendar: usually 5-15 min.
- Google Calendar: up to 24h in the worst case (but often faster).
- Outlook: varies.

This latency is inherent to the iCal protocol and does not depend on Omniscol — a change in the Omniscol timetable appears in the subscribed calendar as soon as it pulls the feed again.

See also

[Sharing link](#)[iCal — subscription and dynamic link](#)

Incompatibility (between subjects)

Source: [help/en/glossary/incompatibility.md](#) · id: [glossary.incompatibility](#) · Updated: 2026-06-13

Also known as : [exclusion](#) · [anti-chaining](#) · [sequence ban](#)

An **incompatibility** is a pedagogical constraint that forbids a subject from **following** another. It is a **directed** rule: you declare that "subject A must not be followed by subject B" — the direction matters, A-then-B and B-then-A are two distinct rules. The typical case: "no maths lesson right after sports".

Configured at class level in the Incompatibilities tab of the current timetable.

Scope of the ban

You choose the window in which the ban applies when you create the rule:

- **Consecutive** — not in the lesson that immediately follows, on the same day.
- **Half-day** — not later in the same half-day.
- **Day** — not later in the same day.
- **Week** — not later in the same week.
- **Always** (*cyclic or calendar timetable*) — never afterwards, over the whole period. This is the **sequencing** tool: finish a module before moving on to the one that depends on it.

Separately: a subject's **self-incompatibility** option (a global setting) prevents the same subject from coming back twice in the day for a student — often simpler than a multitude of pairwise rules.

Difference from subject time constraints

Not to be confused with a subject's **time constraints** (🕒), which fix the **absolute** placement of a single subject in the grid. An incompatibility, by contrast, bears on the **relative** order of two subjects.

Constraint	Bears on	Example
Incompatibility	the order of two subjects	"no maths after sports"
Subject time constraint	the absolute placement of a subject	"no maths in the first period"
Specialisation	the classroom required by a subject	"chemistry requires a laboratory"

An optimization constraint, not a blocker

An incompatibility is not a blocking constraint: the solver treats it as a penalty it tries to eliminate, just like undesirable availabilities. If the other constraints leave no alternative, an incompatibility can therefore remain in the generated timetable; the **diagnostic** then reports it so that you can arbitrate. Avoid an avalanche of incompatibilities: each one reduces the solver's freedom and lengthens the computation. To balance a subject across the week, the subject's pedagogical weighting is often more effective.

🔗 See also

[Subject](#)

[Solver — how it arbitrates constraints](#)

Large room (several simultaneous lessons)

Source: <help/en/glossary/large-room.md> · id: [glossary.large-room](#) · Updated: 2026-06-25

Also known as : *maximum number of classes · exam room · theater · gym · swimming pool · outdoors*

A **large room** is a **classroom** that can host **several different lessons at the same time**, with different teachers and groups unrelated to each other. You declare it by filling in the **Maximum number of classes** field on the room: this number sets how many distinct lessons can take place there in parallel.

Typical examples: an **exam room**, a **hall**, an **outdoor space** (courtyard, theater, sports field, museum...), a **gym**, a **swimming pool** — all places where several independent activities coexist.

Reserved for specialised rooms

The **Maximum number of classes** field only appears on a room that has a **specialisation**. You therefore first give the room a specialisation (sports, exam, swimming pool, outdoors...), then set its maximum number of

simultaneous classes. This is consistent: a shared large room is always a space specialised for a particular activity, and the specialisation also serves to steer the right subjects to it.

How Omniscol uses it

On each time slot, the [solver](#) and conflict detection allow as many lessons to be placed in the room as the [Maximum number of classes](#) permits, as long as the sum of their headcounts stays within the room's [capacity](#). Beyond either of these ceilings — too many parallel lessons, or capacity exceeded — Omniscol reports a conflict.

A room left without a [Maximum number of classes](#) behaves like an ordinary room: one lesson at a time.

Large room or multi-room?

The two notions play on the room ↔ lesson pair, in two opposite directions:

- a **large room** gathers **several lessons in a single room**;
- a **multi-room** lesson conversely occupies **several rooms for a single lesson** (its capacity is then the sum of the assigned rooms).

See also

[Classroom](#)[Classroom specialisation](#)[Multi-room](#)[Sites, time grids, classrooms, resources](#)

Lesson / Session

Source: [help/en/glossary/lesson.md](#) · id: [glossary.lesson](#) · Updated: 2026-06-13

Also known as : [class period](#) · [slot](#)

A **lesson** (or *course lesson*) is the scheduled unit placed on a timetable grid: it is what you place, move, lock, edit, cancel or substitute. A lesson carries:

- a **class** or a [group](#) (the audience),
- a **subject** ([subject](#)),
- one or more **teachers** ([co-teaching](#)),
- one or more **rooms** (see [multi-room](#)),
- one or more **resources** (mobile projector, tablet cart...),
- a **duration** (usually 1 or 2 time units of the site's grid),
- optionally a **type** (tutorial, practical, lecture, exam, etc., defined via [+ Create](#)),
- optionally a **memo** (free comment, with a publication restriction: administrators only, administrators + teachers, or everyone).

Lesson or course?

The Omniscol interface says "lesson" for the unit placed on the grid. The [course](#) remains the right word for the teaching itself — the subject taught to a class, with its volume and its attributes: you speak of the English course, of course hours, of course days, and one course gives rise to one or more lessons in the timetable. The Type of course label likewise keeps the word "course", and the notion of [complex lessons](#) describes courses that combine several lessons.


Vocabulary by context

- **Lesson** — the interface term for the scheduled unit, whatever the type of institution.
- **Class period** — a common synonym in primary and secondary education.
- **Slot** — refers to the time position on the grid rather than the lesson that occupies it.
- An **adjunct session** is a lesson covered by a visiting (sessional) instructor as opposed to a permanent one; it is not a different course type, only the instructor changes.

Simple lessons, complex courses

A simple lesson = one audience, one teacher, one room, at one time slot.

Complex courses combine several lessons:

- **Concatenated** — two lessons back to back (a 2-hour practical = a 1-hour practical concatenated with another 1-hour practical).
- **Associated** — groups rotating across two consecutive lessons (group A in biology then in physics, group B in physics then in biology).
- **Alternating** — lessons that only recur every other week (week A vs week B), every third week, etc. Configured via  Save.

