



AnyLogic 6.5.1 New Features

Define recurring patterns with the new object: Schedule	2
New Rail Library: use flowchart objects and 3D animation	3
3D animation of pedestrian models. Improved 2D animation	4
Extended and improved support for CAD files (.DXF)	5
Fully benefit from multi-core processors	6
Paste images directly into the graphical editor	7
Autosave: automatic model backup and recovery	8
Easily change the value of any parameter or variable at runtime	9
Customize the palettes layout	10
New options for custom random number generator	11
Three more ways to create empirical distribution	12
More languages: Chinese, German, Italian (beta version)	13
New example models	14
Other new features and improvements	15

© 2010 XJ Technologies Company Ltd. All Rights Reserved
AnyLogic and XJ Technologies are registered trademarks of XJ Technologies Company



Define recurring patterns with the new object: Schedule

A new object **Schedule** allows you to visually define various recurring patterns with any period and including possible exceptions such as: shifts, timetables, arrival patterns, working hours, and so on. **Schedule** implements mapping from the time axis to boolean, integer, or real value. Daylight saving time is supported if calendar is in use.

The **Schedule** object is integrated with **Source**, and **ResourcePool** from the Enterprise Library, with the **PedSource** object from the Pedestrian Library, and **TrainSource** object from the Rail Library. In addition, you can associate an arbitrary action with events generated by the schedule.

1. Define recurrence interval, value type, and change patterns

Name: Show name Ignore Show at runtime

Value type: on/off integer real

The schedule defines: Intervals (Start, End) Moments

Duration type: Week Days/Weeks Custom (no calendar mapping)

Default value: on off

Repeat schedule weekly

Mon	Tue	Wed	Thu	Fri	Sat	Sun	Start	End	Value
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	9:00	18:00	<input checked="" type="checkbox"/> on
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	13:00	14:00	<input type="checkbox"/> off
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	9:00	14:00	<input checked="" type="checkbox"/> on

Action:

2. Define exceptions (e.g. holidays, vacations)

All day

Annually	Day	Value
<input checked="" type="checkbox"/>	04.11.2010	<input type="checkbox"/> off

3. Preview schedule

	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
0:00																				
1:00																				
2:00																				
3:00																				
4:00																				
5:00																				
6:00																				
7:00																				
8:00																				
9:00	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
10:00	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
11:00	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
12:00																				
13:00																				
14:00	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
15:00																				
16:00																				
17:00																				
18:00																				

Schedule defining working hours with exception and preview

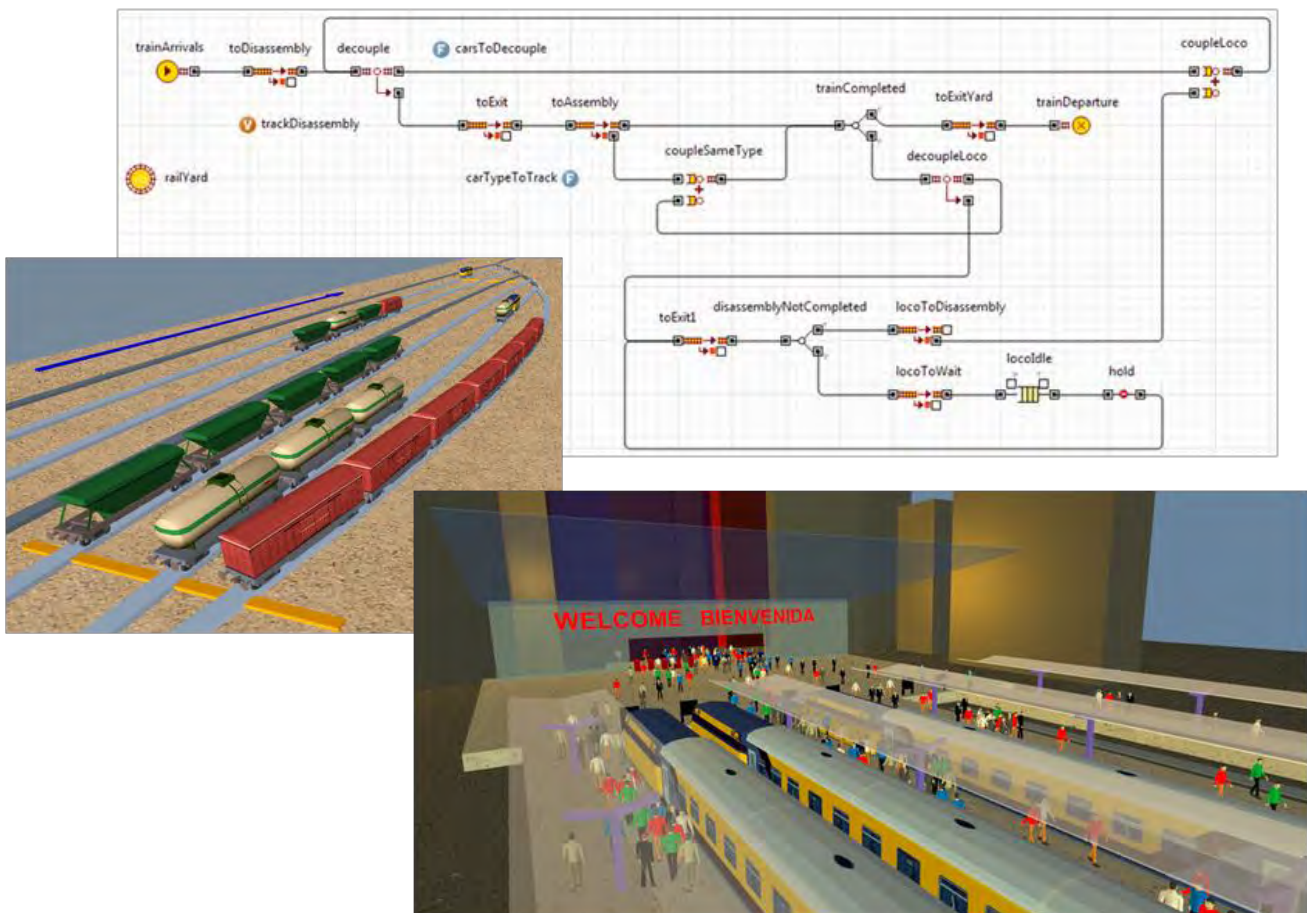
New Rail Library: use flowchart objects and 3D animation

