# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td>1</td>
</tr>
<tr>
<td>List of Figures</td>
<td>5</td>
</tr>
<tr>
<td>List of Tables</td>
<td>6</td>
</tr>
<tr>
<td>History and Overview</td>
<td>7</td>
</tr>
<tr>
<td>NADS miniSim™ Introduction</td>
<td>9</td>
</tr>
<tr>
<td>miniSim™ Community</td>
<td>10</td>
</tr>
<tr>
<td>Why Choose miniSim™?</td>
<td>10</td>
</tr>
<tr>
<td>Pricing</td>
<td>11</td>
</tr>
<tr>
<td>Typical costs for complete miniSims and services are as follows:</td>
<td>11</td>
</tr>
<tr>
<td>Support and Maintenance</td>
<td>12</td>
</tr>
<tr>
<td>Shipping, Installation and Training</td>
<td>13</td>
</tr>
<tr>
<td>Other Resources</td>
<td>14</td>
</tr>
<tr>
<td>miniSim™ Technical Description</td>
<td>16</td>
</tr>
<tr>
<td>Overview</td>
<td>16</td>
</tr>
<tr>
<td>miniSim™ ¼ Cab Driving Simulator</td>
<td>17</td>
</tr>
<tr>
<td>miniSim ¼ Cab Dimensions</td>
<td>18</td>
</tr>
<tr>
<td>miniSim ¼ Cab Hardware Description</td>
<td>19</td>
</tr>
<tr>
<td>miniSim™ Simplified Cab Simulator</td>
<td>20</td>
</tr>
<tr>
<td>miniSim™ Desktop Driving Simulator</td>
<td>23</td>
</tr>
<tr>
<td>Software Description</td>
<td>24</td>
</tr>
<tr>
<td>Real-Time Core</td>
<td>24</td>
</tr>
<tr>
<td>Simulation Control Front-End GUI</td>
<td>24</td>
</tr>
<tr>
<td>Driving Control Feel and Instrumentation</td>
<td>24</td>
</tr>
<tr>
<td>Vehicle Dynamics</td>
<td>24</td>
</tr>
<tr>
<td>Scenario Control</td>
<td>24</td>
</tr>
<tr>
<td>Visual Rendering</td>
<td>25</td>
</tr>
<tr>
<td>Audio Engine</td>
<td>25</td>
</tr>
</tbody>
</table>
Data Acquisition ................................................................................................................. 25
miniSim Operator Interface ................................................................................................. 27
Dependent Measures ........................................................................................................... 29
  Values and locations of Maximum and/or Minimum ......................................................... 29
  Average and deviation ........................................................................................................ 29
  Reaction time ..................................................................................................................... 29
  Steering Responses ........................................................................................................... 29
  Lane change detection ...................................................................................................... 30
  Eye-tracking (Optional) ..................................................................................................... 30
  Crash detection ................................................................................................................ 30
  Time to collision ............................................................................................................... 30
Data Stream Contents ......................................................................................................... 30
  Driver Input Variables ...................................................................................................... 32
  Collision Detection Variables ......................................................................................... 33
  Scenario Variables .......................................................................................................... 33
  Vehicle Dynamics Variables ........................................................................................... 36
ISAT™ (Interactive Scenario Authoring Tool) ..................................................................... 38
TMT™ (Tile Mosaic Tool) .................................................................................................... 39
Relationship Between Software Modules ............................................................................ 40
Virtual Environments .......................................................................................................... 41
  NADS miniSim™ Included Tiles and Databases (Right-Hand-Traffic Only) ................. 42
  Scenario Object Library (SOL) ........................................................................................ 47
    Vehicles ......................................................................................................................... 47
    Control Signs ................................................................................................................ 50
    Warning Signs .............................................................................................................. 50
    Construction Signs ..................................................................................................... 50
    Route Direction Signs ................................................................................................. 51
    Speed Limit Signs ........................................................................................................ 51
    Interstate and Highway Signs ...................................................................................... 52
    Objects ......................................................................................................................... 53
City Signs.......................................................................................................................... 57
Street Signs ....................................................................................................................... 58
Guide Signs ...................................................................................................................... 59
Rail Functionality ............................................................................................................ 60
Rail Vehicles .................................................................................................................. 60
Railroad Tiles .................................................................................................................. 63

miniSim Options ............................................................................................................. 66
Custom Virtual Environment Development ................................................................. 66
Mobile Trailer Installation Option ................................................................................ 66
4-Channel Video Capture System Option ...................................................................... 72
Eye Tracking Integration Option ..................................................................................... 74
NADS Springfield 2014 Road Network Option ............................................................. 76
Logical Database API ...................................................................................................... 78
  Description of API Functionality: .............................................................................. 79
  Road Information ........................................................................................................ 79
  Static Object Functions ............................................................................................... 81
  Dynamic Object Functions ......................................................................................... 81
  Example program ....................................................................................................... 81
Vehicle Automation Option ............................................................................................ 81
  Lane Following ............................................................................................................ 82
  Speed Following ......................................................................................................... 82
  Lane Change ............................................................................................................... 82
  Brake to Stop .............................................................................................................. 82
  Merge / Exit ............................................................................................................... 82
  Turning ......................................................................................................................... 82
  Pull Over ..................................................................................................................... 83
  Abort ............................................................................................................................ 83
Transfer of Control ......................................................................................................... 83
Driving Style .................................................................................................................. 83
Triggers ............................................................................................................................ 83
User-Defined Simulator Subsystem ................................................................................ 84
LIST OF FIGURES

