



CAMBRIDGE
SYSTEMATICS

Think  Forward

Aimsun Model Review Training

Day 1: View Modes and Scripting

Caltrans On-Call Traffic Simulation Training

presented to

Caltrans District 7



presented by

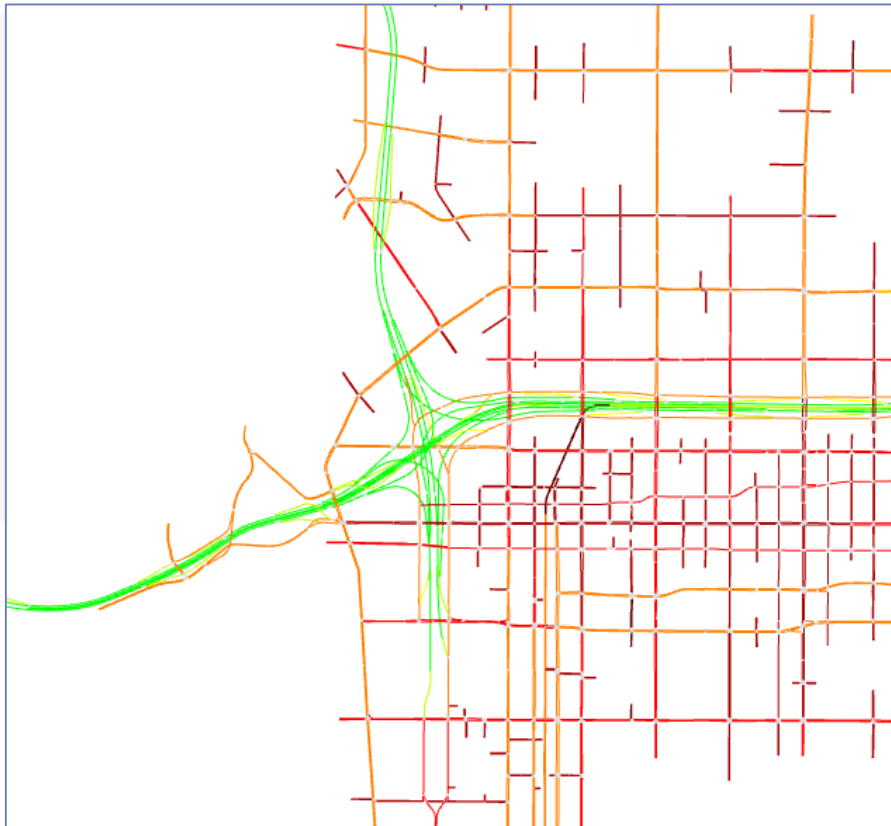
Cambridge Systematics & Aimsun

Keir Opie

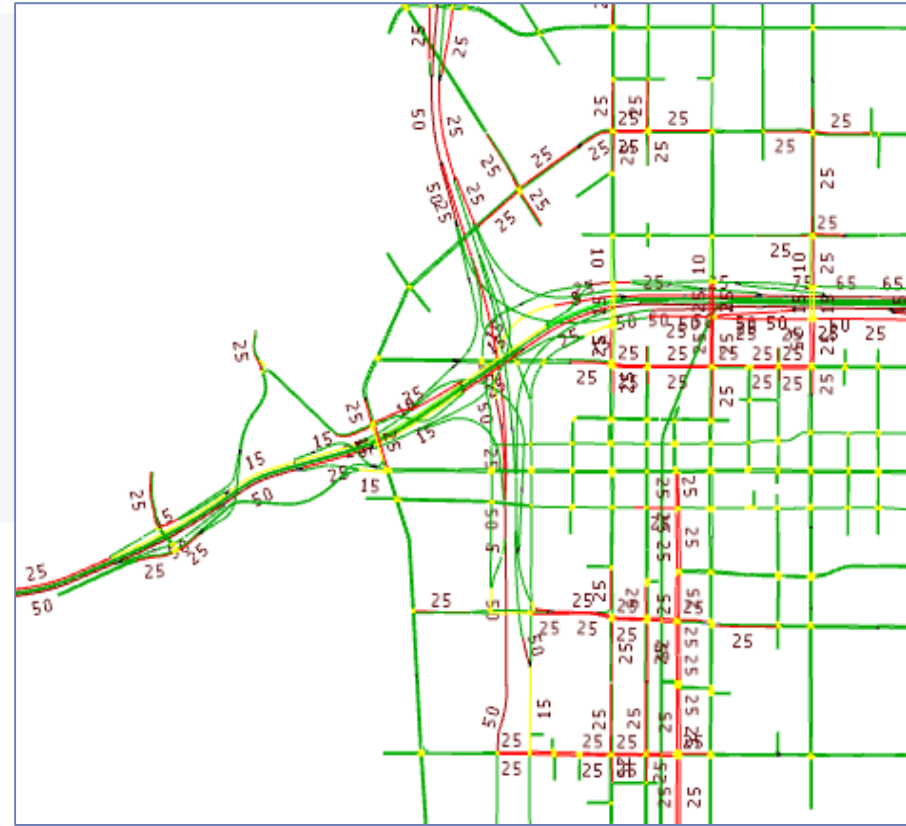
Laura Torres

View Modes

View Modes



Speed Limit

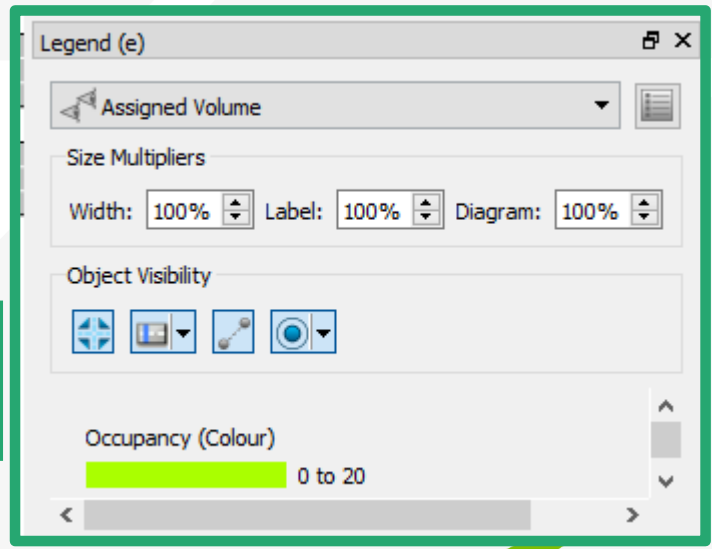
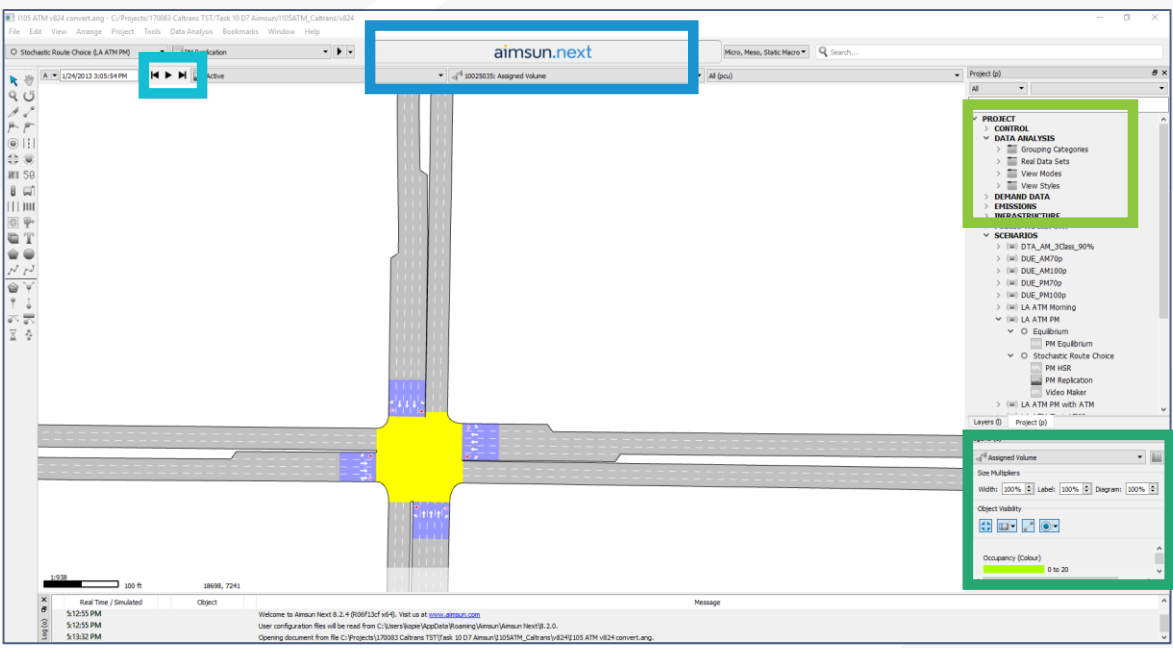
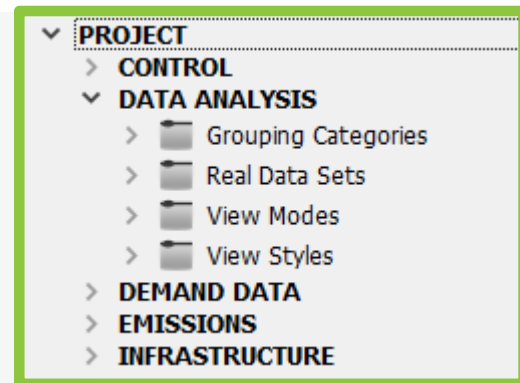


Aggressiveness

View Modes

- Used for
 - » Visualizing model data of most any element
 - » Static or Dynamic data
- View mode = a set of view styles
 - » Setup View Styles
 - » Assemble View Styles into a View Mode
- Options:
 - » Lane level or link level
 - » Zoom levels

View Mode Dialogs



Styles:

- Colour
- Pen Width
- Pen Style
- Brush Style
- Spot
- Hide
- Colour Opacity
- Attribute
- Diagram

View Modes

➔ Example 1: Roadway Speed Limits

View Style: 11396913, Type Section, Attribute Maximum Speed

Main Conditions







Name: PARAMETERS: Section Speed (Colour) External ID:

Objects of Type: Section Style: Color

Attribute: Maximum Speed Normalise

Divided by: None

Continuous Discrete Size Units: pixels Format: 1 Decimals

Symbol	Range	Label
	0.00 - 6.21	
	6.21 - 24.85	
	24.85 - 37.28	
	37.28 - 49.71	
	49.71 - 62.14	
	62.14 - inf	

Symbol Editor

Color: Color Ramp: Do Not Use the Ramp

Help OK Cancel

View Style: 11396905, Type Node, Attribute ID

Main Conditions


Name: GENERALS: Grey Nodes External ID:

Objects of Type: Node Style: Color

Attribute: ID Normalise

Divided by: None

Continuous Discrete Size Units: pixels Format: 1 Decimals

Symbol	Range	Label
	0.00 - inf	

Symbol Editor

Color: Color Ramp: Do Not Use the Ramp

Help OK Cancel

1. View Styles Setup

View Modes

➔ Example 1: Roadway Speed Limits

View Mode: 11396912, Name: PARAMETERS: Section Speed {ca8b275a-bb2a-437e-811a-671058d8f2ad}

Name: PARAMETERS: Section Speed External ID:

Options
Automatic Activation: Never

Drawing
 Draw Spots below their Object
 Objects Drawn on Top by Ascending GENERAL: Grey Nodes

Used Styles

Name	Type	Attribute	Representation	Position	Show in Legend
GENERAL: Grey Nodes	Node	ID	Color		<input checked="" type="checkbox"/>
PARAMETERS: Section Speed (Colour)	Section	Maximum Speed	Color		<input checked="" type="checkbox"/>

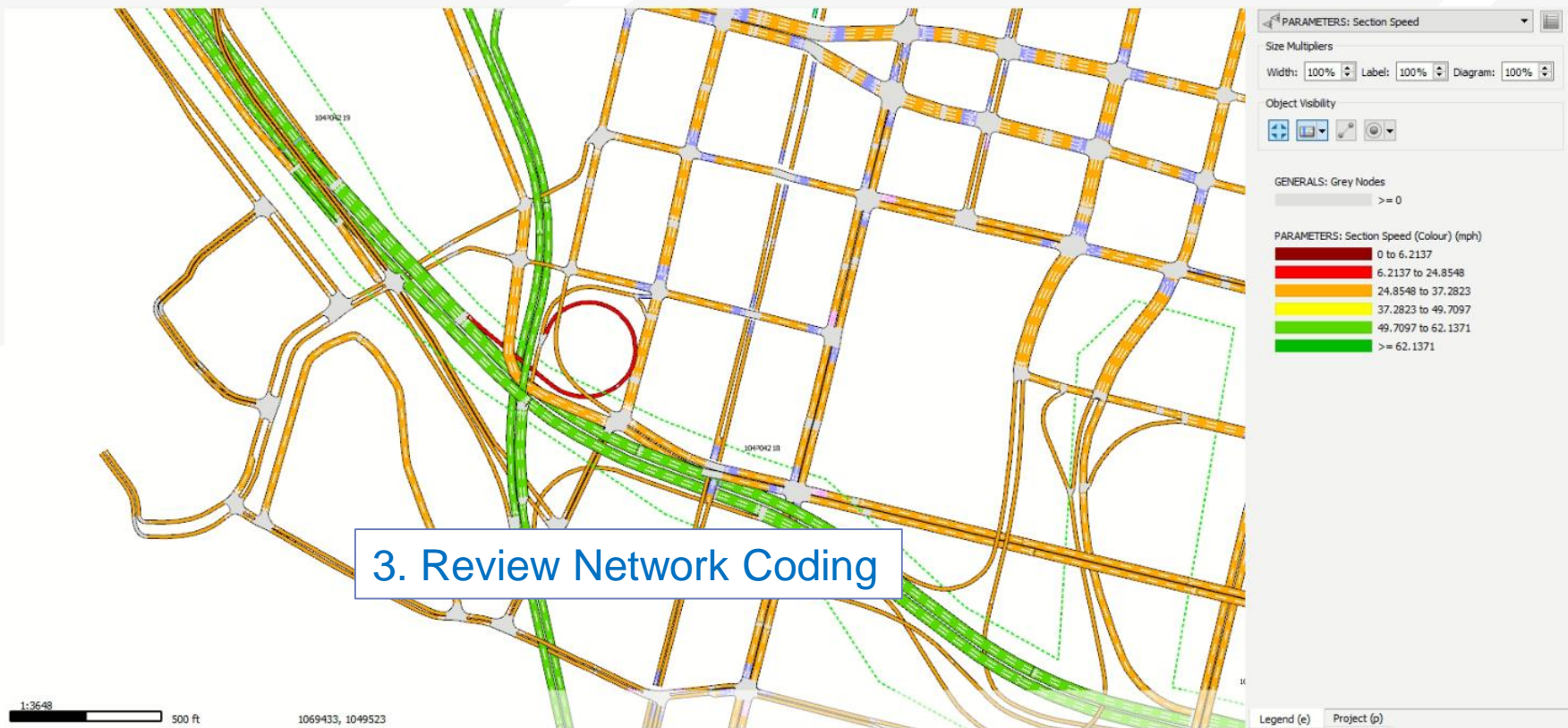
Add New Style
Add Existing Style
Remove Style

2. Assemble View Modes

Hidden Objects

View Modes

➔ Example 1: Roadway Speed Limits



View Modes

➔ Example 2: Roadway Type

View Style: 93, Type Section, Attribute Road Type

Main Conditions











Name: PARAMETERS: Section Road Type External ID:

Objects of Type: Section Style: Color

Attribute: Road Type Normalise

Divided by: None

Continuous Discrete Size Units: pixels Format: 1 Decimals

Symbol	Value	Label
	8	8: Arterial
	32	32: Freeway
	34	34: On/Off Ramp
	36	36: Pedestrian Area
	38	38: Ringroad
	40	40: Road
	42	42: Roundabout
	44	44: Street
	46	46: Urban Road
	94	94: Signalised Street

New Delete

Calculate Ranges

Symbol Editor

Color: Color Ramp: Do Not Use the Ramp

1. View Styles Setup

View Modes

➔ Example 2: Roadway Type



View Modes

➔ Example 3: Simulated Density

View Style: 384, Type Section, Attribute Density - All - Last Generated

Main Conditions

Name: Simulated Density (Colour) External ID: []

Objects of Type: Section Style: Color

Attribute: Density [] Normalise

Divided by: Alone

Continuous Discrete Size Units: pixels Format: 1 Decimals

Symbol	Range	Label
[Green]	0.00 - 32.19	
[Light Green]	32.19 - 64.37	
[Yellow]	64.37 - 96.56	
[Orange]	96.56 - 128.75	
[Red-Orange]	128.75 - 160.93	
[Red]	160.93 - 193.12	
[Dark Red]	193.12 - inf	