Beginning with AnyLogic version 6.5.1 the Rail Library supports an easy-to-use **flowchart interface** to define the logic of rail system operations in a drag-and-drop style without any coding. It is more flexible and powerful than the low-level interface originally implemented (and still supported). There are five new objects in the library:

- **TrainSource** – creates trains, performs initial setup and puts them in the yard.
- **TrainDispose** – remove trains from the model, including trains that are still in the yard.
- **TrainMoveTo** – controls movement of trains. Can calculate routes and set switch states as the train goes along the route. Supports acceleration and deceleration.
- **TrainCouple** – couples two trains that "touch" each other into one train.
- **TrainDecouple** – decouples cars from the incoming train and creates a new train from those cars.

These five objects can be mixed in a flowchart with objects from the Enterprise Library, such as **Delay**, **SelectOutput**, **Hold**, **Seize**, **Release**, **Queue**, etc. The latter are used to define time delays (such as train stops, loading and unloading), make decisions, and manage sharing of the rail yards resources – tracks and switches.

The AnyLogic version 6.5.1 Rail Library introduces **3D animation** functionality. The rails and the rail cars can now be a part of the model's 3D scene. The 3D Objects palette contains ready-to-use 3D objects for locomotives, several types of freight cars, and passenger cars. Since the Enterprise Library and the Pedestrian Library also support 3D animation, you can now easily create full dynamic 3D models of subway and railway stations, airport shuttles, or any other system where rail transportation and pedestrian flows mix.



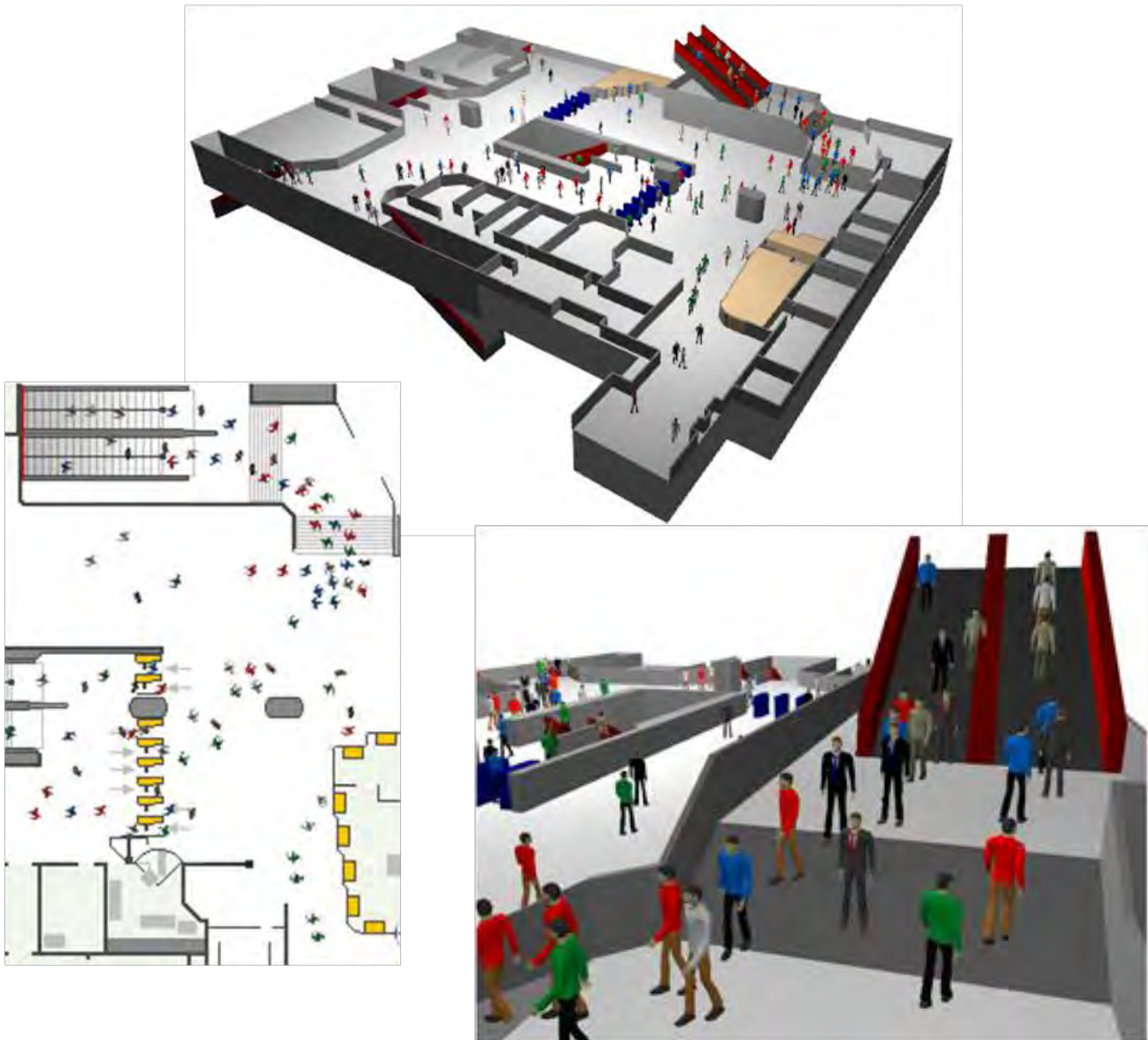
Flowchart of a classification yard. 3D animation of a railway station: Rail Library works together with Pedestrian Library

