

CELCAT® Timetabler Automation Release Notes

Contents

1. Introduction	1
1.1 Reasons to Use Automation	1
2. Overview of Automation Types	2
2.1 Templates.....	2
2.2 Assignments.....	2
2.3 Entries	2
2.4 Constraints	2
3. Overview of Process.....	3
3.1 Specify Constraints.....	3
3.2 Create Templates	3
3.3 Constrain Templates	3
3.4 Run the Automation Engine.....	3
3.5 Evaluate.....	3
3.6 Commit.....	3
4. Template Creation	4
4.1 Templatisation Wizard.....	4
4.2 Course Planner Wizard.....	4
4.3 Template Editor	5
5. Constraint Modification	5
6. The Automation Engine and Evaluator	6

1. Introduction

This document is not a training guide, nor is it a replacement for the help available within the software. Instead, it intends to be a general introduction to the Automation system.

The CELCAT *Timetabler* Automation system generates timetables based on specified parameters. Existing timetables created using the manual system can be used as a source for this parameterisation. Existing course and module relationships can also be used, in addition to creating the parameters from scratch.

Automation allows the entire timetable to be modelled, tested and continually refined, and removes the repetitive nature of searching for available times, staff and rooms. It can lead to better utilisation of space and resources.

1.1 Reasons to Use Automation

Courses are becoming increasingly modular, offering more electives to students. Manually scheduling such electives whilst avoiding potential clashes for students is becoming increasingly difficult.

It takes a lot of effort to prepare the timetable for a department each year, even though the courses offered may have changed little from the previous year. Once a course is entered in Automation using templates and constraints, they can be reused each year with little or no changes required, even if there are significant changes to the staff or rooms available.

The rough layout within each teaching week for a course is often known, and being able to express this in terms of a sequence of events spaced out over the week may be much simpler than actually trying to manually find a suitable non-clashing timeslot for each event in the course, even with the help of *Timetabler* Advisors.

Automation lets you inform *Timetabler* of the sequence of events, and leaves the computer to do the more tedious work of making sure that every event is scheduled and that there are no double-bookings. This is particularly helpful in situations where double-bookings only come to light later on, when it is discovered that some other department has scheduled some of the same resources in a timeslot that was originally thought to be available.

With manual scheduling, it is often difficult to balance requests for rooms from various faculties or departments. This is frequently handled on a first-come, first-served basis. Automation offers the possibility that all requests can be processed at the same time after a significant proportion of the requests have been gathered. Resources can then be more 'fairly' distributed according to needs and the alternatives available in each request.

By requiring departments to enter constraints instead of simply fixing suitable rooms or timeslots, it is possible to see why a department needs a certain room in a given timeslot. Using the manual system exclusively, the room may have been placed in a certain timeslot because historically it has always been at that time, and that normally works every year. In reality, it may be that the constraints indicate that the event needs to be later in the week than some events but earlier than

others, and that a certain item of equipment is required. It may also need to be at a time when students from other departments are available.

With the former method of specifying the room and timeslot, Automation has little choice but to try to honour the request, and flag a conflict if it is not possible. But with the latter method of specifying suitable constraints, the Automation system understands what the user is trying to achieve and has many more options at its disposal to help the user meet his goals without the timeslot and room being fixed in advance.

It is useful to note that the Automation system can coexist with the manual system. For example, some departments may be using the Automation system and working alongside other departments who are still scheduling manually. Even if all departments are using Automation, it is still necessary for ad-hoc room bookings and other events to be firmly bookable in real time, and to be unaffected by any Automation schedule. This is possible, as the Automation system has been designed to fully coexist with the manual system.

2. Overview of Automation Types

The automated system expands on the types already familiar within *Timetabler* in order to accommodate the Automation system. In addition to the existing types of events, resources and classifications, the new types are:

2.1 Templates

Templates are potential events, and do not have times associated with them, since the automation system finds suitable times for you. The final times chosen can be shaped using constraints.