Symbol Editor

Color Ramp: Do Not Use the Ramp

View Style: 433, Type Section, Attribute Density - All - Last Generated

Main Conditions

Name: Simulated Density (Label) External ID: []

Objects of Type: Section Style: Attribute

Attribute: Density [] Normalise

Divided by: Alone

Continuous Discrete Size Units: pixels Format: No Decimals [] Percentage

Label Size: 12 []

Color [] Show Attribute Name [] Show Value Label

1. View Styles Setup

View Modes

➔ Example 3: Simulated Density

View Mode: 383, Name: Simulated Density {40b6c3de-300c-49de-8892-fc197aa36979}

Name: External ID:

Options

Automatic Activation:

Drawing

Draw Spots below their Object

Objects Drawn on Top by

Used Styles

Name	Type	Attribute	Representation	Position	Show in Legend
Simulated Density (Colour)	Section	Density - All - Last Generated	Color		<input checked="" type="checkbox"/>
Simulated Density (Label)	Section	Density - All - Last Generated	Attribute	Centre	<input checked="" type="checkbox"/>
Simulated Density (Width)	Section	Density - All - Last Generated	Pen Width		<input type="checkbox"/>
Simulated Density x Lane (Colour)	Section	Density - by Lane - All - Last Generated	Color		<input type="checkbox"/>

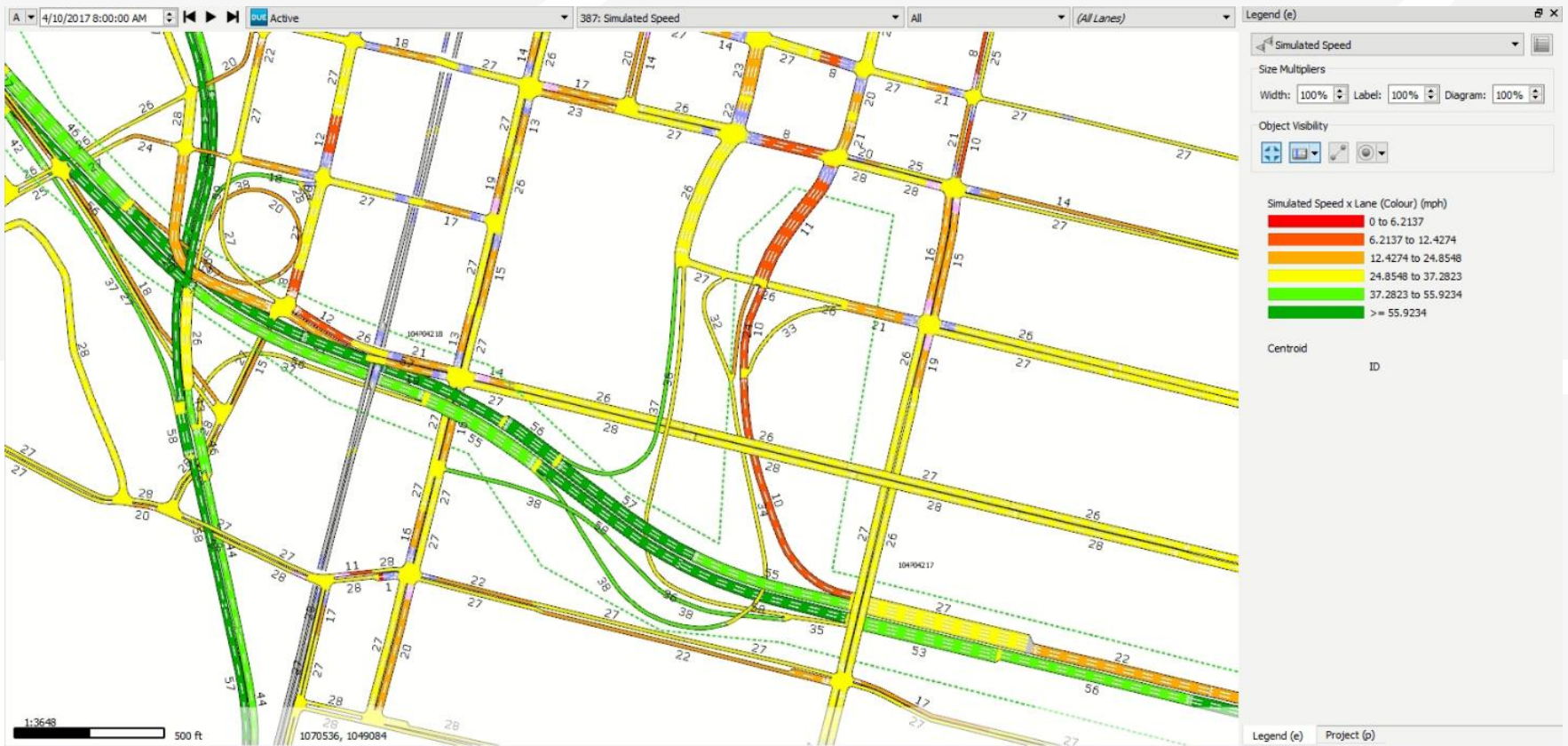
Add New Style
Add Existing Style
Remove Style