Lessons and classes outside the time grid

By default, lessons align with the time slots of the site's time grid. But Omniscol also handles lessons that do not fit into it (a 2-hour exam starting at 10:20 a.m. on a grid with a 50-minute step). See [off-grid lessons and classes](#).

See also

[Course](#)[Course type \(tutorial, practical, lecture...\)](#)[Lesson status](#)[Lesson modality](#)[Subject](#)[Complex lessons](#)[Multi-room](#)


Lesson modality


Source: [help/en/glossary/lesson-modality.md](#) · id: [glossary.lesson-modality](#) · Plan: [premium](#) · Updated: 2026-06-25

PREMIUM

Also known as : *in person · in-person · remote lesson · distance learning · hybrid · self study*

PREMIUM

On **Premium** accounts, the **Modality** indicates **how** a [lesson](#) takes place. You choose it in the classroom selector , next to the assignment. Four values, each with its own icon on the lesson:

- **In person** — the lesson takes place in a [classroom](#) (the default case).
- **Remote** — the lesson is held remotely (icon ); no classroom is required.
- **Hybrid** — the lesson takes place in a classroom **and** is broadcast remotely (for example a lecture streamed by video conference): it therefore occupies a classroom and needs a link.
- **Self study** — work carried out independently, with no classroom required. It is equivalent to locking on the absence of a classroom (which disables the alert when no classroom is assigned to the lesson).

Effect on the classroom and generation

The modality has a concrete effect on planning: a **Remote** or **Self study** lesson **does not require a classroom**. The [solver](#) does not reserve one for it, and automatic classroom assignment skips it. Lessons in **In person** and **Hybrid** mode, on the contrary, do require a classroom.

Video conference link

For remote or hybrid lessons, you can enter a **video conference link** in the same classroom selector. A link can also be defined as a default on the class and then apply to its lessons. When no link is set, an alert is displayed.

Statistics by modality

The [diagnostic](#) offers a breakdown table **by modality**: it totals the teaching time delivered in person, remotely, in hybrid mode and in self study. This is useful to measure, for example, the share of remote teaching in a

program.

When the modality is not set explicitly, Omniscol **infers** it from the elements present: a classroom **and** a video conference link make a hybrid lesson; a classroom alone, in-person teaching; a video conference link alone, remote teaching; a lesson explicitly without a classroom, self study. An explicit choice always overrides this inference.

See also

[Editing a lesson](#)

[Status](#)

[Lesson / Session](#)

[Classroom](#)


Lesson status

Source: <help/en/glossary/lesson-status.md> · id: [glossary.lesson-status](#) · Plan: [premium](#) · Updated: 2026-06-25

PREMIUM

Also known as : [course status](#) · [planned](#) · [draft](#) · [canceled lesson](#) · [completed lesson](#) · [done](#)

PREMIUM

On **Premium** accounts, the **Status** indicates where a [lesson](#) stands in the planning cycle. You set it lesson by lesson — from the  **Status** icon — or **in bulk** from the listing mode. Four values:

- **Planned** — Default status for scheduled lessons. This is the state of an ordinary lesson, placed on the grid.
- **Draft** — a lesson in preparation, **ignored by automatic generation** and by the **dashboards**. Handy for preparing a lesson without yet imposing it on the timetable or on the [solver](#).
- **Canceled** — the lesson remains **displayed as canceled** (kept for the record) but is likewise **ignored by automatic generation** and by the **dashboards**.
- **Done** — the lesson is marked as **done**, which includes it with that detail in the teachers' **dashboard**, for **billing**.

Display

When shown on the planning grid, these lessons are recognizable by their rendering:

- a **Draft** lesson carries a **dashed orange border** and stays slightly faded;
- a **Canceled** lesson is **struck through** and heavily faded — it becomes readable again on hover, which lets you keep it for the record without cluttering the grid.

Effect on generation

Draft and **Canceled** lessons are set aside by the [solver](#): it does not place them, and they carry no weight in conflicts (classroom occupancy, teacher collisions, capacities). **Planned** lessons remain the real lessons that generation positions.

"Done": only in calendar mode

The **Done** status is only offered on [calendar](#)-type timetables: that is where, on dated lessons, marking a lesson as actually delivered makes sense. It then feeds the **tracking of the hours teachers actually taught** (teachers' dashboard) and their **billing**. On a weekly or cyclic timetable, this status does not appear.

See also

[Editing a lesson](#)

[Modality](#)

[Lesson / Session](#)

[Calendar mode](#)

Multi-room (one lesson in several rooms)

Source: <help/en/glossary/multi-room.md> · id: [glossary.multi-room](#) · Updated: 2026-06-13

Also known as : [multiple rooms](#) · [room duplication](#) · [room split](#)

Omniscol lets you assign **several rooms** to the same lesson. This feature is available **on all timetable types**: [weekly](#), [cyclic](#) and [calendar](#).

Typical use cases

- **Exams split across lecture halls** — a 200-candidate exam spread over three lecture halls (capacity 70 + 60 + 80) with a single instructor in charge. Capacity is computed as the sum of the assigned rooms.
- **Broadcast lecture** — a lecture in a main lecture hall, streamed by video to a satellite room (for example on another site, or even for an entirely remote cohort).
- **Split practicals without splitting the group** — when a 30-student practical fits in no single computer room, but can be supervised simultaneously in two adjacent rooms (15 + 15) by the same teacher.

Total capacity = sum of the rooms

When you assign several rooms to a lesson, Omniscol adds up their capacities to check that the group fits. No false over-capacity alert is raised as long as the sum is sufficient.

Tip. If you see over-capacity alerts disappear after adding a second room, that is exactly the expected behavior: the sum now covers the headcount.

Limitation to keep in mind

If the sum of the capacities remains **below** the group size, Omniscol still displays a conflict — it is up to the administrator to decide how to resolve it (by adding a room, reducing the group, or accepting the conflict if it is deliberate, for example when you know that not all enrolled students will attend).