For each event you would like in your final timetable, a corresponding template must be produced. Wizards are available to assist in creation and shaping of multiple similar templates simultaneously. Templates can also be sequenced, forcing them to be scheduled together on the same day or in a specific order through the week.

2.2 Assignments

These are fixed resources assigned to the template, such as modules. Every assignment made to a template will be copied into the final event created, no matter what time or day it is scheduled.

2.3 Entries

These are flexible staff or rooms assigned to the template. The automation system will find suitable staff and rooms for you based on the constraints that you specify and their availability.

2.4 Constraints

These are the parameters that tell the engine what times, staff and rooms are suitable for each template. They can also specify individual resource and global options, such as staff part time preferences, global lunch time settings, sequencing rules across a week or through the year, and concurrency of electives in courses.

In order to produce timetables that reflect your institution's wishes and policies, constraints must be specified.

3. Overview of Process

The creation of timetables using the automated system is a multi-stage process as explained below.

3.1 Specify Constraints

Decide upon and set constraints against your resources to identify availability and preferences. For example, you might stipulate that a member of staff prefers working at certain times in the day. Setting constraints against large batches of resources can be performed in a single operation if necessary.

3.2 Create Templates

Decide what events you want to be created using the Automation system. The choices here will likely reflect your institution's curriculum requirements. Add a template for each event you would like scheduled.

As noted earlier, it is possible for the manual system to still be used alongside the Automation system. For example, you may decide to use the Automation system for the bulk of your timetabling requirements, while the manual system is responsible for a smaller portion.

Decide how each event should be composed, such as the number of staff or rooms required. Add staff entries and room entries to the templates as required.

These steps can be performed together in a wizard, eliminating tedious data entry.

3.3 Constrain Templates

Decide what is suitable for each event. For example, if you know that only certain staff are capable of teaching a certain subject, add constraints against the staff entries stating this. Any constraint can easily be applied to a whole set of templates in a single operation.

Another example is a constraint specifying that it is preferable for the template to run in the afternoon only. Another possibility is to constrain the template's room. Instead of specifying a list of possible rooms, you could specify that the room must have at least 30 workstations.

It is essential that the parameters specified against individual resources, templates and their entries are comprehensive enough to accurately reflect your wishes. Without adequate parameters and direction, the Automation engine is unlikely to produce results that you find satisfactory.

3.4 Run the Automation Engine

Run the scheduling engine against your templates.

3.5 Evaluate

Evaluate the results, and if you feel that the timetables produced require further modification, revisit the templates and constraints.

3.6 Commit

When you feel that the timetables produced are suitable, the results can be committed to the system. These then become visible as scheduled 'normal' events to all users, including those users who are not using Automation ('manual' users).

Depending on the settings used, and access rights that have been set within *Timetabler*, the events can be further manually tweaked even after changes are committed to the database, perhaps in response to last-minute changes or in response to unexpected enrolment numbers.

