VISUM-MTM model

Documentation

Author(s): Richárd Horváth dr.

Author’(s’) affiliation (Partner short name): BKK, SZE

Revision History

|  |  |  |
| --- | --- | --- |
| Version No. | Date | Details |
| 1.0 | 14/07/2022 | Initial version |
| 1.1 | 26/10/2022 | Second version (after first test running) |
|  |  |  |
|  |  |  |
|  |  |  |

Contents

[1 Introduction 5](#_Toc88036942)

[1.1 Scope and objectives 5](#_Toc88036943)

[2 Requirements. 5](#_Toc88036944)

[2.1 Software requirements 5](#_Toc88036945)

[2.2 Input/Outputs 5](#_Toc88036946)

[2.2.1 Inputs 5](#_Toc88036947)

[2.2.2 Outputs 5](#_Toc88036948)

[2.3 Paths structure 5](#_Toc88036949)

[3 Model Description 6](#_Toc88036950)

[3.1 Name\_File1 6](#_Toc88036951)

[3.2 NameFile\_2 6](#_Toc88036952)

[4 Instructions to run the model 6](#_Toc88036953)

[4.1 Command line execution of the model 6](#_Toc88036954)

[4.1.1 Instructions and commands 6](#_Toc88036955)

[4.1.2 Arguments 6](#_Toc88036956)

[4.2 Requirements 6](#_Toc88036957)

[4.2.1 Testing requirements 6](#_Toc88036958)

[4.2.2 Folder1 6](#_Toc88036959)

[4.2.3 Folder2 7](#_Toc88036960)

List of tables

[Table 1 Customer.csv – Input 7](#_Toc117703894)

[Table 2 Journey.csv – Input 7](#_Toc117703895)

[Table 3 Vehicle.csv – Input 7](#_Toc117703896)

[Table 4 Vehicle\_type.csv – Input 8](#_Toc117703897)

[Table 5 Result.json – Outputs 8](#_Toc117703898)

[Table 6 VISUM model execution values 9](#_Toc117703899)

# Introduction

## Scope and objectives

This document serves as technical documentation for the VISUM-Macroscopic Transport Model. It explains its scope, the requirements to run the program, the inputs and outputs, and how to access it.

The PTV VISUM is a transport modelling software. BKK, as partner of the LEAD project possess a full scope transport model for the city of Budapest, which is a strategic Macroscopic Transport Model (MTM). The VISUM software is used as a tool to run the MTM. MTM is actually a dataset which contains data in the form of matrices. With the model the passenger and freight traffic can be investigated. The model is available for the public and software independent, but all of the registered usage done by PTV VISUM.

The model can be used to investigate the effects of major modifications in the transport network (new tram line, new bridge, closed road, etc.) or in the territorial development structure (new shopping centre, new university, new residential area, etc.).

The modelling process based on the 4-step modelling methodology:

1. Trip generation: based on statistic information (population, traffic attractive facilities – schools, stores, workplaces etc.) the origin and destination traffic volume are generated for each zones.
2. Trip distribution: the distribution of the trips between the zones based on the distance and other parameters
3. Mode choice: Based on several parameter, like car ownership, access time and cost by different modes, etc. the choice of the mode is happening in this step
4. Assignment: search for the best route for each zone relation and each mode

The 1-3. steps are the part of the demand model, which was integrated to the VISUM in 2019. Before that these steps run out of the VISUM (used Excel sheets, where the calculations are done by predefined Macros.)

The steps above are true for the public and private transportation layers, but the freight transport layer has not been a part of the demand model, yet. For this layer there are fix matrices for 4 different freight transport group (based on the weighs of the vehicles: J1T: < 3,5 t; J2T: 3,5 – 7,5 t; J3T: 7,5 – 12 t; J4T: > 12 t), so with the freight layers happen only the assignment step.

In the first step, the network or the structure is modified in the model. Than user have to decide, that is there any further modification needed in the parameters (for example in the utility function), if yes, the user do it and then choose the appropriate elements of the procedure sequences. After that user start to run the model. The running time based on the number of the scenarios and the capacity of the computer (usually ~2 hours/scenario). The results of the model are the loaded network and many statistic parameters.



Figure 1 The loaded network

In LEAD project, the model is used to examine the effects of different delivery methods in the service area of logistics service provider (WSZL) in connection with the passenger transport of the area. In order to connect the model to the LEAD Platform, a Python based server program was created, because VISUM only runs on Windows platform. This program works as a RESTful server on Windows platform and receives queries and communicates with the other programs, which work on Linux platform. This program transfers the queries to the Visum system. After the Visum assignment steps, the short program sends the results back to the Linux based system.