See also

[Classroom](#)[Multi-room exams in higher education](#)[Site](#)

Publication / Activation of a timetable

Source: <help/en/glossary/publication.md> · id: [glossary.publication](#) · Updated: 2026-06-13

Also known as : [going live](#) · [activate a timetable](#) · [active weeks](#)

Publishing (or *activating*) a timetable makes it **visible** in the Timetable module everyone consults, on the **weeks of the school year** you choose.

A generated timetable is **not automatically published**. Until it is published, it is a draft: visible only to administrators, in the Timetable management module.

How to publish

On the home page of the Timetable management module:

1. Select the relevant school year.
2. Click [+ Add period](#) (Timetable allocation).
3. On the grid that appears, choose the weeks of each timetable.
4. Save.

One publication per week (standard account)

With a standard account, only **one publication** per week is allowed. If you want to publish two different timetables, they must cover disjoint week ranges.

Simultaneous publications

In Premium — or, exceptionally, on a Standard account with an activation scoped by contract — several timetables can be published on the same weeks and merged when consulted. This case is detailed in [Multiple active timetables in parallel](#) and in the full page about publication.

Calendar timetable: binary publication

A timetable in [calendar](#) mode is not published by week ranges (its lessons are already individually dated): publication is binary — the timetable is published, or it is not.

See also

[Timetable](#)[School year](#)[Publishing a timetable — full page](#)[Multiple active timetables in parallel](#)

Resource

Source: <help/en/glossary/resource.md> · id: [glossary.resource](#) · Updated: 2026-05-10

Also known as : [mobile equipment](#) · [projector](#) · [tablet case](#) · [kit](#)

A **resource** in Omniscol is a piece of **movable** equipment, not attached to a particular classroom. Each resource carries a **name** and an **available quantity**.

Examples: three portable projectors, a case of 30 tablets (the case counts as 1, not 30 — you enter the number of cases, not the number of tablets), a microphone kit, a set of first-aid manikins.

How it is used

When configuring a course, you indicate the resources it requires. The solver guarantees that **simultaneous lessons never request more resources than exist**: if you have 3 portable projectors, no more than 3 simultaneous lessons can each request one.

Difference with classrooms

A [classroom](#) is attached to a [site](#) and never leaves it. A resource is not attached to any classroom (but it is attached to a site, which prevents using a case from site A for a lesson on site B).

Typical cases where no resource is used

Many courses have no declared resource — that is the default. Declare a resource only if it constitutes a **real limit**: if you have 30 portable projectors and at most 4 simultaneous lessons, there is no point in modeling it; if you have 3 and several lessons requesting one, you want the solver to arbitrate.

See also

[Site](#)[Classroom](#)

School year

Source: <help/en/glossary/school-year.md> · id: [glossary.school-year](#) · Updated: 2026-05-10

Also known as : [academic year](#) · [academic session](#)

A **school year** in Omniscol is the reference period over which timetables are published. It is an object declared at school level in School years.

Why it is required

You must have at least **one** school year declared before you can publish a timetable. The school year materializes the **timeline** on which you will then activate timetables for specific ranges.

Naming conventions

For a clean display, name your years by their dates: "2025-2026" rather than "Current year". You can then keep the history of past years and prepare the next one in parallel.

Holidays

You can enter the school year's holidays:

- **by copy-pasting** a table (from Excel, Numbers, etc.),
- **by automatic import** of the dates pre-filled for your country (offered when the year is created if the database knows them),
- **by entering the periods manually**.

Holidays are removed from the timeline — published timetables do not apply to them.

Alternation shifts

If your school uses [alternate weeks](#) (A/B), holidays can disrupt the alternation. You can create a **virtual shift** to handle these cases (click the relevant weeks on the timeline).

Several years in parallel

At any time, you can have several school years defined: the current year, the previous year (history), and the next one in preparation. The [▶ Next school year](#) button (▶) on the timeline switches between years.

The **current year** is set in Current school year. It determines what users see by default.

See also

[Alternate lessons / A/B weeks](#)

[Managing a school year](#)

[Preparing the next school year](#)

Sharing link

Source: <help/en/glossary/share-link.md> · id: [glossary.share-link](#) · Updated: 2026-06-13

Also known as : [shared link](#) · [public link](#) · [token-based sharing](#) · [share link](#)

A **sharing link** in Omniscol is a signed URL granting access to a resource without prior login:

- the timetable of a class or a teacher,
- a targeted entry screen, for example for availability,
- the Absence management module in read-only mode,
- a specific view for an external team (audit, communication).

For public timetables, access is read-only. Some internal sharing links can however allow a limited action, such as filling in a specific form. The typical case is the link for entering teacher availability: the school can send it to a

teacher with a deadline, without creating an Omniscol login for them.

Security

Each link embeds a signed **JWT**. For web and API links, the signature depends on the account that generated the link and on the hash of its password. For iCal links, the verification hash is built on the same principle. Four consequences:


1. **Mandatory expiration date** — the link expires automatically on the chosen date. This is an important safeguard: no eternal links.
2. **Invalidation on password change** — if the bearer account changes its password (or resets it to the same value, which changes the random salt), all the links it generated are invalidated immediately.
3. **Invalidation through the bearer account** — if that account is deactivated or deleted, the links it carried stop working.
4. **Limited scope** — the link only opens the intended view or action. Do not describe it as global access to the account.

For collective distribution, use a clearly identified service account rather than a personal account. The service account becomes the administrative bearer of the issued links and can be rotated or deactivated if a sharing campaign must be invalidated.

Difference with an API token

Aspect	Sharing link	API token
Audience	Human in a browser	Third-party software
Permission	Bounded scope, often read-only	Depends on selected endpoints
Use case	One-off distribution	Software integration
Auth	JWT in the URL	JWT in the <code>Authorization</code> header
Required level	Creator's role	Administration rights

Creating a link

The  **Sharing** button appears on shareable screens (timetables, availability, absences). Click it, choose an expiration date, then copy the URL in the format you need.

See also

[API token](#)

[Sharing a timetable](#)

[Public share links](#)

Site

Source: [help/en/glossary/site.md](#) · id: [glossary.site](#) · Updated: 2026-05-10

Also known as : [location](#) · [main building](#) · [branch site](#)

A **site** in Omniscol represents a physical location where lessons take place. A site hosts [classrooms](#) and [resources](#).

Do not confuse it with the [campus](#), which is an organizational notion: branch, faculty, hub or internal grouping. The site carries the hours, the classrooms and the travel times; the campus is used rather for classifying and filtering.