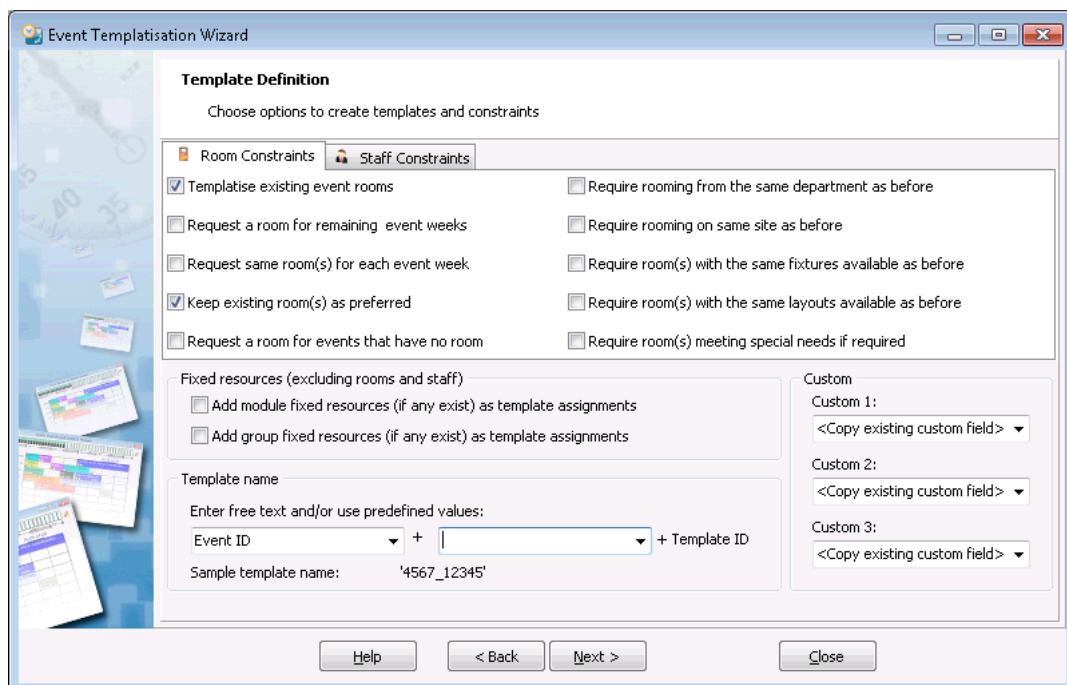
Additionally, any portion of the schedule (such a single course) can be re-run through the Automation engine without affecting any other part of the overall schedule or other courses.

4. Template Creation

Templates can be created from a variety of sources. A series of wizards have been created to facilitate the quick creation of templates.

4.1 Templatification Wizard

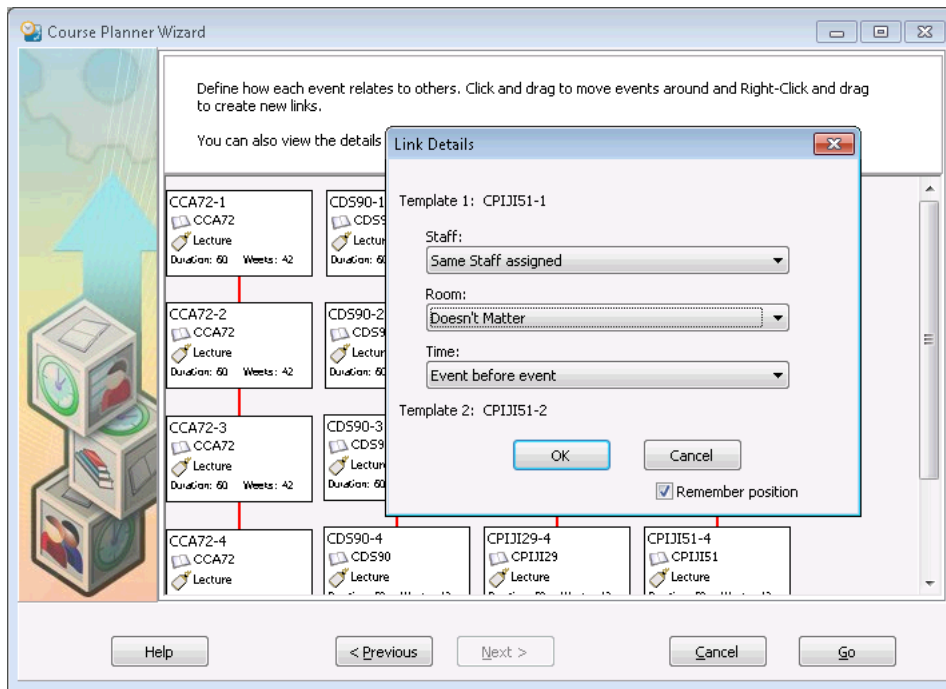
The Templatification Wizard creates templates using existing timetables. This makes it possible to quickly see if your existing timetables can be improved upon.