3D animation of pedestrian models. Improved 2D animation

Beginning with version 6.5.1 the AnyLogic Pedestrian Library supports 3D pedestrian animation. This allows you to create impressive and realistic pedestrian models. You can assign one of the standard AnyLogic 3D objects (such as Person, Worker, Doctor, Nurse), or a custom 3D object to a pedestrian and the object will move in a 3D scene.

Moreover, now you can place pedestrians on terrains or surfaces having different Z levels, or even on slopes. This enables you to accurately visualize multi-story buildings, stairs, and escalators. See the new parameters of **PedGround** and **PedArea** objects.

Another improvement allows you to associate a custom 2D shape with a pedestrian, so in 2D animation you are no longer limited to circles, points, and crosses.

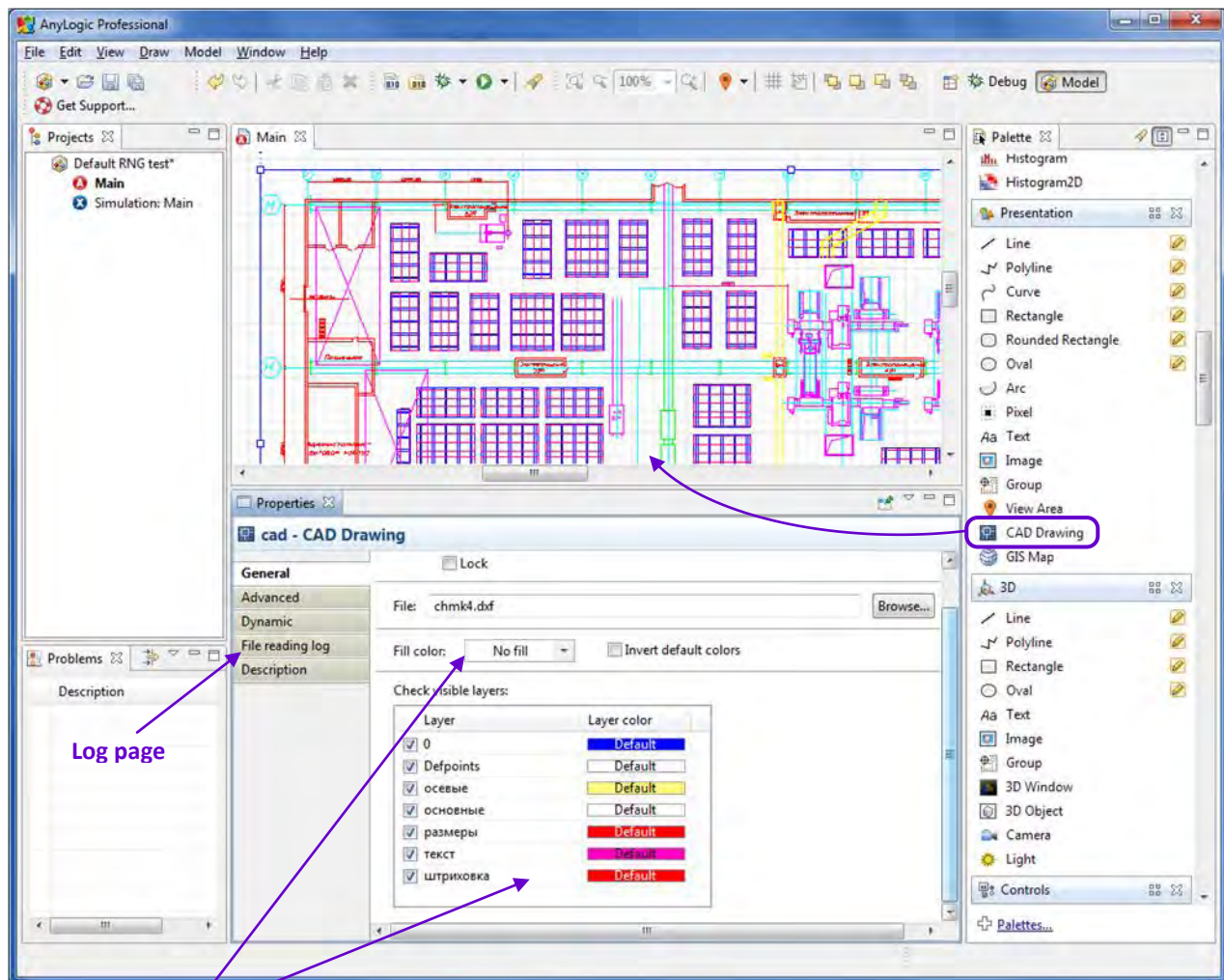


3D and 2D animation of the Subway Entrance Hall model

Extended and improved support for CAD files (.DXF)