Each timetable has at least one site. If all your buildings are in the same geographical place, a single site is enough. If your school spans several sites with instructors or students who move around, create as many sites as needed — and **enter the travel time between them**, otherwise the solver may teleport your teachers or your students.

Time grid at site level

Each site carries its own **time grid** (lesson times, breaks, lunch, closures). It is guessed automatically from the information entered, but should be customized by adding the breaks, the lunch break, the closures (Wednesday afternoon, Saturday).

Practical case: two virtual sites for the same location

A sometimes useful (but rare) configuration: creating two "virtual sites" for a single physical location shared between two entities (middle school + high school) that do not have the same time grid but share teachers. The travel time between the two virtual sites is then zero. Be careful to duplicate the shared classrooms in both sites and to enter exclusive opening hours (see [classroom](#)).

When to prefer two accounts over two sites

If two distinct schools have **no instructor in common** (or if the few shared teachers are assigned sequentially to one site then the other on well-defined days), it is often simpler to use two separate Omniscol accounts. Multiple sites become necessary as soon as there is a real circulation of instructors or students to model.

With **Premium**, another option is to publish **two timetables active simultaneously** on the same account (see [Multiple active timetables in parallel](#)), each corresponding to one site, rather than configuring two sites in a single timetable.

Whether with several accounts or several timetables in parallel, it is possible to share teacher and classroom occupancy, with cross-conflict detection (which the planner can disable). With several accounts, contact the Omniscol team to configure the link between them. Only operational, published timetables serve as the source for determining the occupancy of shared entities.

See also

[Classroom](#)[Resource](#)[Campus](#)[Multi-site for higher education](#)

Solver / Automatic generation algorithm

Source: [help/en/glossary/solver.md](#) · id: [glossary.solver](#) · Updated: 2026-06-13

The Omniscol **solver** automatically computes the placement of the lessons of a [weekly](#), [cyclic](#) or [calendar](#) timetable. It places the created lessons that are not yet positioned, can move already positioned lessons if that improves the timetable, and keeps locked lessons in place.

How it works

When you start a generation, Omniscol spins up a dedicated, parallelized computing environment. There is no queue to manage on the school side; initialization often takes around ten seconds before the first visible computations. The result depends on the quality of the data entered: sites, rooms, teachers, groups, hour volumes, availability and incompatibilities. The more numerous or contradictory the constraints, the longer the generation can take or the more partial the result can be.

The engine is a **neuro-symbolic Monte-Carlo metaheuristic** optimization AI: it combines stochastic search, symbolic business rules and constraint scoring. The important point is operational: the solver is designed for highly constrained cases, while reporting impossible situations rather than hiding conflicts.

Hard constraints

The solver strictly respects the constraints that would otherwise make the timetable invalid:

- a teacher cannot be in two places at the same time;
- a class cannot attend two lessons at the same time, except in cases covered by groups in a division;
- availability and constraints at the **unavailable** level block placement;

- a standard room cannot host two lessons at the same moment;
- a large room can host several lessons only within its configured capacity and class-count limits;
- specialised rooms, capacities, material resources and inter-site travel must remain compatible;
- a locked lesson keeps its position.

Optimization constraints

Soft constraints are used to score and improve the solutions found. They are not ignored: they become penalties that the solver tries to reduce.

Frequent examples:

- **undesirable** availability or constraints for a teacher, a room, a subject, a group or a class;
- pedagogical incompatibilities between subjects, for example avoiding a practical before the corresponding lecture, avoiding sport right after a very dense lesson, or enforcing the order of a progression;
- avoiding the same subject twice in the same day for a class;
- minimum or maximum teaching hours per day or per week;
- day balance;
- reducing gaps in the timetables of classes and teachers;
- number of attendance days for teachers, depending on the practices of the school or the country.

Finding a valid solution is the first step. Optimizing the penalties can then become the longest part of the computation.

Availability

- **Weekly timetables** — available.
- **Cyclic timetables** — available.
- **Calendar timetables** — available with dated lessons and a date range.

What to do if the generation fails

If no complete solution is found, Omniscol keeps the best computed timetable and leaves the impossible-to-place lessons in the list of unplaced sticky notes. You can then inspect the partial timetable, read the diagnostic and fix the constraints.

Frequent causes:

- overly restrictive availability (a teacher has no compatible free time slot),
- not enough rooms, or no specialised room available,
- inconsistent alignment (different hour volumes between the aligned groups),
- too many or too strict incompatibilities,
- theoretical class headcount > capacity of all candidate rooms.

See [Diagnosing a failed generation](#).

Forcing the placement of a lesson

You can **lock** the position of a lesson before running the generation again: the solver then keeps it in place, but adjusts the others around it. Handy for anchoring "immovable" lessons (external interventions fixed by contract, dated exams, etc.).

See also

[Weekly timetable](#)

[Availability](#)

[Diagnostic and conflicts](#)

[Automatic generation](#)

Staffing

Source: <help/en/glossary/staffing.md> · id: [glossary.staffing](#) · Options: [staffing](#) · SKU: [omniscol/staffing-only](#) · Updated: 2026-06-13

👑 OPTION: STAFFING

Also known as : [monitoring](#) · [duty grid](#) · [supervision planning](#) · [staff planning](#)

The **Staffing** module schedules **staff by task** rather than by lesson. It was originally designed for student supervision teams: education assistants, pastoral staff, monitors and equivalent personnel who cover, over the course of the day, the playground, reception, corridors, the library, the cafeteria, sports outings, study hall, etc.

The module also handles:

- **exam supervision** (midterm sessions, school-leaving exams, competitive exams),
- **one-off supervision duties** (outings, trips, class councils),
- **active pedagogies** (Montessori) where teachers are scheduled by activity rather than by lesson.

Data model

- **Tasks** = needs to fill, defined by place + time slot + number of people + optional list of authorized staff.
- **Grid** = the set of tasks over a date range, typically with quarter-hour granularity.
- **Assignment** = associating a staff member with a cell of the grid.
- **Duty roster** = per-person view of their tasks.

Independence from lessons

Staffing can be used:

- as a **complement** to the teaching timetables (education-assistant tasks are added alongside the lessons in the account);
- as a **standalone offer**, without any teaching timetable management.