Figure 1 - NADS-1, the world's most advanced driving simulator................................. 7
Figure 2 - NADS current and past partnerships ............................................................. 8
Figure 3: The miniSim™ Driving Simulator with Partial Cab, and Operator Display ........... 9
Figure 4: High-Resolution Rendering of Typical Scene .................................................. 9
Figure 5 – miniSim User’s Website (www.nads-sc.uiowa.edu\minisim) ...................... 12
Figure 6 - Multi-use crate for shipping ........................................................................... 13
Figure 7 - The miniSim(TM) driving simulator with 3 plasma displays ......................... 16
Figure 8: The miniSim™ Driving Simulator with Partial Cab, and Operator Display ...... 17
Figure 9: miniSim Quarter-Cab Dimensions ................................................................. 18
Figure 10: Simplified Cab shown with three 24-inch LCDs ........................................... 20
Figure 11: Simplified Cab shown with triple floor standing monitor stand, 46”-inch LCDs, standing desk ................................................................................................. 21
Figure 12: Simplified Cab shown with single monitor stand, 46” LCD, standing desk ... 21
Figure 13: Enhanced Logitech G27 wheel and button boxes with labeled buttons ...... 22
Figure 14: Desktop miniSim™ Driving Simulator with single 42” Display (L), and optional three 20” LCD Displays (R). ............................................................. 23
Figure 15 - miniSim(TM) computing architecture with optional motion subsystem and custom subsystem ........................................................................................................... 26
Figure 16: miniSim User Interface .................................................................................. 28
Figure 17: miniSim Data Collection Window ................................................................. 28
Figure 18 - daqViewer Screenshots (MATLAB GUI) .................................................... 31
Figure 19 - NADS ISAT(TM) scenario authoring tool .................................................... 38
Figure 20 - Geo-typical Tile database assembly within the Tile Mosaic Tool ............... 39
Figure 21 - A high resolution model and driving environment from the miniSim™ driving simulator ................................................................................................................. 41
Figure 22 - The West Liberty raceway (left is the actual location, right is the virtual environment) ................................................................................................................. 41
Figure 23 - Examples of different texture types, feature density, overall appearance between Levels 1 and 3. ............................................................. 67
Figure 24 - An example of a Level 1 virtual environment .............................................. 68
Figure 25 - An example of a Level 2 virtual environment .............................................. 68
Figure 26 - Examples of Level 3 virtual environments .................................................. 69
Figure 27 - The Verasity driving simulator based on miniSim(TM) components (photo courtesy of Tom Zwica). ......................................................................................................................... 72
Figure 28 - Interior view of miniSim(TM) driving simulator installed in the back of a mobile trailer ........................................................................................................................................... 72
Figure 29 – Mobile Driving Simulator .................................................................................. 73
Figure 30: Vidcap System Output with Quad Split .......... Error! Bookmark not defined.
Figure 31: Vidcap System Configuration .............. Error! Bookmark not defined.
Figure 32: Vidcap User Interface .................. Error! Bookmark not defined.
Figure 33: Video Camera Capture Configuration WindowError! Bookmark not defined.
Figure 34: FaceLAB Configuration for miniSim Data Logging. ....... Error! Bookmark not defined.
Figure 35 MiniSim Data Flow ............................................................................................ 78
Figure 36 Data flow for API ............................................................................................. 79

LIST OF TABLES

Table 1 - Characteristic of graphical virtual environments at Levels 1, 2 and 3 .......... 67
History and Overview

Located on the University of Iowa’s Research Park, the National Advanced Driving Simulator (NADS) is committed to the goal of pursuing the state-of-the-art in driving simulation research and development. The NADS facility is home to the world’s most advanced driving simulator, the NADS-1. The NADS performs driving safety research for federal and state agencies and executes R&D contracts for private industry partners.