AnyLogic now uses a new and improved CAD file renderer (.DXF file extension) and allows you to customize the CAD drawing appearance. The list of improvements includes:

- You can use CAD drawings of any size and complexity.
- Any version of .DXF file will be read and rendered; elements that cannot be recognized are listed in a log.
- More languages and more fonts are supported
- You can choose the background color, for example replace black with white or make it fully transparent
- You can customize the individual layer colors
- Models that include .DXF files are now much smaller in size
- CAD Scaling and rendering are now performed much faster during runtime and design.



Customize background and layer colors

3D animations in AnyLogic models. A new 3D palette

Fully benefit from multi-core processors

If your computer has more than one processor core, AnyLogic will now take advantage of that and place different simulation runs on different cores in parallel. For example, on a machine with four cores a parameter variation experiment now executes 3-4 times faster than on a machine with a single core. This applies to all experiments where multiple simulation runs can be done independently or partially independently, such as:

- Parameter variation
- Monte Carlo
- Sensitivity analysis
- Optimization
- Calibration

By default, AnyLogic will try to use maximum number of processors available. The simulation progress indicator in the status bar of the running model shows as many progress bars as there are simulation runs going in parallel.

The screenshot shows the AnyLogic Professional interface during a calibration experiment. The main window, titled 'Calibration of Agent Based SIR Model', displays a progress chart and a table of current and best values. The status bar at the bottom indicates 'Simulations: 2' with two progress bars, indicating parallel execution. An inset window shows the 'AnyLogic preferences' dialog, specifically the 'Performance settings' section, where the 'Number of processors for parallel execution' is set to 2. A purple arrow points from the 'Simulations: 2' status bar to the 'Number of processors for parallel execution' setting in the preferences dialog.

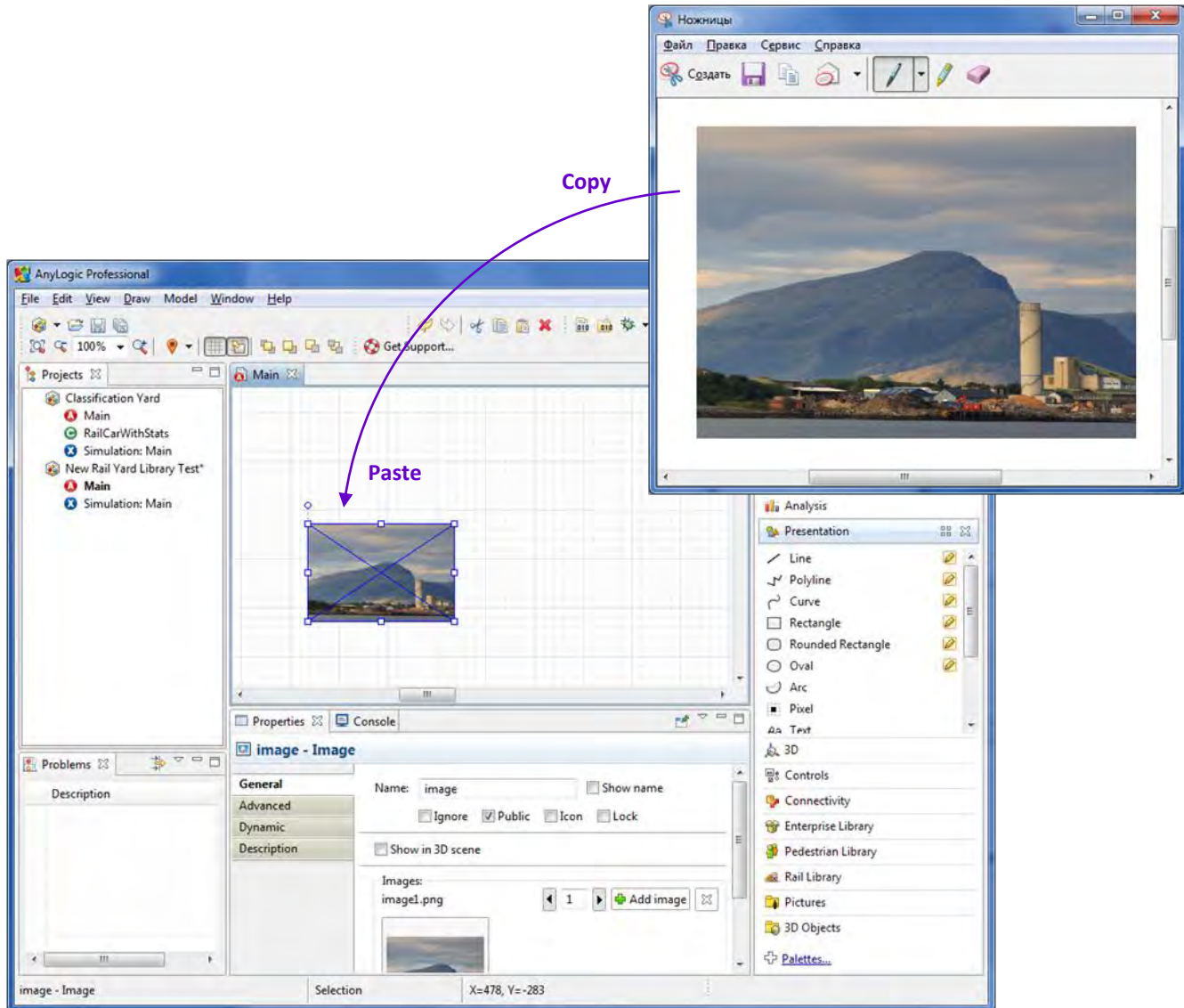
Calibration experiment running on a 2-core processor. Parallel execution settings in AnyLogic Preferences