Staff role

The **Staff** role is designed for the student supervision team (pastoral staff, education assistants, monitors, attendants): access to Staffing, to their own absences and to their personal schedule; no access to the global configuration.

It can be combined with the Teacher role when the same person both teaches and is scheduled in Staffing. See [Users and roles](#).

🔗 See also

[Overview of the Staffing module](#)

[Building a service grid](#)

[Defining the tasks to cover](#)

[Assigning staff](#)

[Rosters](#)

Student / Pupil / Learner

Source: <help/en/glossary/student.md> · id: [glossary.student](#) · Updated: 2026-05-14

Also known as : [participant](#) · [trainee](#)

A **student** in Omniscol is an individual enrolled in a [class](#), taking courses, and possibly a member of one or more [groups](#) within that class. **Student** is the canonical interface term; depending on the institution's context, the everyday vocabulary changes:

- **Primary / secondary** — "pupil",

- **Higher education** — "student",
- **Continuing education** — "learner", "participant", "trainee".

All refer to the same Omniscol entity.

Students are optional

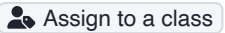
Important point: Omniscol works perfectly well **without any student entered**. Timetables, teacher availability, generation and dashboard statistics operate at class / group / teacher level.

What entering students individually brings:

- **Personalized timetables** — each student sees their own timetable after signing in, filtered by their class and groups.
- **Detailed statistics** — per-student dashboard (teaching hours, attendance days).
- **Absences** — a student can declare their own absences (with administrative validation).
- **Repeating / off-curriculum students** — managing students who must take courses from a class other than their own.

If you do not need these features, do not enter the students — you lose nothing essential.

Assignment to a class and groups

Assigning a student to their class and groups is done in Students, after the first timetable of the year is published. You can also assign in bulk via the  operation on a multiple selection.

Repeating a year and off-curriculum cases

A student can be assigned to several classes (for example, a 3rd-year student who must also take a 2nd-year module) or be assigned only to some groups of a class (a *fuori corso* student retaking a single subject). These cases are handled via the multi-class / multi-group selection of the assignment screen.

See also

[Class](#)

[Group](#)


[Managing students](#)

Subject

Source: help/en/glossary/subject.md · id: [glossary.subject](#) · Updated: 2026-05-10

Also known as : *discipline* · *teaching unit* · *course unit* · *module*

A **subject** is the discipline taught by a **course**: maths, physics, English, accounting law, machine learning, project methodology... The subject is the pure discipline; the course is that subject applied to a specific class.

For each country, Omniscol provides a **catalog of common subjects** (sometimes a few hundred, sometimes a few thousand depending on the official nomenclatures), pre-filled at account level. If a subject is missing, you create a **custom subject** in .

Vocabulary in higher education

The pedagogical nomenclatures of French higher education use different aggregation levels:

- **UE** (teaching unit) — a set of courses / modules,
- **EC** (constituent element) — a sub-unit of a UE,
- **ECUE** (constituent element of a UE) — even finer-grained,
- **module** — the generic term in continuing education.

All these levels map to the Omniscol "subject" entity. If you need to represent the pedagogical hierarchy itself (UE → EC → ECUE), do it through a naming convention (for example prefixing the codes: UE3-EC2-ECUE1 Algèbre linéaire) or through the [subject families](#).

Subject copied into the timetable

When you assign a subject to a class in a timetable, Omniscol makes an **internal copy** of the subject inside the timetable structure. This is a deliberate choice: if you later delete or rename the subject at school level, the existing timetables keep their historical version. Consequently, **check the spelling** of custom subjects before using them in timetables — a fix made afterwards does not propagate to the timetables already configured.

Associated course type

When assigning a subject to a class, you can specify a **course type** (tutorial, practical, exam, lecture...). The (subject, type) pair forms an independent identifier: for example, the "maths course" and the "maths tutorial" are two independent entries in the timetable structure.

Required classroom specialisation

A subject can **require a classroom specialisation** (chemistry → chemistry laboratory, sport → gym). This constraint is respected strictly by the solver. See [classroom specialisation](#).

Pedagogical weight

You can set a **pedagogical weight** on a subject in a class: the solver will then aim to balance the subject across the days of the week rather than concentrating everything.

See also

[Course](#)[Type of course](#)[Lesson / Session](#)[Managing subjects](#)[Classroom specialisation](#)

Substitution / Replacement

Source: [help/en/glossary/substitution.md](#) · id: [glossary.substitution](#) · Updated: 2026-05-14

Also known as : [substitute](#) · [cover](#)

A **substitution** is the assignment of another teacher to a lesson whose regular teacher is [absent](#).

Single-lesson substitution

For a specific lesson, the administrator opens the absence's substitution management panel and uses [Assign a substitute](#). This assignment applies only to the chosen lesson and takes precedence over the long-absence rules.

Substitution rule

For a long absence, the administrator creates one or more rules with [+ Assign a substitute](#). A rule can specify:

- the substitute;
- validity dates;
- time slots;
- subjects;
- classes;
- a comment.

The rules apply in the order displayed. If no rule covers a lesson, it remains affected by the regular teacher's absence.

Displaying substitutions

On the timetable, the substituted teacher remains visible as the absent regular teacher and the substitute appears on the lesson. The details of the absence reason and comment depend on viewing rights.

See also

[Absence](#)

[Substitution policies](#)

[Single-lesson substitution](#)

Teacher / Instructor

Source: <help/en/glossary/teacher.md> · id: [glossary.teacher](#) · Updated: 2026-05-10

Also known as : *faculty* · *lecturer* · *trainer* · *adjunct*

In Omniscol, **teacher** refers to any person who delivers the teaching of a lesson. The term covers:

- **permanent teachers** (tenured, faculty),
- **adjuncts** (occasional external instructors),
- **visiting professors** (guest academics, guest lecturers),
- **trainers** (continuing education, workshops),
- **experts** (occasional contributors hired for their expertise).

Depending on your institution's culture, you may say "teacher", "instructor", "lecturer" or "trainer" — Omniscol adapts to your vocabulary through its labels (see the settings), but under the hood they are all represented by the same entity.