The first step (selection of events) mirrors *Timetabler's* Event Wizard. Templates can be created using a naming convention for easy location. Additional constraints that may be applied to the templates are chosen towards the end of the wizard.

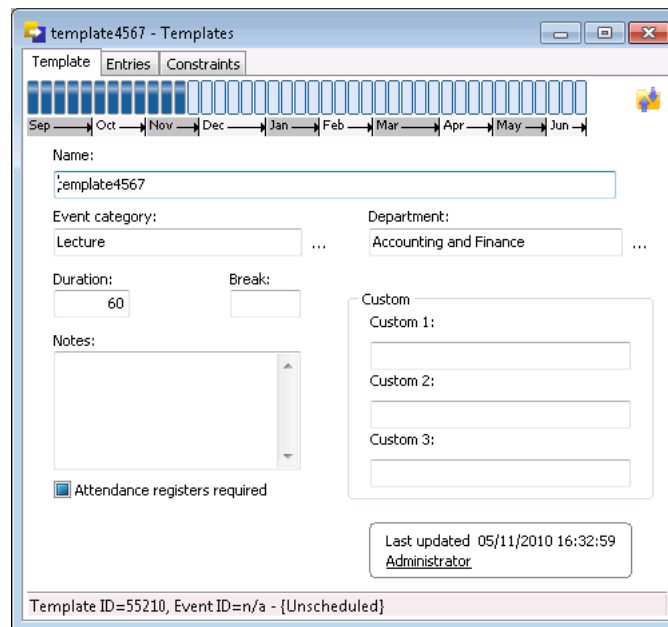
4.2 Course Planner Wizard

An alternative to the Templatification Wizard is to use your existing course and module records and hierarchy as a starting point, rather than your existing events. You can specify the number of templates that each course and module requires, as well as relationships between the templates that it will create.



4.3 Template Editor

The manual creation of templates is also possible. This can be done using the Template Editor window. This record window can also be used to edit existing templates.