To determine how many processor cores your system has and to set the numbers of cores to be used by AnyLogic open **AnyLogic Preferences** (main menu item **Window | Preferences**). To prevent parallel execution of simulation runs for all models set the **Number of processors for parallel execution** to 1. You can also prevent parallel execution for a particular experiment by opening the **Advanced** properties of the experiment and uncheck the checkbox **Allow parallel execution** at the very bottom.

Paste images directly into the graphical editor

With AnyLogic 6.5.1 you can paste images directly from the clipboard to the graphical editor. This is a shortcut for the sequence of actions:

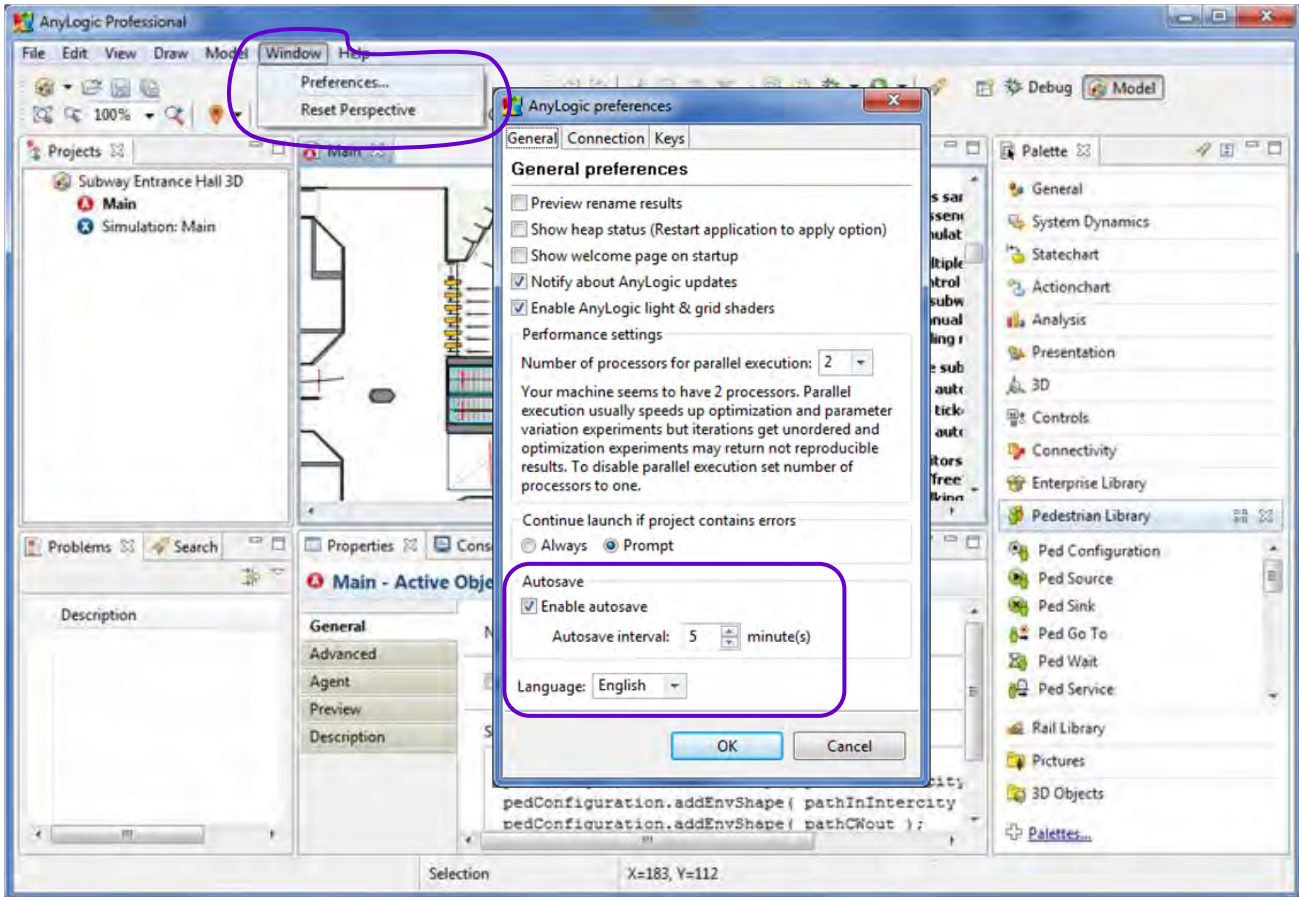
- Save image in a file
- Drag a new **Image** object from the **Presentation** palette
- Open its properties
- Add the image file to the **Image** object



Paste images into the graphical editor

Autosave: automatic model backup and recovery

AnyLogic will now automatically save your models (.alp files) every 5 minutes. If your computer crashes or freezes while you are working AnyLogic will, after restart, offer to revert to the latest autosave file, if it is newer than the original .alp file. The extension for the autosave file is **<your model name>.autosave.alp**. You can turn autosave off and on and change its time interval in **AnyLogic Preferences** (main menu item **Window | Preferences**).