Hidden Objects

2. Assemble View Modes

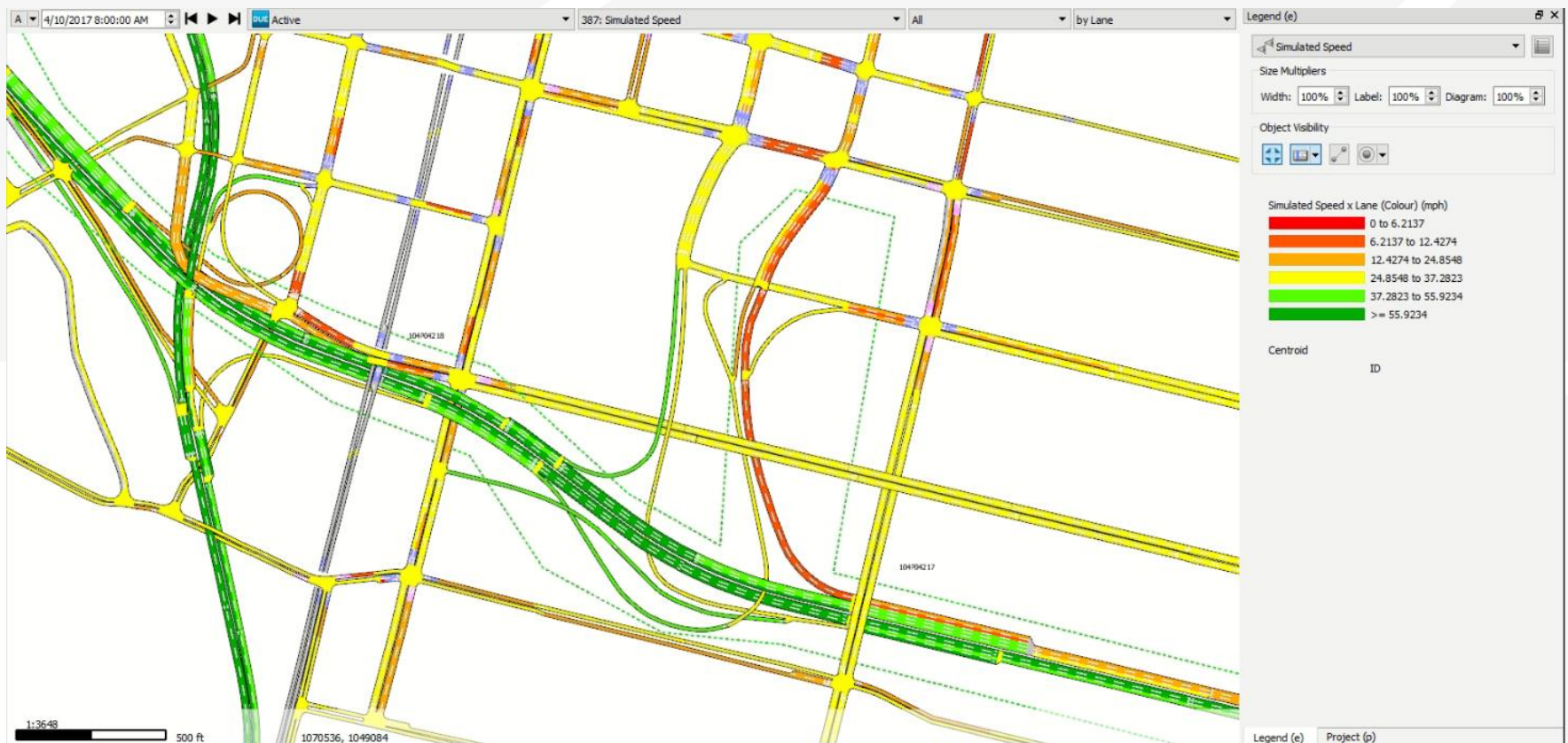
View Modes

➔ Example 3: Simulated Density (All lane)



View Modes

➔ Example 3: Simulated Density (By lane)



View Modes

Example 4: Vehicle Speed

View Style: 2782, Type Simulation Vehicle, Attribute Speed

Main Conditions










Name: Vehicles speed - style 1 External ID:

Objects of Type: Simulation Vehicle Style: Colour

Attribute: Speed Normalise

Divided by: None

Continuous Discrete Size Units: pixels Format: 1 Decimal

Symbol	Range	Label
	0.00 - 10.00	
	10.00 - 20.00	
	20.00 - 30.00	
	30.00 - 40.00	
	40.00 - 50.00	
	50.00 - 60.00	
	60.00 - 70.00	
	70.00 - 80.00	
	80.00 - 90.00	