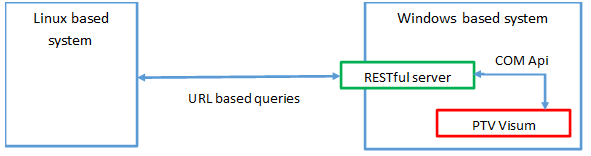


Figure 2 Server program

# Requirements.

## Software requirements

This model requirement the VISUM simulation software and an interface software which communicates the local VISUM with the external user. The interface software uses the Python language and the Flask, Pywin32, PyProj libraries.

## Input/Outputs

### Inputs

The model needs four main parameters as input:

* name of VISUM .ver file. The MTM model of BKK is very large, and it is not necessary to be uploaded, because the infrastructure does not change.
* list of addresses of customers
* list of delivery routes
* used vehicles and their types

Table 1 Customer.csv – Input

|  |  |
| --- | --- |
| Inputs | Description |
| 1. parameter | Customer number |
| 1. parameter | Address – city |
| 1. parameter | Address – street and house number |
| 1. parameter | Address – postcode |
| 1. parameter | Coordinate – latitude in WGS84 format |
| 1. parameter | Coordinate – longitude in WGS84 format |

Table 2 Journey.csv – Input

|  |  |
| --- | --- |
| Inputs | Description |
| 1. parameter | Journey number |
| 1. parameter | Vehicle code |
| 1. parameter | Departure/arrival time |
| 1. parameter | Waiting / service time |
| 1. parameter | Customer number |

Table 3 Vehicle.csv – Input

|  |  |
| --- | --- |
| Inputs | Description |
| 1. parameter | Vehicle no (for VISUM) |
| 1. parameter | Vehicle code |
| 1. parameter | Vehicle description |
| 1. parameter | Vehicle type code |

Table 4 Vehicle\_type.csv – Input

|  |  |
| --- | --- |
| Inputs | Description |
| 1. parameter | Vehicle type code |
| 1. parameter | Vehicle type description |

### Outputs

VISUM generates a file in JSON format, which contains the delivery distances and the type of the vehicle. Table 5 shows the outputs generated in the Excel file.

Table 5 Result.json – Outputs

|  |  |
| --- | --- |
| Outputs | Description |
| 1. parameter | Vehicle code |
| 1. parameter | Distance (km) |

## Paths structure

Root

# Model Description

This section describes the different files and scripts present in the model

|  |  |  |
| --- | --- | --- |
| File name | Location | Description |
| customer.csv | root | File containing the data of shops |
| journey.csv | root | File containing the data of routes |
| vehicle.csv | root | File containing the data of used vehicle |
| vehicle\_type.csv | root | File containing the data of used vehicle types |

# Instructions to run the model

## Command line execution of the model on the server side

The interface program which receives the data and forwards it to the Visum can run on the server.

## Command line execution of the model from the client side

curl --silent --location --request POST "localhost:5000/tupload" ^

--form "file\_customer=@./customer.csv" ^

--form "file\_journey=@./journey.csv" ^

--form "file\_vehicle=@./vehicle.csv" ^

--form "file\_vehicle\_type=@./vehicle\_type.csv" ^

--form "file\_ver=@./test\_model.ver" ^

--output "result.json"'

The BKK-MTM is very large, more than 1.9 Gbytes, so another calling method was created. In this method the .ver file is not necessary, but the name of .ver file is needed.

curl --silent --location --request POST "localhost:5000/tupload" ^

--form "file\_customer=@./customer.csv" ^

--form "file\_journey=@./journey.csv" ^

--form "file\_vehicle=@./vehicle.csv" ^

--form "file\_vehicle\_type=@./vehicle\_type.csv" ^

--form-string "file\_ver=@./test\_model.ver" ^

--output "result.json"'

Table 6 VISUM model execution values

|  |  |  |
| --- | --- | --- |
| Attributes | Possible Values | Description |
| POST | 1.2.3.4:8000 | URL and port of BKK |
| file\_customer | <input\_directory\_path> | Add the input customer file and path |
| file\_journey | <input\_directory\_path> | Add the input journey file and path |
| file\_vehicle | <input\_directory\_path> | Add the input vehicle file and path |
| file\_vehicle\_type | <input\_directory\_path> | Add the input vehicle type file and path |
| file\_ver | <name>.ver | Name of the VISUM .ver file or the name of .ver file on the server of BKK |
| output | <name>.json | Name of the result |

## Requirements

### Folder - Root

The folder contains the needed data file (customer, journey, vehicle and vehicle\_type) for the model.