Figure 1 - NADS-1, the world’s most advanced driving simulator

NADS is also at the forefront of conducting vehicle and safety research for the U.S. Department of Transportation and NHTSA (National Highway Traffic Safety Administration). Since 2004, NHTSA has funded over $5 million dollars worth of research at NADS. With its partnerships with NHTSA and other agencies, NADS has developed some of the most realistic vehicle driving simulations available anywhere in the world.
Figure 2 - NADS and miniSim current and past partnerships
NADS miniSim™ Introduction

This document describes the NADS miniSim™ Driving Simulator. The NADS miniSim is a PC-based driving simulator with powerful scenario editing and data acquisition capabilities that is based on over 15 years of research and development at the University of Iowa’s National Advanced Driving Simulator (NADS). The hardware is available in three configurations: Quarter-Cab, Simplified Cab, and Desktop. The software is the same for all versions and can support custom cab designs, external subsystems, and vehicle automation applications. NADS technology is used by NADS staff on a daily basis to fulfill scientific research contracts on our NADS-1, NADS-2, and miniSim™ simulators, ensuring that you get the most up-to-date capabilities. The miniSim can be configured a variety of ways, including single and multi-screen displays and desktop, simplified cab, ¼-cab and ½-cab configurations.

Figure 3: The miniSim™ Driving Simulator with Partial Cab, and Operator Display.

Figure 4: High-Resolution Rendering of Typical Scene
miniSim™ Community

We have a growing community of researchers and trainers using NADS simulator technology in their programs as shown in the map below. More information is available on the miniSim website: www.nads-sc.uiowa.edu/minisim.

Why Choose miniSim™?

- miniSim provides the best value for many levels of fidelity on a not-for-profit basis
- Unmatched support, including remote simulator access with TeamViewer™
- Our development team can create custom scenarios, data reduction, road networks, and hardware solutions to meet your needs exactly.
- Access to a national network of researchers who are using the same core simulation technology, facilitating standardization, collaboration and sharing of scenarios, data, and ideas.
- NADS conducts ground-breaking driving simulation research and is home to the most advanced driving simulators in the world.
- NADS simulation platforms represent over twenty years of simulator development experience, longer than any other driving research organization.
- Our strategic plan allows us to maintain state of the art simulators with continued updates to provide greater fidelity than any other organization.
Pricing

Ask NADS for a miniSim proposal, configured for your needs. Typical costs for complete miniSims and services are as follows:

**One Time Hardware Costs:**
- Desktop miniSim Hardware: $14,000 approx. (PC, monitors, wheel, tactile, etc.)
- Simplified-Cab miniSim Hardware: $25,500 approx. (Cab, PC, 48” monitors+stand, etc.)
- Quarter-Cab miniSim Hardware: $55,500 approx. (Cab, monitors+stand, etc.)
- Half- and Full-Cab Simulators: $175-450k (depending on cab, display, motion options)

**Installation and Training:**
- Training at NADS or remotely (4 days): $3,800 (only for Desktop and Simplified Cab systems)
- Installation, and Training On-Site: $12,000+ (for Quarter Cab and larger)

**Yearly License Fee:**
- Yearly Software Support/Maint: $6,250

**Options:**
- Springfield Road Network: $5,250
- 4 Channel HD Video Capture System: $10,300
- Eye-Tracking: $TBD (Ergoneers, Tobii, FOVIO, Smart Eye)
- Custom Scenarios, Models, Development, etc: $TBD

For contracting with us, there are three elements:
- The hardware, installation, training, and any project-specific development or customization is under a UI research contract.
- The ISAT, miniSim, Tiles, and nDaqTools are delivered by NADS for:
  - *Internal R&D and Non-profit applications*: a software transfer agreement through the UI Research Foundation is used. A $6k/year software support and maintenance fee is required.
  - *Commercial applications*: a commercial license can be obtained from the UI Research Foundation. This typically requires a license fee based on a percentage of revenue model. This typically allows the holder to derive revenue directly from simulator based services or distribution.
- The Tile Mosaic Tool is delivered by, and fully supported by NADS, but a $0 loan agreement from Rockwell Collins is required.
Support and Maintenance

The miniSim software support contract is $6,250 for every year the customer possesses the software. This entitles the customer to use of the software, telephone, email and remote assistance in using the miniSim software, and to periodic software updates on a schedule defined by NADS. This also entitles the user to enhanced access to the miniSim user's website. The customer will be provided with a phone number and e-mail for contacting technical support. Personnel will be available for support during normal business hours (8am-5pm CST) Monday thru Friday (except University of Iowa holidays), and will respond to requests within 24 hours.

Hardware support and repairs will be defined by NADS and provided at our cost.

Custom software and hardware enhancements will be defined by NADS, and will be available for purchase and installation on terms mutually agreed upon.
Shipping, Installation and Training

As a part of your simulator project, NADS provides shipping installation and training services.

The simulator cab is shipped in a sturdy crate to protect it from damage. Accessories and displays are also packed carefully in wood crates.

![Image of a multi-use crate for shipping]

Figure 6 - Multi-use crate for shipping

Installation is performed by a NADS engineer at the customer’s site. Training is provided for setting up, operation, and use of the simulator and the accompanying software tools.