New Delete

Calculate Ranges

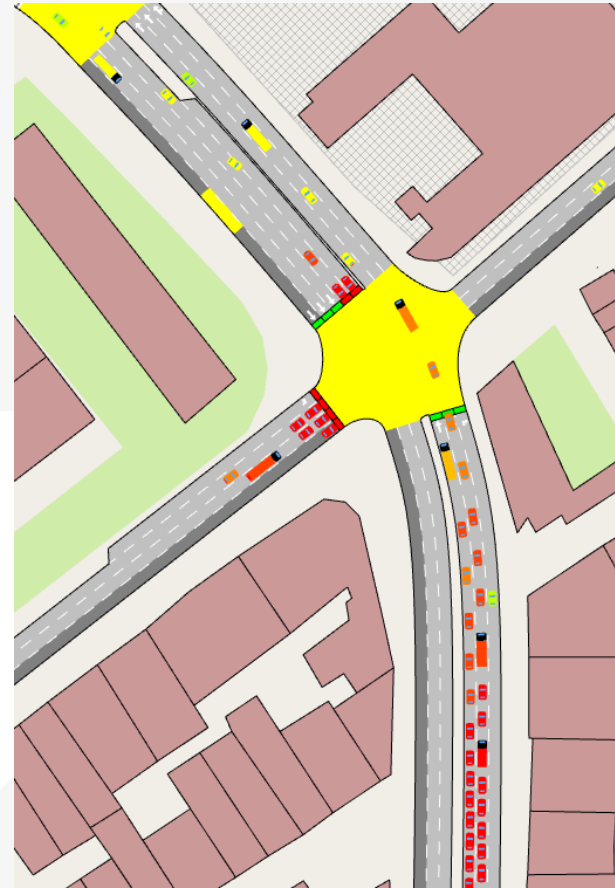
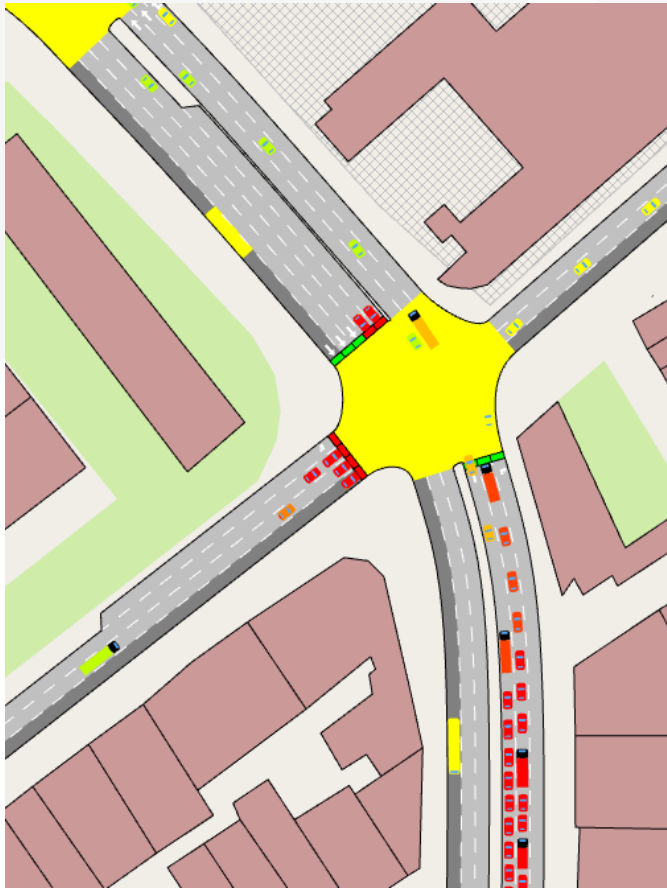
Symbol Editor

Colour Colour Ramp: Do Not Use the Ramp

1. View Styles Setup

View Modes

➤ Example 4: Vehicle Speed (Dynamic)



Scripting

Aimsun Programming

Scripting (Python)

- » Modifying the model
- » Importing/Exporting Data
- » Calculating new metrics from existing data

API (C++ or Python)

- » Communicate between simulators and external applications
- » Plugins for new ITS, ATDM, Ramp Meters, ATCS, etc.

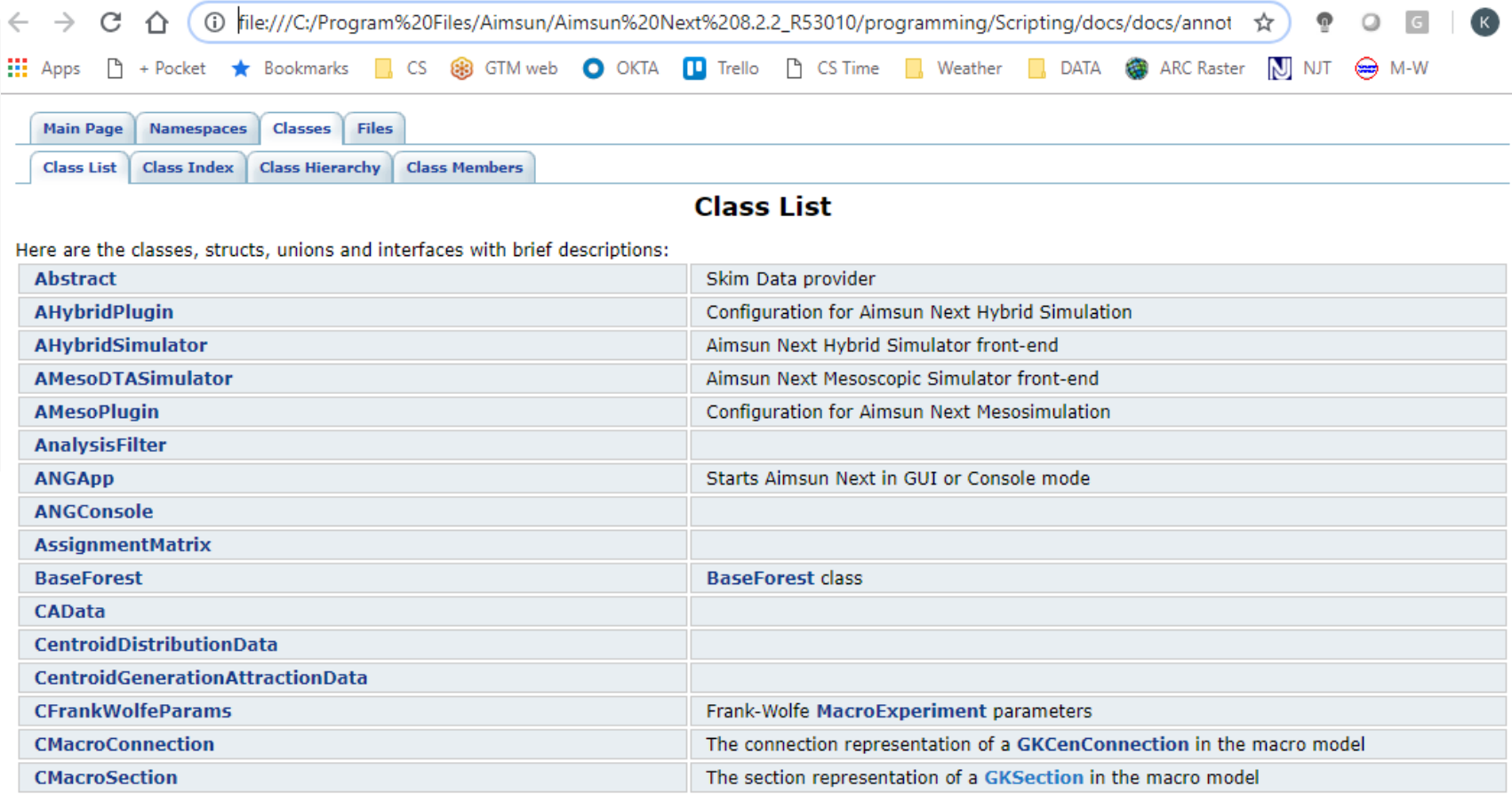
MicroSDK (C++)