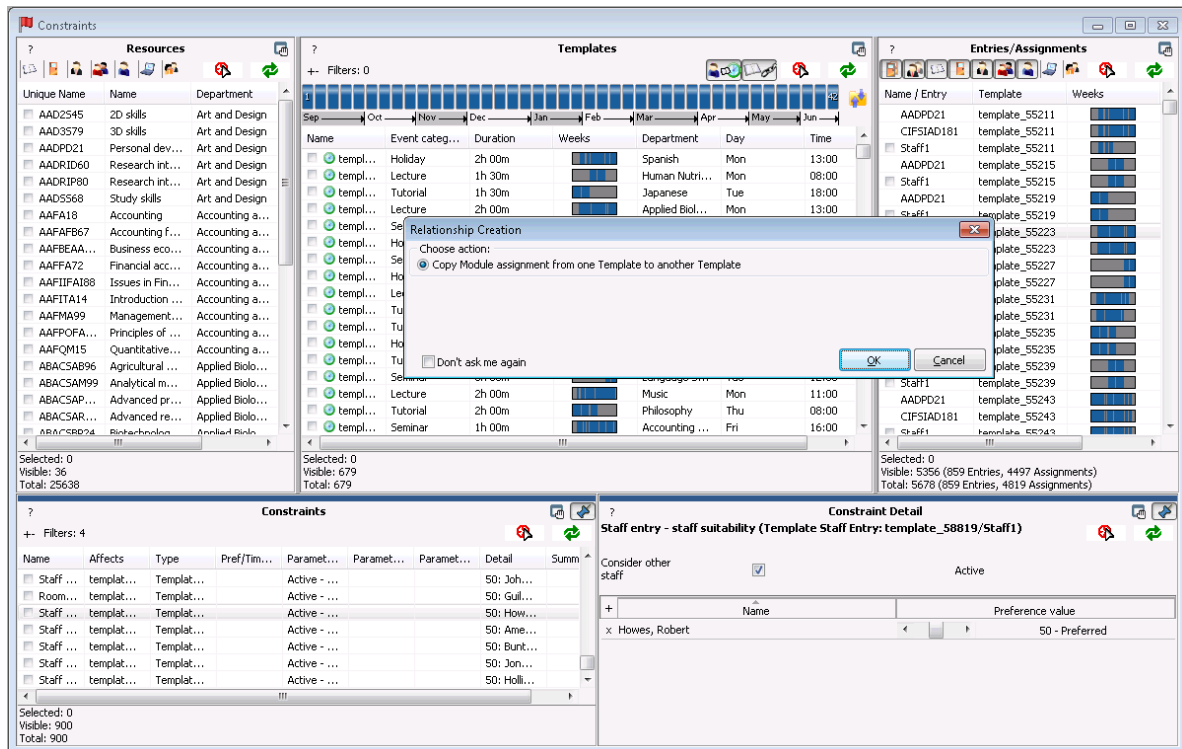


The window shows details about an individual template, including the assignments, entries and constraints associated with it.

5. Constraint Modification

An overview of many templates and their settings can be obtained using the Constraint Editor. It allows you to display templates associated with existing resources, to show constraints that are

applicable to templates, and also the assigned resource and proposed rooms/staff for these templates.



The Constraint Editor also allows modification of templates, assignments, entries and constraints, either individually or en-masse. Five panels house the automation types (resources, templates, entries, constraints and constraint details). A change to one panel affects the other panels in a cascade operation. Therefore, checking items shows only related items in the other lists. The details that are shown within a panel can be filtered using the filter buttons on each panel.

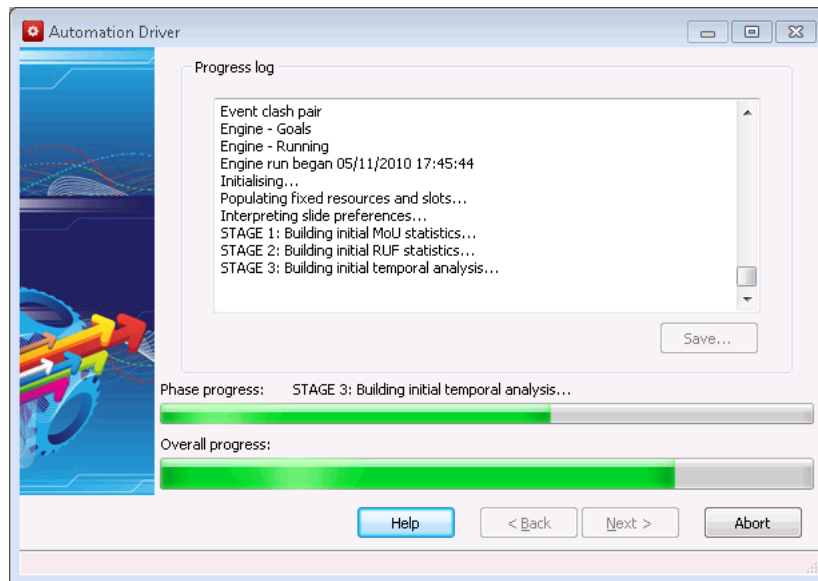
Dragging items from one panel to another gives context-sensitive drag options. Checked items can also be modified in a batch. For example, when working to construct the timetable for a course, you can list all the resources you are likely to want in the resource panel for reference. This enables you to quickly drag and drop to assign resources into single or multiple templates at once (instead of having to keep selecting them from a list) – a real time saver!

6. The Automation Engine and Evaluator

After templates and constraints have been specified, they can be put through the Automation engine. This finds suitable times, rooms and staff for your templates.

It is possible to automatically timetable a small portion of the entire timetable, such as a single course or department, ensuring that the results reflect what you want to achieve, getting the results quickly. Alternatively, it is possible for the Automation engine to be run against all the templates at one time, which would take longer but can be used to reallocate all requested resources and timeslots in a fairer manner than ‘first-come first-served’.