NADS will send 1 staff member to deliver the system and provide training on its operation. The installation and training will take 4 days and will be hands-on. The training will cover the following:

- Hardware connections
- Configuration of simulator initialization and calibration files
- Selecting different scenarios and ownship vehicles
- Starting and stopping the simulator
- Troubleshooting
- Maintenance procedures
- Procedures for getting help
- How to assemble database tiles into a road network with the NADS TMT™
- How to create scenarios using the ISAT™
- Importing new scenarios and running them on the miniSim
- Viewing the data file collected on the miniSim in the ISAT tool
- Using the supplied daqConvert utility to convert the data files into Matlab format
- NADS nDaqTools for Matlab to view data, and create automated data reduction scripts
Other Resources

There are videos of the miniSim available on our website:

http://www.nads-sc.uiowa.edu/media.php?pageno=1&showAll=video

NADS publications:

http://www.nads-sc.uiowa.edu/brochures.php
http://www.nads-sc.uiowa.edu/publications.php

NHTSA Publications:

http://www.nhtsa.gov/Research/Driver+Simulation+(NADS)/COMPLETED+NADS+RESEARCH+REPORTS
http://www.nhtsa.gov/Research/Vehicle+Dynamics+Simulation

If you are new to driving simulation NADS has prepared materials for an introduction to driving simulation workshop:

http://www.nads-sc.uiowa.edu/workshop/

Lastly, there is the Handbook of Driving Simulation for Engineering, Medicine, and Psychology that was written by researchers in the community about the technology, best practices, etc.:

http://www.nads-sc.uiowa.edu/workshop/wkshp_resources.php

VT MAK’s Di-Guy Advanced Digital Pedestrians available for miniSim:
http://www.mak.com/products/humans.html

We have negotiated special pricing with VT MAK for miniSim users: $1,250 for single screen and $2,500 for 3-screen: http://www.mak.com/component/content/article/20-news/686-2014-12-01-13-15-58.html

Standardization of measures used in the driving simulation community is an ongoing process. A new SAE Recommended Practice is available to address this issue:

SAE J2944 Operational Definitions of Driving Performance Measures and Statistics

A draft version of the Practice is available here:


These references will be helpful to those developing data reduction scripts.

User-Developed Python Data Reduction Tools ('undaqTools' by Roger Lew at University of Idaho):
https://github.com/rogerlew/undaqTools
User Videos and Links:

http://www.youtube.com/watch?v=OgsrFAe_Lgc
http://sonify.psych.gatech.edu/research/driving/index.html
https://www.youtube.com/watch?v=8uIO10Hk09E
http://www.newsweek.com/2015/04/10/ensuring-your-childs-drivers-license-isnt-license-kill-317487.html#.VSCBmA2FvJg.email
http://www.toyotaownersmagazine.com/2014/10/01/fair-warning/

A user-publication list is available upon request.
miniSim™ Technical Description

Overview
The NADS miniSim™ driving simulator, Figure 7, is a highly flexible PC-based driving simulation system. It uses the same advanced technologies built into the more sophisticated large NADS simulators, at a fraction of the cost to build and operate, and is easy to customize for specific applications. The system has a very small footprint and is highly portable. The configuration data for the virtual environment are in the same format as those used by the large NADS simulators; therefore, studies designed for the miniSim™ can be easily carried out on the latter when a simulation of higher fidelity is necessary.

Figure 7 - The miniSim(TM) driving simulator with 3 plasma displays
miniSim™ ¼ Cab Driving Simulator

Figure 8: The miniSim™ Driving Simulator with Partial Cab, and Operator Display.

We propose a partial-cab miniSim driving simulator combined with plasma displays, Figure 8. The simulator will be provided with the following:

- miniSim cab with the following features:
  - LCD display that will act as the virtual instrument cluster
  - Real-vehicle seat, steering wheel with column gear selector, and pedals
  - ‘Active Steering Loader’ with DC motor/microprocessor control
  - 2.1 channel sound system with vibration transducer under seat
  - Audio Amplifier with External Controls

- Single rack-mounted PC:
  - Windows 10 Pro
  - Intel Core i7 3.4GHz 6-Core Processor
  - 2 SSD RAID (240 Gb ea, 240 Gb usable)
  - 32 Gb DDR3 SDRAM
  - NVIDIA GeForce GTX 1080 and GTX 780 GPUs

- Three 1080p LED LCD widescreen displays of 46” diagonal measurement.