- » Core Microsimulation Module Replacement

Aimsun Documentation

- Help Files (Aimsun Next Scripting)
- Scripting Manual (PDF)
- Scripting Documentation (HTML)
- Build in Scripts (as examples)

Scripting Documentation



file:///C:/Program%20Files/Aimsun/Aimsun%20Next%208.2.2_R53010/programming/Scripting/docs/docs/annot

Apps + Pocket Bookmarks CS GTM web OKTA Trello CS Time Weather DATA ARC Raster NJT M-W

Main Page Namespaces Classes Files

Class List Class Index Class Hierarchy Class Members

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Abstract	Skim Data provider
AHybridPlugin	Configuration for Aimsun Next Hybrid Simulation
AHybridSimulator	Aimsun Next Hybrid Simulator front-end
AMesoDTASimulator	Aimsun Next Mesoscopic Simulator front-end
AMesoPlugin	Configuration for Aimsun Next Mesosimulation
AnalysisFilter	
ANGApp	Starts Aimsun Next in GUI or Console mode
ANGConsole	
AssignmentMatrix	
BaseForest	BaseForest class
CAData	
CentroidDistributionData	
CentroidGenerationAttractionData	
CFrankWolfeParams	Frank-Wolfe MacroExperiment parameters
CMacroConnection	The connection representation of a GKConnection in the macro model
CMacroSection	The section representation of a GKSection in the macro model

Class Example (GKCentroid)

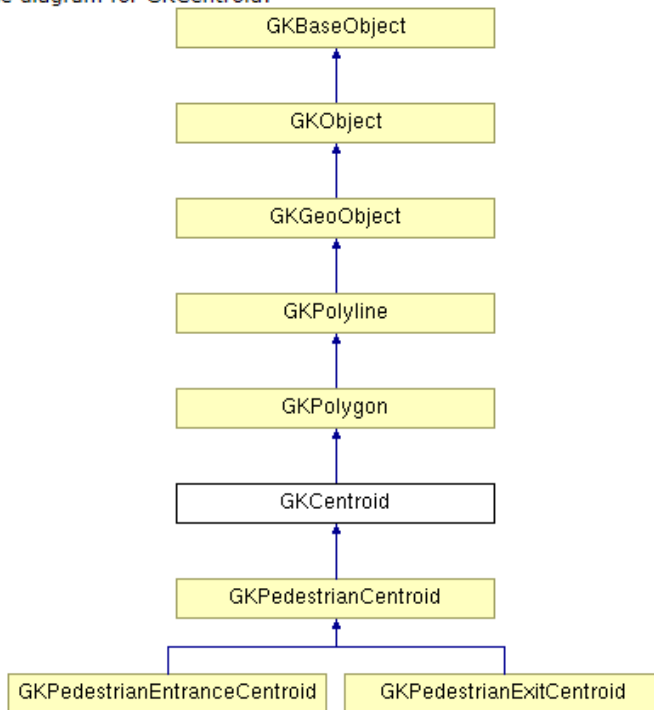
[Main Page](#)
[Namespaces](#)
[Classes](#)
[Files](#)

[Class List](#)
[Class Index](#)
[Class Hierarchy](#)
[Class Members](#)

GKCentroid Class Reference

Trip origin and/or destination object. [More...](#)

Inheritance diagram for GKCentroid:



Public Types

```

enum GKCentroidPercentages {
    eNothing, eOrigin, eDestination, eBoth,
    eSame
}
enum GKCentroidTraversalTypes { eNoTraversal, eInTraversal, eOutTraversal }
    
```

Public Member Functions

```

GKCentroid ()
virtual GKPoint getPosition () const
virtual void setFromPosition (const GKPoint &p)
void setConsideredPercentages (GKCentroidPercentages)
const GKCentroidPercentages getConsideredPercentages () const
void setUseBestEntrance (bool)
bool getUseBestEntrance () const
    
```

virtual void GKCentroid::setFromPosition (const GKPoint & p) [virtual]

Set the coordinates of the centroid from a single point.
Reimplemented in [GKPedestrianCentroid](#).

void GKCentroid::setManualPosition (GKPoint p)

Sets the position of the centroid independently of the polygon.

void GKCentroid::setPositionByConnections ()

Sets the position of this centroid using the position of the objects where it is attached.

Template Aimsun Scripts

▼ SCRIPTS

- 00 - Average Calculator
- 01 - Change Experiment
- 02 - Change Traffic Demand
- 03 - Change Turning Function
- 04 - Control Plan Editing
- 05 - Create Detector
- 06 - Delete Object
- 07 - Export Matrices
- 08 - Import Matrices
- 09 - Import Traffic State
- 10 - Look for Signalised Nodes
- 11 - Matrix To Excel
- 12 - Print Sections
- 13 - Public Transport Lines Per Section
- 14 - Public Transport Stops Per Section
- 15 - Reduce Sections Speed
- 16 - Setup Demand
- 17 - Show PT Lines
- 18 - Split Matrix
- 19 - Take Snapshot
- 20 - View Mode: Simulation Vehicles By Origin
- 21 - View Mode: Simulation Vehicles By Destination
- 22 - View Mode: Simulation Vehicles By Vehicle Types
- 23 - Calculate Nodes Flows
- 24 - View Mode: Nodes Flows
- 25 - Change Turning Stoplines Lengths
- 26 - Export OD Routes
- 27 - Import OD Routes
- 28 - C-Logit Factor Loading
- 29 - Area Statistics
- 30 - Copy Control Plan
- 31 - Delete OD Routes
- 32 - Mark Difference
- 33 - Lost SI Balls
- 34 - OD Balls
- 35 - Geocode
- 36 - Open Street Map
- 37 - Assignment
- 38 - Adjustment
- 39 - Traversal
- 40 - Import Control Plan
- 41 - Remove Overlapped Section Points
- 42 - Export Control Plan
- 43 - Export PT Lines and PT Stops
- 44 - Import PT Lines and PT Stops
- 45 - Export PT Lines Timetables
- 46 - Import PT Lines Timetables
- 47 - Print Polygon Area