Various drafts of either part or the whole timetable can be run and there is no limit to the number of drafts that you may produce. It is important to note that producing another draft of the timetable takes significantly less time than the manual method of timetabling.

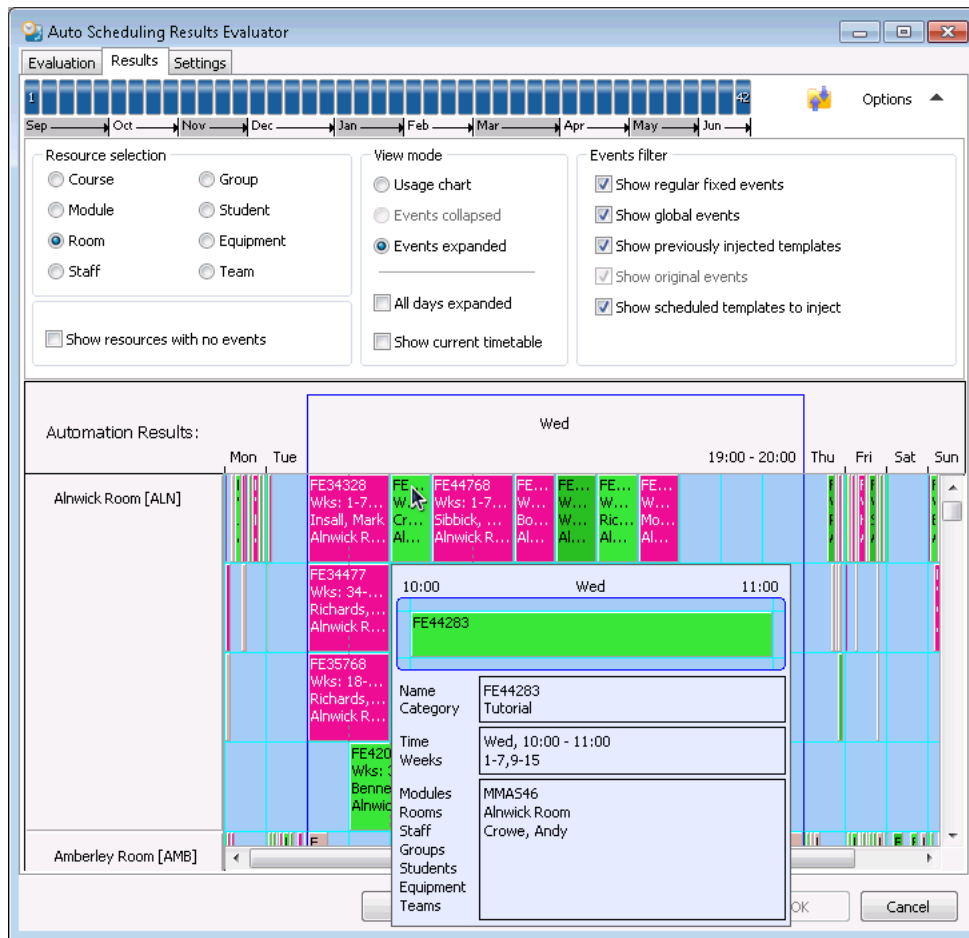


The results of the Automation process are reported after the engine has finished. The success rate is given as a percentage.

The templates that have been scheduled are shown in a grid, not dissimilar to *Timetabler's* usage chart. This view shows proposed timetables for the different resource types, and displays more than one day at a time, with the current day being expanded fully.

The collapsible options bar at the top of the window can be used to filter the events that are shown in the grid, and also the type of grid that is displayed. When hovering the cursor over an event in the evaluator, a pop-up hint displays details in a format similar to *Timetabler's* event window (pictured).

The current timetable for a resource can also be displayed underneath the grid for comparison.



When the results are satisfactory, they can be committed to the main timetable, permitting further modification using the manual system.

If the results require further work, they can be discarded and modification of the templates and constraints should be carried out using the template and constraint editors.