- A 22” operator LCD for miniSim, ISAT and TMT interfaces

- Software as per the separate UIRF Software Transfer Agreement:
  - NADS miniSim™ real-time driving simulation software
  - NADS ISAT™ scenario authoring tool for creating/modifying scenarios
  - NADS TMT™ virtual environment authoring tool with 92 sample tiles
  - NADS nDaqTools and GUI for Matlab to assist in Data Reduction
  - NADS VidCap Video Capture software

- Tile Mosaic Tool software per the separate Rockwell Collins agreement
miniSim ¼ Cab Dimensions

**Monitor Stand Details**

- **Field of View:**
  - 138° horizontal, 27° vertical at 48” viewing distance
  - Angles and viewing distance adjustable
- **Display Type:**
  - NEC 46” Professional LED backlit LCD (1920x1080)
  - Display Dimensions: (W x H x D) 41.2” x 24.5” x 2.5”
  - Display Weight: 30 lbs
- **Dimensions with Displays:**
  - Width 3-Displays: 120”
  - Width 1-Display: 41.2”
  - Height: 58”, 78” with glare canopy
  - Weight: 200 lbs (3 displays), 75 lbs (1 display)
- **Electrical:**
  - 120 VAC, 300W (3 display), 100 W (1 display)

**Cab Details**

- **Characteristics:**
  - LCD display instrument cluster, 12.1” widescreen (16:9)
  - Single Integral PC, removable HDD RAID 5
  - Column Shift with Correct Spring Detents
  - Microprocessor Controlled Steering Loader, 3 turns
  - Turn signals and two steering wheel buttons
  - Passive Brake loader with realistic force-displacement curve
  - Drive by wire accelerator pedal
  - Black vinyl seat, adjustable
  - 2.1 surround sound (2 front, 1 subwoofer under seat)
  - Tactile Transducer (road vibration)
- **Dimensions:**
  - Length: 67” (seatback folded), 75” (seatback upright)
  - Width: 27”
  - Height: 41”
  - Weight: 300 lb.
- **Electrical:**
  - Power Requirement: 120 VAC, 500 W

*Figure 9: miniSim Quarter-Cab Dimensions*
miniSim ¼ Cab Hardware Description

Displays
The ¼-cab simulator uses three 46” LED backlit LCD displays. Other display technology options are available including the using of smaller LCD screens, and front or rear projection systems.

- Screen size 46” (each display)
- Resolution 1920x1080
- Refresh rate 60Hz
- Horizontal FOV 138 to 180 degrees
- Vertical FOV 27 to 50 degrees

Operator Controls
The simulator features an actual vehicle steering wheel and pedals controls. The steering wheel uses a semi-active controller that is driven by a DC motor and microcontroller. The steering column features a functional turn signal indicator. The brake and accelerator pedals feature a realistic feel. Force feedback on the brake is available as an option. Other switches can be instrumented per customer requirements.

Hardware controls:
- Steering
- Accelerator pedal
- Brake pedal
- Shifter
- Turn signal
- Seat
- Wipers

Functioning gauges and lights on virtual instrument cluster:
- Speedometer
- Tachometer
- Gear selector
- Turn signals
- Odometer

Computers
The simulator uses a single computer to control all aspects of the real-time simulator system. The PC uses off-the-shelf parts for easy maintenance and upgrades.

1 Variable with how close the displays are placed to the driver’s eye-point
2 Variable with how close the displays are placed to the driver’s eye-point
miniSim™ Simplified Cab Simulator

Figure 10: Simplified Cab shown with three 24-inch LCDs

- NADS miniSim computer loaded and tested prior to shipment
- LCD Virtual Instrument Cluster
- Automotive seat
- Tilt wheel adjustment
- Logitech G27 steering loader enhanced with a 350 mm leather-wrapped wheel
- 6 Wheel buttons that can be used for driver response, look left/right, shoulder check left/right
- Labeled buttons for Engine Start, Gear Select, Wipers, Hazard, Mirrors, Headlights, and two Auxiliary Buttons
- Utilizes the high quality top-pivot pedals from ECCI
- 2.1 Audio and tactile transducer and amplifier provides road and engine vibration
- A 22” operator LCD for miniSim, ISAT and TMT interfaces
- Software as per the separate UIRF Software Transfer Agreement:
  - NADS miniSim™ real-time driving simulation software
  - NADS ISAT™ scenario authoring tool for creating/modifying scenarios
  - NADS TMT™ virtual environment authoring tool with 92 sample tiles
  - NADS nDaqTools and GUI for Matlab to assist in Data Reduction
  - NADS VidCap Video Capture software
- Tile Mosaic Tool software per the separate Rockwell Collins agreement
The following options are available for the Simplified Cab miniSim:

- Single display (46 inch to 65 inch or more)
- Three front 24 inch monitors
- Three front 46 inch monitors
- ECCI Trackstar 6000 wheel (350mm wheel, 250-deg rotation only, passive spring-damper)
- Standing Desk

![Image of Simplified Cab with triple floor standing monitor stand, 46"-inch LCDs, standing desk](image1)

**Figure 11:** Simplified Cab shown with triple floor standing monitor stand, 46”-inch LCDs, standing desk

![Image of Simplified Cab with single monitor stand, 46” LCD, standing desk](image2)

**Figure 12:** Simplified Cab shown with single monitor stand, 46” LCD, standing desk
Figure 13: Enhanced Logitech G27 wheel and button boxes with labeled buttons
miniSim™ Desktop Driving Simulator

Figure 14: Desktop miniSim™ Driving Simulator with single 42” Display (L), and optional three 20” LCD Displays (R).