Autosave options

Easily change the value of any parameter or variable at runtime

Now you can change the value of a variable or a parameter at runtime simply by clicking on it and entering a new value. It is no longer necessary to know in advance which variables you are going to change and associate sliders or a text fields with them. The edit field for the new value opens in the inspect window. This is supported for the following objects:

- Plain variable
- Parameter
- Stock

And for the following data types:

- Numeric
- Boolean
- String

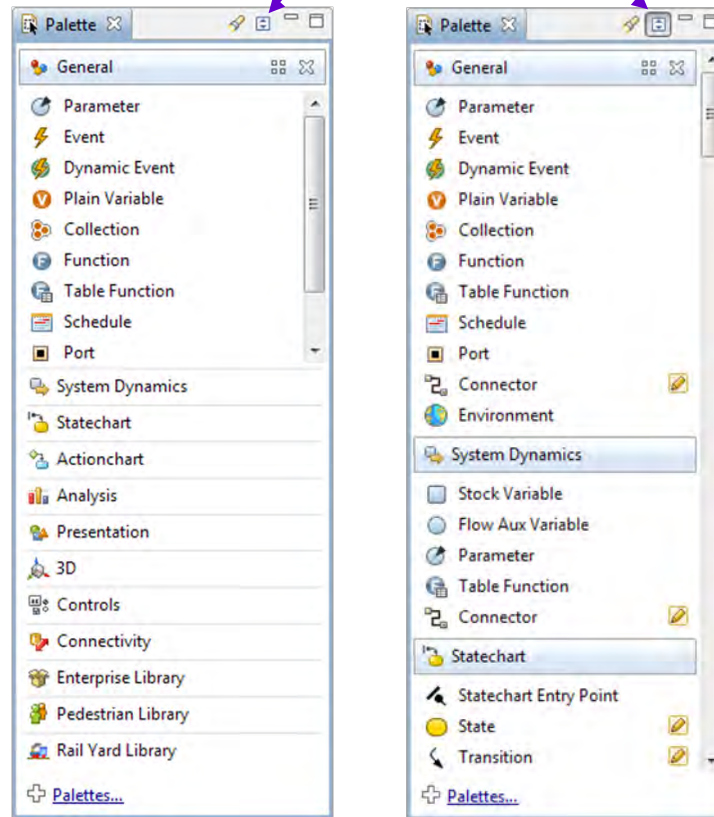
The screenshot displays the AnyLogic Professional software interface. The main window shows a simulation model with various parameters and stocks. The 'averageLifetime' parameter is highlighted in yellow, and its inspect window is open, showing the current value of 67.0. Three numbered arrows point to the inspect window: 1. Click the parameter (pointing to the parameter icon in the model), 2. Choose change (pointing to the 'change' button in the inspect window), and 3. Enter new value (pointing to the text input field in the inspect window). The inspect window also shows the parameter type as 'double'. The simulation is running, and the status bar at the bottom indicates 'Run: 1 Running', 'Simulation: 77%', 'Memory: 349.27511', and '24.5 sec'.

Changing the value of a parameter in the Inspect window

Customize the palettes layout

A new palettes layout is now available: you can expand all palettes and access them as a single ribbon with a scroll bar. To toggle between the compact and expanded layout use the **Expand all palettes** button.

Use Expand all palettes button
to toggle between the layouts

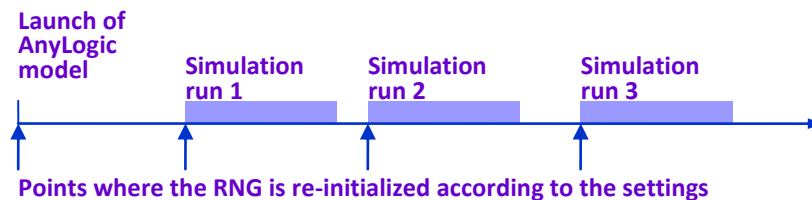
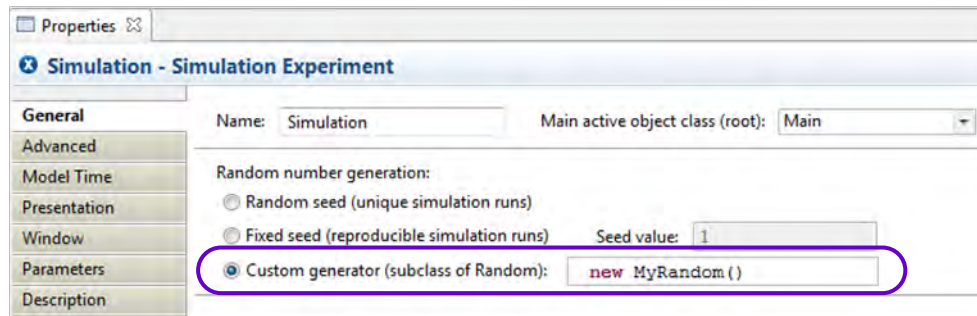


Compact and expanded layouts

New options for custom random number generator

You can now incorporate your own random number generator (RNG) as the default one at the very beginning of the model initialization before the RNG is used for the first time. To set up a custom RNG you would enter the expression that creates and returns it in the **Custom generator** field on the **General** page of the experiment properties. The expression will then be re-evaluated and the result set as default RNG at the beginning of each simulation run. A custom RNG must be a subclass of class **Random** – Java class for random number generators.

