

## DROP-OFF ZONE (COM)

### Short description

---

Vissim-integrated scripting example to generate passengers exiting from a car and model a moving barrier.

### Requirements

---

- PTV Vissim Module: COM Interface
- For Python scripting to be available, the following software needs to be installed on your computer:
  - PTV Vision Python (<https://cgi.ptvgroup.com/php/vision-setups/>)  
or a manual installation of Python including pyWin:
  - Python 2.7 or Python 3.7 (<https://www.python.org/downloads/>)
  - pyWin Build 218 or higher (<http://sourceforge.net/projects/pywin32/files/>)

### Objective

---

The Vissim model shows the main entrance of a hotel. Any car stopping in front of the entrance needs to pass a barrier. As soon as no car is present on the layby of the hotel entrance, the next car of the main flow is directed towards the barrier. After passing the barrier, all passengers except for the driver will exit the car and enter the hotel. The car then drives off.

The moving barrier (shown in 3D mode) and the alighting passengers are modelled by COM scripts integrated in the Vissim workspace. In contrast to running scripts from outside Vissim, the integrated method does not restrict the use of the Vissim GUI.

### Modelling Technique

---

This example is provided with various scripting languages. All relevant files for each scripting language are contained in the corresponding subdirectory. The model is prepared to be viewed in 3D mode with the two predefined camera positions "Barrier" and "Drop-off zone".

There are two script files which provide the functionality mentioned above:

- MOVE BARRIER
- GENERATE PASSENGERS

Each of these script files contains several subroutines. The subroutine `Main()` is called from within Vissim and includes resp. refers to all code which is necessary for the script functionality (see below).

For correct initialization of the script variables and objects, the subroutine `Initialization()` needs to be called once at simulation start. While in VBS this is done directly within the script, for Python the subroutine must be called from within Vissim.

## Move Barrier

---

Controls the movement of the barrier 3D object. For the purpose of animation it was modelled in V3DM (Vissim 3D modeller) with 7 states for the different opening angles of the barrier. Shortly after standstill of a vehicle arriving at the barrier, the 3D state of the barrier is increased to trigger the movement. Then in every script time step the barrier state is increased until the final state is reached (= the barrier is open). Similarly the barrier closes after the vehicle has left the connector next to the barrier.

The scripts' attribute **Period** controls the speed of the barrier by defining the interval of when the next state should be switched. If **Period** is set to a value too large, then a warning message is triggered during simulation.

The script uses two user-defined attributes:

- `MainObjNo` to know which static 3D object is the barrier, and
- `RelLinkNo` which refers to the link to look for vehicles arriving at the barrier

If either of these attributes is not given, an error message is triggered.

## Generate Passengers

---

Shortly after car standstill in front of the hotel entrance, for each passenger of the car (but not for the driver) a new pedestrian is generated and set into the Vissim network on a pedestrian area to look as if it was alighting from the car. The number of exiting passengers is recorded in the user-defined link attribute 'AlightPax' while the vehicle is at standstill. This value is visualized as link label in 2D graphics mode just below the car.

The scripts' attribute **Period** may be adapted to see the impact of script running on simulation speed: The higher the period value, the faster the simulation. This is because then the script is only executed every <period>th time step (for example: with **Period** = 10 only every 10th time step). For `GeneratePassengers` an update rate less than each time step is sufficient because the initial waiting time of a car before the first passenger exits and the time between two exiting passengers is set in the script to 2 seconds (constant `ALIGHT_INTERVAL`). If **Period** is set to a value too large, then a warning message is triggered during simulation.

The script uses two user-defined attributes:

- `MainObjNo` to know which area the pedestrians should be generated on, and
- `RelLinkNo` which refers to the link to look for the vehicle they alight from.

If either of these attributes is not given, an error message is triggered.