Creation

Teachers are created at school level in Teachers. You fill in:

- first name, last name, login, password,
- e-mail (used for the invitation and for the "forgotten password" process),
- optionally an external identifier (staff number, SIS ID),
- the **subjects taught** (makes assigning them to lessons easier later),
- the **service hours** (planned number of weekly hours, used as the default in new timetables).

Assignment to a timetable

Once created at school level, teachers must be **assigned to a timetable** from . This step distinguishes the "available personnel" (at school level) from the "personnel engaged on this schedule" (at timetable level).


Availability

A teacher can enter their **availability** (unavailable, undesirable or preferred time ranges). For weekly timetables, it requires administrative validation; for calendar timetables, it is consolidated in real time. In secondary education, it is also called **wishes**.

Adjuncts (external instructors)

For **adjuncts**, see the dedicated entry: [External teacher / adjunct](#).

Virtual teacher

The  button creates a *virtual teacher* — a temporary record for a position to be filled, which appears with a ghost next to its name. Once the actual person is recruited, you turn the record into a real teacher with the button on its row. Handy for preparing the new school year before all the recruitments are finalized.

See also

[External teacher \(adjunct\)](#)

[Availability](#)

[Managing teachers](#)

[Assigning teachers to a timetable](#)

Teachers' availability (wishes)

Source: <help/en/glossary/wishes.md> · id: [glossary.wishes](#) · Updated: 2026-06-21

Also known as : [unavailability](#) · [preferences](#) · [teacher availability](#)

A teacher's **availability** tells the software which time slots are **impossible** (black, hard constraints), **undesirable** (red, soft constraints) or **preferred** (green). In secondary education the term **wishes** is also used — it is the same concept, when the teacher enters their theoretical weekly availability and the school uses it for timetable generation (possibly after adjusting or weighting it).

The solver **strictly** respects the impossible slots and does its best to respect the undesirable / preferred ones depending on the other constraints.

Two modes depending on the timetable type

- **Weekly timetable** — availability entered on the typical week, validated by the administration before generation. A later change is flagged but does not interfere.
- **Calendar timetable** (on Premium accounts) — availability entered date by date, consolidated in real time. Conflicts are detected as they arise.

Entry by teachers themselves

Teachers can enter their own availability from their account ([teacher portal](#)) — a major time saver for the administration. They can also receive a **direct share link to the availability screen** (without having to sign in), generated from [Download](#) with an expiry date.

See also

[Teacher](#)

[Solver](#)

[Teacher availability and time constraints](#)

[Availability in calendar mode](#)

Timeline

Source: <help/en/glossary/timeline.md> · id: [glossary.timeline](#) · Updated: 2026-06-29

Also known as : [time bar](#) · [time navigation strip](#) · [week strip](#) · [time strip](#)

The **timeline** is the horizontal strip found at the top of every screen that displays a schedule or statistics. It is how you **navigate through time**: choosing the school year, the displayed period and, depending on the screen, switching between a week, month or year view.

Its **left / right arrows** move back or forward by a whole school year. The displayed year (or period) is named in plain text above the timeline.

[Timeline and time navigation](#)

Timetable

Source: <help/en/glossary/timetable.md> · id: [glossary.timetable](#) · Updated: 2026-06-13

Also known as : [weekly schedule](#) · [course schedule](#) · [master schedule](#) · [lesson planner](#)

A **timetable** in Omniscol represents the complete organization of a school's lessons over a period. It is the central object of the software.

A timetable brings together:

- **classes** with their **groups** and their **divisions**,
- assigned **teachers**,
- **sites** with their time grids, classrooms and resources,
- constraints (availability, incompatibilities, alignments),
- **lessons** (positioned or not).

Three timetable modes

When you create a timetable, you choose its mode:

- **Weekly** — lessons follow a recurring typical week. Typical of primary and secondary education.
- **Cyclic** — lessons recur over an arbitrary number of days (5, 6, 7... depending on your cycle). Typical of North American systems.
- **Calendar** — lessons are dated individually, as in a diary. Typical of higher education (engineering schools, continuing education).

Several timetables per school

You can have several timetables in parallel in the same account:

- unpublished **drafts**,
- timetables for different **periods** (S1 vs S2, preparation of the next school year vs the current timetable),
- with Premium (or, exceptionally, on a Standard account with activation governed by contract), timetables **active simultaneously** over the same weeks (for example a recurring morning timetable + an afternoon timetable in calendar mode) — see [Multiple active timetables in parallel](#).

Timetable life cycle

Creation → Configuration → (Automatic generation?) → Review → Publication → Day-to-day changes

- **Creation** — choice of mode, label, basic settings.
- **Configuration** — sites, classrooms, teachers, classes, groups, subjects, lessons.
- **Generation** — automatic via the solver (available in all three modes: weekly, cyclic, calendar) or manual positioning.
- **Review** — verification before publication.
- **Publication** — the timetable becomes the active timetable on the chosen weeks of the school year. Visible in the *Timetable* module by all authorized users.
- **Day-to-day changes** — throughout the year (moving lessons, classroom changes, absence management) without going through the generation phase again.

See also

[Weekly vs cyclic vs calendar](#)

[Creating a timetable](#)

[Publishing a timetable](#)

Timetable display mode

Source: [help/en/glossary/schedule-view-mode.md](#) · id: [glossary.schedule-view-mode](#) · Updated: 2026-06-13

Also known as : [grid view](#) · [list view](#) · [table view](#) · [switch view](#)

Three interchangeable representations of the same lessons on the same timetable, selected with the buttons in the title banner:

- **Grid** (Grid view): classic calendar view, lessons positioned on a day × time grid. Default mode for quick consultation.
- **List** (List view): chronological sequence of lessons, one per line, with day, times, duration, class, group, subject, teachers, site and classroom. Convenient for browsing sequentially.
- **Table** (Table): spreadsheet-style rendering of the current view — days as columns and time slots as rows (or the reverse), or one lesson per row when the list view is active. Useful for an audit, a copy to a spreadsheet, or a file export depending on the screen (PDF, CSV, Excel / XLSX). This mode is restricted to administrators, on a computer.

The three modes operate on the **same data set**; switching neither modifies nor filters anything, it is strictly a different presentation. The choice is not saved: it applies to the current display.