The default RNG is the one used by the AnyLogic simulation engine for internal purposes and by the model if the model makes calls to distribution functions not explicitly requiring that another RNG should be used. For example, the call of `exponential(10)` uses the default RNG, and the call of `exponential(10, myOtherRNG)` uses a different RNG.



AnyLogic RNG seed settings and the points when they are applied

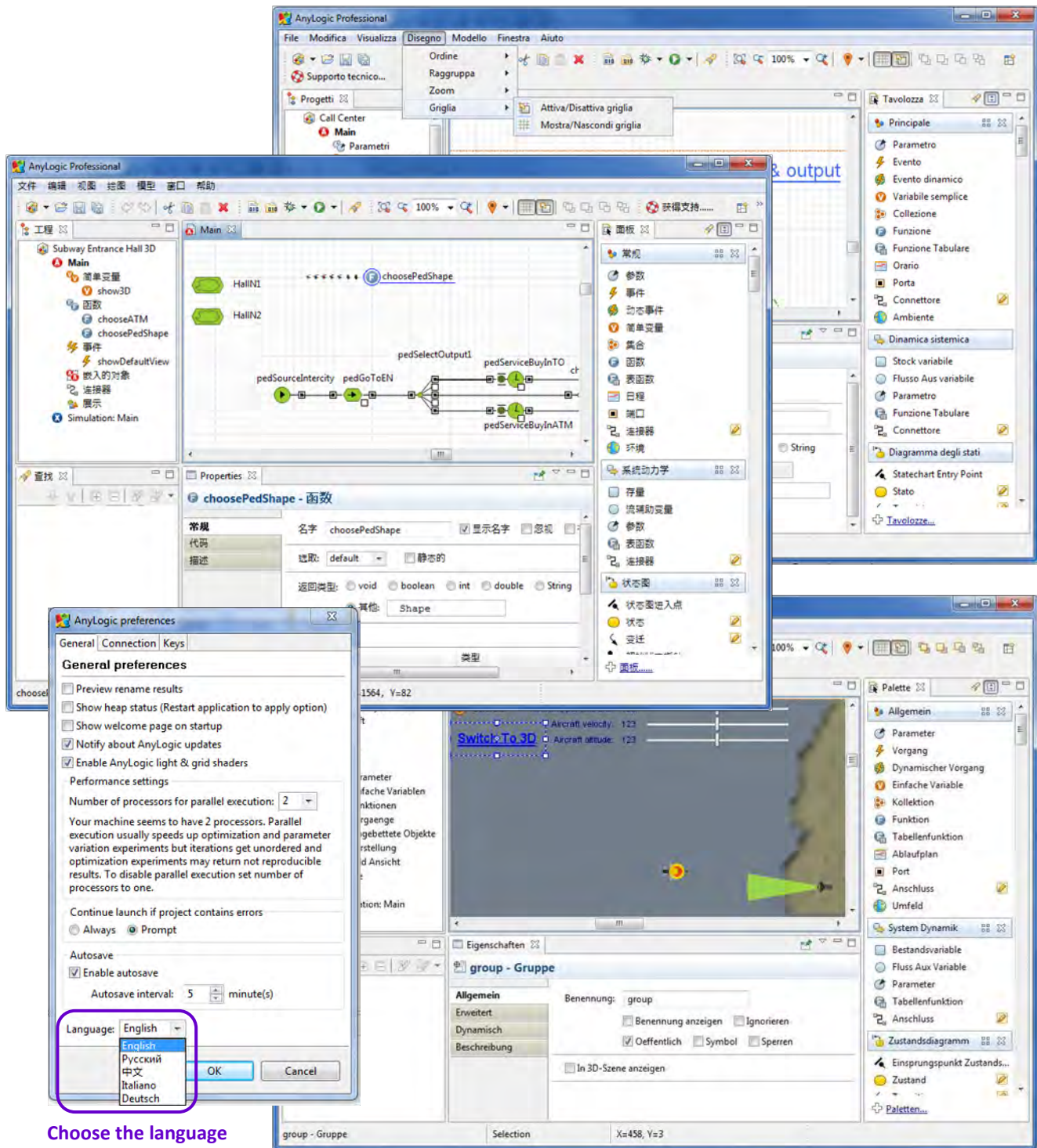
Three more ways to create empirical distribution

In version 6.5.1 we have added three more constructors for empirical distribution (class **CustomDistribution**):

- **CustomDistribution(double[] observations)** – creates an empirical probability distribution based on actual continuous data points (**observations**).
- **CustomDistribution(int[] observations)** – same for integer observations
- **CustomDistribution(double[] intervalstarts, int[] nobervations)** – creates an empirical probability distribution based on a histogram. The histogram should be provided as two arrays: the start points of the intervals followed by the end point of the last interval (**intervalstarts**), and the number of observations in each interval (**nobervations**). The second array must have either one element shorter than the first one or contain 0 as the last element.