A Desktop miniSim driving simulator is shown in Figure 14. The simulator is provided with the following:

- NADS miniSim computer loaded and tested prior to shipment
- Three 20” front channel displays, and 16” dashboard display.
- Logitech G27 USB steering wheel and pedal set*
- A 22” operator display with USB Extension for keyboard and mouse
- 2.1 Audio and Tactile transducer and amplifier provides road/engine vibration
- Software as per the separate UIRF Software Transfer Agreement:
  - NADS miniSim™ real-time driving simulation software
  - NADS ISAT™ scenario authoring tool for creating/modifying scenarios
  - NADS TMT™ virtual environment authoring tool with 92 sample tiles
  - NADS nDaqTools and GUI for Matlab to assist in Data Reduction
  - NADS VidCap Video Capture software
- Tile Mosaic Tool software per the separate Rockwell Collins agreement

Notes: Desk not included

The MiniSim is also compatible with the following driver controls:

- ECCI: http://ecfi6000.com/6000_productportal.htm

The following options are available for the desktop simulator:

- Single display (46 inch to 65 inch or more)
- Three front 24 inch monitors
- Three front 46 inch monitors
- ECCI Trackstar 6000 wheel (350mm wheel, 250-deg rotation only, passive spring-damper)
- Standing Desk
- Logitech G27 steering loader enhanced with a 350 mm leather-wrapped wheel and 6 wheel buttons that can be used for driver response, look left/right, shoulder check left/right
- Button Boxes for Engine Start, Gear Select, Wipers, Hazard, Mirrors, Headlights, two Auxiliary
Software Description
The miniSim Driving Simulator contains the following core software components:

- Real-Time Core
- Simulation Control Front-End GUI
- Driving Control Feel and Instrumentation
- Vehicle Dynamics
- Scenario Control
- Visual Rendering
- Audio Engine
- Data Acquisition

Real-Time Core
The real-time core is the underlying engine that coordinates the execution of the various functionalities of the simulator and provides the real-time data communication mechanism among components, either through shared memory on the same computer, or through a local TCP/IP connection between different computers. The other components are built on top of the real-time core.

Simulation Control Front-End GUI
The front-end GUI provides control to the driving simulation process and to data collection and analysis. The operator can start and stop drives, select different scenarios, play presentation slides, and collect and analyze driving data through the front-end GUI. The GUI will be displayed on a small touchpad screen.

Driving Control Feel and Instrumentation
The driving control feel and instrumentation component reads the driver’s control input via the steering wheel and foot pedals and passes it to the vehicle dynamics component. It also takes the vehicle data such as speed and engine RPM and renders them to the instruments, which are simulated on the main display.

Vehicle Dynamics
The vehicle dynamics component provides a physics-based vehicle model that simulates the vehicle the subjects drive during the runs. It is identical to the corresponding component in the large NADS simulators. By changing input data, the same software can be used to simulate different types of vehicles.

Scenario Control
The scenario control component is responsible for simulating the behavior of other vehicles and objects in the virtual environment. It is identical to the scenario control component in the large NADS simulators and takes the same scenario configuration files generated by ISAT as input.
Visual Rendering
The visual rendering component renders the visual scene to the main display(s) of the driving simulator. One PC with a high-performance graphics card will be used for the rendering of up to three main channels. The renderer is built on top of an OpenGL-based open-source graphics library.

A range of environmental conditions can be rendered in the environment using the scenario control system:
- Rain
- Snow
- Fog
- Wind (speed and direction)

The driver can control the windshield wipers. Precipitation accumulation, the effect of wind on the precipitation, and the wiping effect is simulated.

Audio Engine
The audio engine provides audio cues to the driver. The component produces sounds from the own vehicle, including engine, powertrain, and tires, as well as sounds from the surrounding environment. The PC that hosts the audio engine is equipped with a PC audio card and connected to a set of speakers.

Data Acquisition
The driver’s data are collected at the same frequency that the miniSim runs. The set of data that can be collected includes driving control inputs, own vehicle states, and scenario data. All data are stored on file for replay and future analysis.
Figure 15 - miniSim(TM) computing architecture with optional motion subsystem and custom subsystem
miniSim Operator Interface
The miniSim software interface is the primary method a user interacts with the simulator, Figure 16. The user interface has the following functionality:

Scenario Selection and Start/Stop
The user selects the scenario to run from a drop-down list and can start and stop the simulation.