Scripting Interface

Code (internal)

Set Context (if any)

Store Externally (if desired)

Python Script: 346, Name: 39 - Traversal {56e7685b-f108-4a30-8958-eb554c1631fd}

Main Settings

```
#####  
# (c) Aimsun #  
#####  
  
# Disclaimer of Warranty: TO THE EXTENT PERMITTED BY APPLICABLE LAW THE AIMSUN NEXT  
# SCRIPTS, INCLUDING BUT NOT LIMITED TO ALL DATA, TOOLS AND CALCULATIONS THEREIN  
# ARE PROVIDED "AS IS" AND WITHOUT EXPRESS OR IMPLIED WARRANTY OF ANY KIND BY  
# EITHER AIMSUN OR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION  
# OR DELIVERY OF THE AIMSUN NEXT SCRIPTS, INCLUDING BUT NOT LIMITED TO ANY EXPRESS  
# OR IMPLIED WARRANTY OF MERCHANTABILITY, ACCURACY, QUIET ENJOYMENT,  
# NONINFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE. NO COVENANTS, WARRANTIES  
# OR INDEMNITIES OF ANY KIND ARE GRANTED BY AIMSUN TO YOU THE USER. SHOULD THE  
# PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,  
# REPAIR OR CORRECTION.  
  
# This script runs a macro traversal calculation  
  
from PyMacroKernelPlugin import *  
from PyMacroToolPlugin import *  
  
if target != None:  
    subnetType = model.getType("GKProblemNet")  
    dlgEditor = GAnyObjectChooserEditor(GKGUISystem.getGUISystem().getActiveGui())  
    dlgEditor.setShowNoObjectsMessage(True)  
    dlg = dlgEditor.getChooser()  
    dlg.setType(model, subnetType, GAnyObjectChooser.eOneObject)  
    if dlgEditor.execDialog():
```

Name: 39 - Traversal External ID:

Find: Find Next Match Case Whole Word

Help Execute Save Line: 1 Column: 1 OK Cancel

Python Script: 346, Name: 39 - Traversal {56e7685b-f108-4a30-8958-eb554c1631fd}

Main Settings

Read Code from External File
File:

Add Script to a Menu
To Context Menu for Object Type: Static Assignment Experiment

Import
Imported Scripts: No imported scripts

Help Execute Save Line: 1 Column: 1 OK Cancel

Scripting Examples

- Modify model contents
- Check network element coding (QA/QC)
- Export model results
- Create new fields to help evaluate simulation outputs
- And others...

Modify Model Contents

- Automate manual processes for efficiency
- Add, modify, or delete elements and data fields
- Examples:
 - » Import or update section data (Import CSV)
 - » Import matrices
 - » Field updates using formulas (Update Attractiveness)
 - » Add detectors to network
 - » Name turns

Check network element coding (QA/QC)

- Automate review process across networks
- Examples:
 - » Find missing turns
 - » Report contents of a master control plan
 - » Review signal coding
 - Do all actuated phases have detectors attached?
 - Are all signal groups included in at least one phase?

Export Model Results

- Export simulation results
 - » Section MOEs
 - » Turn MOEs
 - » Detector MOEs
 - » Network-wide MOEs
- Allows external processing and summary of data (e.g. Excel)

Evaluate Outputs

➔ Calculate new MOEs

➔ E.g. Find Blockages in a meso simulation:

```
for section in model.getCatalog().getObjectsByType(sectionType).itervalues():
    ### Everything returns in metric units
    kJam = section.getDataValueDoubleByID(section.jamDensityAtt)
    timeSeries = section.getDataValueTS(blockageColumn)
    if timeSeries != None:
        model.resetColumn(blockageColumn)

for i in range(newTsDescription.getInterval(fromTime), newTsDescription.getInterval(toTime)+1):
    blockedLanes = 0
    nbLane=section.getLastSegmentNbLanes()
    for laneNb in range(1, nbLane+1):
        flowAtt = model.getColumn("DYNAMIC::GKSection_flow_%s_0_%s" % (replication.getId(), laneNb))
        densityAtt = model.getColumn("DYNAMIC::GKSection_density_%s_0_%s" % (replication.getId(), laneNb))
        tsIndex=GKTimeSerieIndex(i)
        flow = section.getDataValueInTS(flowAtt, tsIndex)[0]
        density = section.getDataValueInTS(densityAtt, tsIndex)[0]
        if density > (kJam * 0.9) or (density > kJam*.2 and flow==0):
            blockedLanes += 1
    tsIndex = GKTimeSerieIndex(i,newTsDescription.dimensions())
    section.setDataValueInTS(blockageColumn, tsIndex, float(blockedLanes)/float(nbLane), 0.0, newTsDescription)
```

And others...

- Powerful manipulation of Aimsun model elements
 - » Network elements (sections, turns, centroids...)
 - » Control devices (signals and turns)
 - » Matrices, Traffic Managements, etc.
 - » Batch simulation runs
- Does require programming skills in Python
- Should be done carefully
 - » Backup your model first !