More languages: Chinese, German, Italian (beta version)

The AnyLogic user interface is now available in three more languages: Chinese, German, and Italian. The support for these languages is not fully complete, so some items may still appear in English. You can change the language in **AnyLogic Preferences** (main menu item **Window | Preferences**).



Choose the language

AnyLogic UI in three new languages

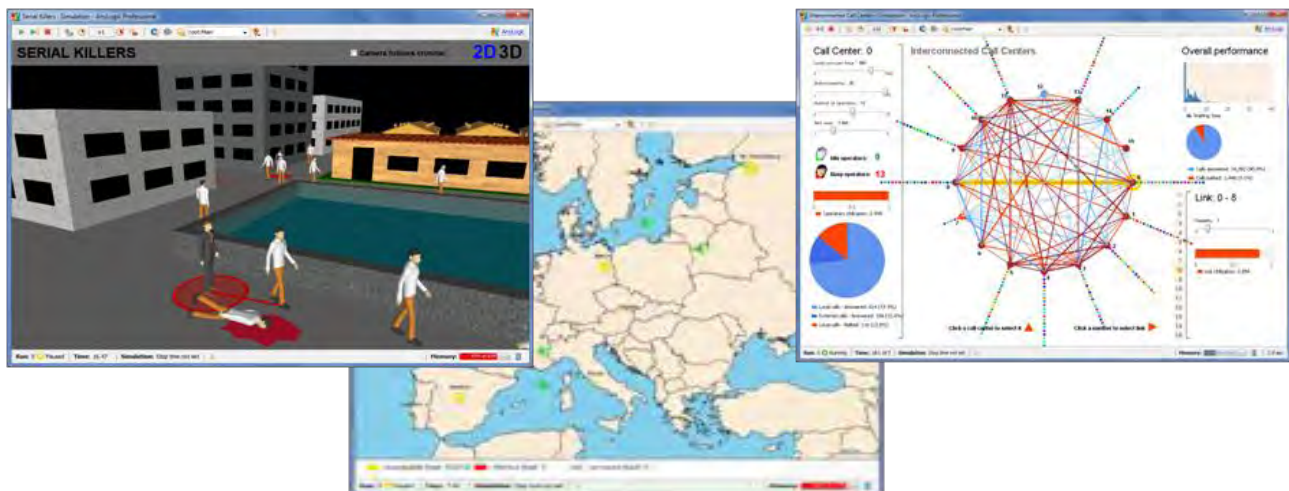
New example models

Main example set (**Examples** folder on the **Welcome page**):

- Serial Killers
- Railway Station
- Restaurant Business
- Interconnected Call Centers

Small how-to models (**How-to Models** folder on the **Welcome page**):

- SIR in GIS Based Space
- Defining a Work Schedule for Resources
- Source Arrival Modes
- Agents Randomly Distributed in a Finite Set of Places
- Agents Randomly Distributed within a Freeform Area
- Click-Sensitive Icon of Active Object
- Custom or Empirical Probability Distribution
- Reproducible Experiment with a Stochastic Process Model
- A Very Simple Rail Yard
- Creating a Rail Yard by Code
- Train Stop
- Ensuring Safe Movement of Trains
- Simple Classification Yard
- Airport Shuttle Train
- Stock Price Fluctuations in a System Dynamics Model
- Pedestrian Groups
- Pedestrian Queues
- Pedestrian Services
- Several Floors 3D
- Controlling Tie Crossing
- Feeding a Collection from Access Database
- Feeding a Replicated Object from Access Database



New example models

Other new features and improvements

Support for Excel 2007/2010 files (extension .xlsx)

The object **Excel file** from the **Connectivity** palette now supports .Excel 2007/2010 files (.xlsx) as well as earlier version binary files (.xls).

Optimization experiments can be exported as standalone applications

Optimization experiments using the OptQuest optimization engine can now be exported as standalone Java applications and run independently from AnyLogic model development environment.

Buy maintenance and support extension directly from AnyLogic

The main menu item **Help | Renew maintenance and support** will take you to the secure online checkout page with the order already set up for your license.

More 3D objects

ATM, shopping cart, cash checkout desk, metal detector, X-ray scanner, 20 feet and 40 feet containers, locomotive, and six types of rail cars have been added to AnyLogic **3D objects** palette.

New item in the main menu: Draw

The commands of the graphical editor (zoom, grouping, Z-order, alignment, and grid settings) have been moved to here.

Alignment in the graphical editor

Shapes and model objects can now be aligned; see the item **Align** in the graphical editor context menu.

Define the size of the model window in the experiment properties

The size of the model window can now be set up on the **Window** property page of the experiment. You can also specify that the window should be maximized on model launch (**Maximized size** option).

Time unit settings moved to Model properties

The setting for model time units (minutes, hours, etc) has been moved from the experiment properties to the model properties. This reflects the fact that models are being developed with a particular choice of time units and it does not make much sense to be able to set different time units in different experiments.

The model stop time is not set by default

The default setting for model stop time (100) has been changed to Never (see the Model Time properties page of the experiment).