Scenario Import and Delete
The user is able to import new scenarios and delete existing ones through a standard windows dialog.

miniSim™ Data Collection Mode
The miniSim data acquisition system (DAQ) runs every time a scenario is run. However, when the miniSim is placed in Data Collection Mode, the operator must select the Experiment Name and Test Subject ID from a list you create. The Experiment and Test Subject ID selected then restricts the allowable scenarios that can be run according to a matrix defined by the user – ensuring that the data collection conforms to your experimental design. In addition, you may enter notes or comments about a particular test subject or drive and these comments are saved for future reference, Figure 17. You may also manually enter The Experiment and Test Subject ID at runtime.

Replay Mode
The replay mode provides a method to document consistency of data across multiple simulators, an important consideration for multi-site data collections, for example. A driver input file is recorded, and the simulation re-run on multiple simulators and the data compared. The driver input file is specific to the road network and scenario being driven. The Runtime Measures Evaluation, described below, offers a quick and easy method to compare and document performance across simulators.

Runtime Measures Evaluation
While the miniSim data collection system (DAQ) collects well over 100 variables for post-processing, the Runtime Measures Evaluation functionality offers the researcher the option of obtaining measures from the simulator immediately when the drive ends. The measures are reported for the entire drive, and for up to 20 events. The user has total control in defining the start and end points of each event through the scenario.

The current list of runtime measures are as follows:
- Collision Count
- Maximum Speed
- Minimum Speed
- Average Speed
- Standard Deviation of Speed
- Lane Departure Count
- Lane Departure Percentage
- Speeding Count
- Speeding Percentage
- Average Headway
- Standard Deviation of Lane Position

Other measures are possible – just ask us!

Driver Input Calibration
The miniSim has two easy-to-use features that allow calibration of the driver inputs. This is important, as many driver input mechanisms use analog sensors (potentiometers) for steering wheel and pedal position, and differences between setups and controls are possible and must be accounted for.

One feature is an automatic calibration routine that asks the operator to turn the wheel in each direction to the end stop and press on each of the pedals. The miniSim captures this data, and produces a calibration file. The second feature is a verification of the driver inputs – the operator can operate the controls and verify that the calibration values are being applied correctly.
Figure 16: miniSim User Interface

Figure 17: miniSim Data Collection Window
Dependent Measures

Values and locations of Maximum and/or Minimum

Typical values are maximum brake pedal force, maximum steering wheel angle, maximum/minimum longitudinal/lateral acceleration and maximum/minimum own vehicle speed. These values are usually used to study the severity of the driver’s action or movement, particularly useful in examining stopping or slowing in unsafe circumstances or unsafe intersection behavior typical of impaired drivers.

- Time to Maximum Brake Press: Time from the start of the event until the driver has maximum brake displacement
- Maximum Deceleration
- Maximum Steering Rate
- Maximum Lateral Acceleration
- Maximum Lateral Jerk

Average and deviation

Values can be used to study the driver’s behavior while distracted or impaired. For example, the lane deviation of the driver’s vehicle has been used to study the influence of alcohol on the driver. These values better capture the variation of driver’s performance versus more extreme responses captured in minimum/maximum values.

- Average own vehicle speed
- Lane deviation of driver vehicle
- Average brake pedal force
- Average steering wheel angle.

Reaction time

Values include accelerator pedal release time, brake pedal press time, sign-recognition reaction time, etc. These values are used to study the driver’s reaction time during an event. For example, if the driver is asked to press a button upon seeing a certain sign, the time between the sign appearing and the button press is the sign-recognition reaction time. This measure indicates how fast and well a driver is monitoring the environment, particularly relevant for landmark and traffic sign identification tasks, other tasks designed to assess visual search and target recognition, or route following tasks using verbal directions to a destination to test verbal memory, visual perception and attention.

- Accelerator Release Reaction Time: Time from the start of the event until the driver has released the accelerator pedal
- Brake Press Reaction Time: Time from the start of the event until the driver initially presses the brake pedal
- Button-push or Verbal Response Reaction Time

Steering Responses

Erratic steering, leading to shoulder incursions or lane excursions, are often indicators of increased driver workload.

- Time to Steering Onset: Time from start of the event until the driver has a corrective steering input exceeding 5 degrees
- Length of Excursion
- Extent of Excursion
- Area of Excursion
- Duration of Excursion
Lane change detection
This measure includes the detection of a lane change, how long the lane change takes, lane change angle, etc. Lane changes are very common events in real life.

- Minimum Time-To-Line Crossing: The minimum time-to-collision from event onset until end of the event.
- Time-To-Line Crossing At Accelerator Release
- Time-To-Line Crossing At Brake Press
- Time-To-Line Crossing At Steering Onset

Eye-tracking (Optional)
Eye-tracking is used to study the driver’s eye movements during the events. Measures include time spent looking forward, into mirrors or over the shoulder. These measures can be used to examine how the driver observes the environment generally, as well as for specific tasks measuring attention, divided attention, and selective attention while driving.