Transverse course

Source: <help/en/glossary/transverse-course.md> · id: [glossary.transverse-course](#) · Updated: 2026-06-21

Also known as : [transverse courses](#) · [shared course](#) · [common course](#) · [cross-program course](#) · [common core course](#) · [cross-cohort teaching](#)

A **transverse course** brings together students from **several classes, programs or cohorts** around the same course — common core, minors, cross-program electives, cross-cohort lectures.

In Omniscol, it is most often modeled with a **group of groups** (other software calls this a "regrouping"): you aggregate the groups concerned into a single entity, on which you place the course. It then appears in all the parent classes and remains **editable afterwards** — you add or remove a group without touching the structure. This is the preferred option, especially in higher education and in calendar mode.

For a **one-off** need, without creating a named entity, you can also **assign several groups directly to the lesson** from its group selector. Faster, but less readable and less reusable.

When the classes are **parallel and identical in structure** (same hour volumes), especially in a weekly or cyclic timetable, the **alignment** remains relevant: a single course shared between the groups, same time slot, same teacher, same classroom — at the cost of a frozen composition.

See also

[Group of groups](#)

[Group alignment](#)

[Alignments and groups of groups](#)

[Groups of groups](#)

Type of course

Source: <help/en/glossary/lesson-type.md> · id: [glossary.lesson-type](#) · Updated: 2026-07-05

Also known as : [lesson type](#) · [course type](#) · [lecture](#) · [tutorial](#) · [practical work](#) · [lab](#)

A **type of course** qualifies the pedagogical format of a lesson. Canonical examples:

- **Lecture** — taught session for the whole cohort,
- **Tutorial** — guided work in small groups,
- **Practical** — hands-on lab work,
- **Exam** — assessed test,
- **Conference** — outside guest speaker,
- **Workshop, Seminar, Forum, Thesis defense...**

Types of course are defined at school level in [+ Create](#). You can create as many as your pedagogical nomenclature requires (higher-education institutions sometimes have 25-30 types).

COUNTRY: US

In US school usage, nearby labels may be **course type** or status labels such as **core**, **elective** or **AP**. In Omniscol, however, a **type of course** means the **teaching format** of the lesson, not the curricular status of the subject.

COUNTRY: GB

In UK school usage, nearby distinctions may be **compulsory** vs **optional**. In Omniscol, however, this page refers to the **teaching format** of the lesson, not to whether the subject is compulsory or optional.

Subject + type pairing

A type of course is assigned to a course, in addition to its [subject](#). The `(subject, type)` pair is treated as an independent identifier: for example, in statistics or filters, the "maths course" and the "maths tutorial" appear separately.

If you want to query both at once (without distinguishing by type), use an open filter that does not specify the type.

Filtering and display

- **Filter in the timetable module** — select a type to see only practicals, or only lectures.
- **Statistics** — counting tutorial vs practical vs lecture hours per class or per teacher.
- **Display** — color or visual marker per type on the grid, depending on the display settings.

See also

[Course](#)[Subject](#)[Lesson / Session](#)[Configuring types of course](#)

Weekly timetable

Source: [help/en/glossary/weekly-timetable.md](#) · id: [glossary.weekly-timetable](#) · Updated: 2026-05-15

Also known as : *typical week* · *recurring* · *weekly schedule*

A **weekly timetable** is based on a **typical week**: lessons are defined as recurring on the days of the week. At publication, you choose the **week ranges** of the year where the typical week applies.

This is the mode used in:

- primary and secondary education (elementary, middle and high schools),
- undergraduate programs in some faculties,
- language training centers with fixed schedules.

Automatic generation

The automatic generation solver works in this mode. It looks for the optimal placement of lessons within the typical week while respecting:

- teacher availability,
- required classroom specializations,
- divisions, alignments, incompatibilities,
- pedagogical weighting (balancing subjects across the days).

See [Solver](#).

Alternate weeks

With the [alternate weeks](#) feature, you can have lessons that only appear one week out of two (or one out of three, etc.) — weeks A/B, 1/2, etc. Configured in [Save](#).

Holidays and alternation

School holidays can disrupt the A/B alternation. If you were in week A just before the holidays and want to resume in week B afterwards, you can create a **virtual offset** on the timeline (click the weeks concerned).

Availability validated in advance

Teacher availability is validated upstream (before the generation). Any later change by a teacher is reported but does not interfere with the timetable already computed. This differs from the [calendar](#) mode, where availability is consolidated in real time.

See also

[Timetable](#)[Choosing the right timetable type](#)[Automatic generation](#)[Alternate lessons](#)

Index

A

Absence

Definition · Declaring an absence · Overview of the Absences module · Student and resource statistics · Substitution policies · Teacher statistics

Alternate lessons

Definition · Complex lessons · Distribute the hours and create the lessons · General settings

API token

Definition · Integrations overview · Security and hosting

Associated lessons

Definition · Complex lessons · Distribute the hours and create the lessons

C

Calendar mode

Definition · Choosing the right timetable type · Higher education use cases · Module overview · Omniscol plans and options · Overview · Timetable creation

Campus

Definition · Advanced settings and customization · Conflicts and diagnostic · Creating the classes and their groups · Sites, classrooms, resources · Sites, time grids, classrooms, resources

Class

Definition · A five-minute guided tour · Architecture and roles · Class statistics · Class, group, subgroup · Complete data model · Creating the classes and their groups · Data organization · OneRoster · What is Omniscol

Class division

Definition · Class divisions · Class, group, subgroup · Complete data model · Creating the classes and their groups · Group alignments · Group hierarchy · Higher education use cases · OneRoster · Overview · Special cases and advanced configurations

Classroom

Definition · Automatic classroom assignment · Classroom statistics · Editing a lesson · Mass import of courses from a spreadsheet · Off-grid lessons · Overview of the Dashboard module · Preparing your data for a mass import · Search and filters · Sites, classrooms, resources · Sites, time grids, classrooms, resources · Subject statistics

Classroom specialisation

Definition · Automatic classroom assignment · Complete data model · Sites, classrooms, resources · Sites, time grids, classrooms, resources · Special cases and advanced configurations

Co-teaching

Definition · Complex lessons · Higher education use cases

Concatenated lessons

Definition · Complex lessons · Distribute the hours and create the lessons

Conflict

Definition · Ad-hoc changes · Algorithm behavior · Automatic generation · Conflicts and diagnostic · Diagnosing a failed generation · Locking a lesson · Omniscol's general philosophy · Special cases and advanced configurations · What is Omniscol

Course

Definition

Cyclic timetable

Definition · Choosing the right timetable type · Timetable creation

D

Dashboard

Definition · Class statistics · Classroom statistics · Overview of the Dashboard module · Student and resource statistics · Subject statistics · Teacher statistics · Using tables and charts

Date window

Definition

Diagnostic

Definition · Algorithm behavior · Automatic generation · Conflicts and diagnostic · Diagnosing a failed generation

Display panel

Definition · Complete data model · Display and UX

E

External teacher

Definition · Assigning teachers · Higher education use cases · Overview

F

Free group

Definition · Complete data model · Free groups

G

Grade

Definition

Group

Definition · Alignments and groups of groups · Class divisions · Class statistics · Class, group, subgroup · Complete data model · Creating the classes and their groups · Distribute the hours and create the lessons · Editing a lesson · Free groups · Group alignments · Group hierarchy · Groups of groups · Higher education use cases · Off-grid lessons · OneRoster · Student and resource statistics

Group alignment

Definition · Alignments and groups of groups · Class, group, subgroup · Complete data model · Group alignments · Groups of groups · Higher education use cases · Overview

Group of groups

Definition · Alignments and groups of groups · Complete data model · Group alignments · Groups of groups · Higher education use cases · Omniscol plans and options · OneRoster · Overview · Special cases and advanced configurations

H

Holidays

Definition

I

iCal (calendar export)

Definition · Integrations overview

Incompatibility (between subjects)

Definition · Time constraints (general system)

L

Large room

Definition · Automatic classroom assignment · Sites, classrooms, resources · Sites, time grids, classrooms, resources

Lesson / Session

Definition · A five-minute guided tour · Ad-hoc changes · Complete data model · Data organization · Distribute the hours and create the lessons · Editing a lesson · Locking a lesson · Manual placement · Mass import of courses from a spreadsheet · Off-grid lessons · Preparing your data for a mass import · Special cases and advanced configurations · Time grid, time slots and durations · Viewing and filtering · What is Omniscol

M

Modality

Definition

Multi-room

Definition · Higher education use cases · Mass import of courses from a spreadsheet · Overview · Sites, classrooms, resources · Sites, time grids, classrooms, resources · Special cases and advanced configurations

P

Publication / Activation of a timetable

Definition · Guided tour · Publishing (activating) a timetable · Timetable creation

R

Resource

Definition · Complete data model · Editing a lesson · Sites, classrooms, resources · Sites, time grids, classrooms, resources · Student and resource statistics

S

School year

Definition · Complete data model · Data organization · General settings · Guided tour · OneRoster · Prerequisites · Publishing (activating) a timetable · Set up the school account

Sharing link

[Definition](#) · [Architecture and roles](#) · [Integrations overview](#) · [Security and hosting](#)

Site

[Definition](#) · [Automatic classroom assignment](#) · [Complete data model](#) · [Sites, classrooms, resources](#) · [Sites, time grids, classrooms, resources](#) · [Time grid, time slots and durations](#)

Solver

[Definition](#) · [Algorithm behavior](#) · [Automatic generation](#) · [Diagnosing a failed generation](#) · [Locking a lesson](#) · [Manual placement](#) · [Omniscol's general philosophy](#) · [Teacher availability](#) · [Time constraints \(general system\)](#) · [Timetable creation](#) · [What is Omniscol](#)

Staffing

[Definition](#) · [Complete data model](#)

Status

[Definition](#)

Student / Pupil / Learner

[Definition](#) · [Architecture and roles](#) · [Guided tour](#) · [Inviting and activating your users](#) · [OneRoster](#) · [Preparing your data for a mass import](#) · [Student and resource statistics](#) · [Users and roles](#) · [What is Omniscol](#)

Subject

[Definition](#) · [Class statistics](#) · [Complete data model](#) · [Creating the classes and their groups](#) · [Data organization](#) · [Guided tour](#) · [OneRoster](#) · [Overview of the Dashboard module](#) · [Preparing your data for a mass import](#) · [Prerequisites](#) · [Subject statistics](#)

Substitution / Replacement

[Definition](#) · [Complete data model](#) · [Overview of the Absences module](#) · [Substitution policies](#) · [Teacher statistics](#)

T

Teacher

[Definition](#) · [A five-minute guided tour](#) · [Architecture and roles](#) · [Assigning teachers](#) · [Class statistics](#) · [Complete data model](#) · [Data organization](#) · [Editing a lesson](#) · [Guided tour](#) · [Inviting and activating your users](#) · [Off-grid lessons](#) · [OneRoster](#) · [Overview of the Dashboard module](#) · [Preparing your data for a mass import](#) · [Prerequisites](#) · [Subject statistics](#) · [Teacher availability](#) · [Teacher statistics](#) · [Users and roles](#) · [What is Omniscol](#)

Teachers' availability (wishes)

[Definition](#) · [Assigning teachers](#) · [Complete data model](#) · [Groups of groups](#) · [Teacher availability](#) · [Time constraints \(general system\)](#)

Timeline

[Definition](#) · [Timeline and time navigation](#)

Timetable

[Definition](#) · [A five-minute guided tour](#) · [Choosing the right timetable type](#) · [Classroom statistics](#) · [Display and UX](#) · [General settings](#) · [Guided tour](#) · [Module overview](#) · [Off-grid lessons](#) · [Omniscol's general philosophy](#) · [Overview of the Dashboard module](#) · [Publishing \(activating\) a timetable](#) · [Search and filters](#) · [Teacher statistics](#) · [Time grid, time slots and durations](#) · [Timetable creation](#) · [Timetable display](#) · [Using tables and charts](#) · [Viewing and filtering](#) · [What is Omniscol](#)

Timetable display mode

[Definition](#)

Transverse course

[Definition](#)

Type of course

[Definition](#) · [Complete data model](#) · [Data organization](#)

W

Weekly timetable

[Definition](#) · [Algorithm behavior](#) · [Choosing the right timetable type](#) · [Module overview](#) · [Overview](#) · [Timetable creation](#)