- Eyes on Road Time
- Eyes on Target Time
- Eyes off road at event onset: An indication of distraction

Crash detection
Crash detection includes detecting the crash, the speed of the driver vehicle at the time of the crash, and the speed of the other vehicle at the time of the crash (if applicable). Conflict risk. Conflict risk is also used to study the degree of danger between the own vehicle and a lead or adjacent vehicle. The degree of danger depends on the speed of the two vehicles and the distance between them.

- Collision: Indicates whether the driver’s vehicle collided with the braking lead vehicle
- Relative Velocity at Collision: Indicates the difference in speed at the time the vehicles collided.
- Near Misses: Indicates that the driver was near colliding with the lead vehicle.
- Conflict: Indicates that the driver was in conflict with the lead vehicle.

Time to collision
Time to collision is defined as how long it would take the own vehicle to collide with another object if the own vehicle maintains its current status.

- Minimum Time-To-Collision: The minimum time-to-collision from event onset until end of the event
- Time-To-Collision at Accelerator Release
- Time-To-Collision at Brake Press
- Time-To-Collision at Maximum Brake
- Adjusted Minimum TTC: Minimum Time-to-Collision adjusted to account for crashes. Negative values indicate how much sooner the driver would need to have responded to avoid the collision.

Data Stream Contents
A large collection of variables are collected at during a person’s drive on the miniSim. At the end of the drive, a DAQ file is written with a large number of variables that log the state of the simulation at each moment during the person’s drive. A DAQ file can be loaded and viewed in the ISAT™ which allows users to playback the drive and view/graph the state of each variable. A free utility called daqConvert is also included that allows users to convert a DAQ file into Matlab or text/ASCII formats. Another free utility called daqViewer is also included that provides a quick way to plot variables from a DAQ file.
Figure 18 - daqViewer Screenshots (MATLAB GUI)
A partial listing of the data acquisition variables are shown below:

**Driver Input Variables**

These variables provide information about the driver’s inputs to the driving simulator. These include information from the steering wheel, accelerator, brake, shifter and turn signal inputs.

<table>
<thead>
<tr>
<th>Driver Input Variables</th>
<th>Definition and Units</th>
<th>Collection Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accelerator Pedal Position</strong></td>
<td>Normalized value between 1 and 0 indicating the accelerator’s position. A value of 0 implies that the accelerator is fully released and value of 1 implies that the accelerator is fully depressed.</td>
<td>60 Hz</td>
</tr>
<tr>
<td><strong>Auto Transmission Mode</strong></td>
<td>Indicates the current gear that the transmission is in. -2 = Park, -1 = Reverse, 0 = Neutral, 1 = First, 2 = Second, 3 = Drive, 4 = Overdrive</td>
<td>Differential</td>
</tr>
<tr>
<td><strong>Steering Wheel Angle</strong></td>
<td>The steering wheel's angle (in degrees). Values typically range between -202.5 to +202.5 degrees. Negative values indicate that the steering wheel is being turned to the left whereas positive values indicate that the steering wheel is being turned to the right.</td>
<td>60 Hz</td>
</tr>
<tr>
<td><strong>Cruise Control State</strong></td>
<td>The state of the cruise control button 0 = Not available, 1 = off, 2 = On, 3 = Set/Accel, 4 = Resume, 5 = Coast</td>
<td>Differential</td>
</tr>
<tr>
<td><strong>Car Horn</strong></td>
<td>1 – off, 2 - on</td>
<td>Differential</td>
</tr>
<tr>
<td><strong>Turn Signals</strong></td>
<td>Indicates the state of the turn signal driver input 1 = no turn signal on, 2 = left turn signal on, 3 = right turn signal on</td>
<td>Differential</td>
</tr>
</tbody>
</table>
Collision Detection Variables
These variables provide information about which objects the driver has collided with during their drive.

<table>
<thead>
<tr>
<th>Collision Detection Variables</th>
<th>Definition and Units</th>
<th>Collection Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collision Count</td>
<td>How many collisions with the ownship vehicle have taken place so far. The value ranges between 0 and n where n is the total of number of collisions in the entire drive so far.</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Collision List Size</td>
<td>Number of objects that are colliding with the driver’s vehicle and are present in the collision list. Values range between 0 and 10. Note: A maximum of 10 collisions are recorded per frame of execution. Priority is given to objects that are on the list in the previous frame.</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Collision Object SOL Identifiers</td>
<td>The SOL identifiers of objects that are colliding with the driver’s vehicle. The SOL identifier gives the user an indication of what type of object has collided with the driver. This is an array of 10 variables. If the value of the Collision_List_Size is n, only the first n values in this array are valid. Each valid value ranges between 1 and n where n is the highest SOL identifier.</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Collision Object Type Identifiers</td>
<td>The type identifier of the objects that are colliding with the driver’s vehicle. This is an array of 10 variables. If the value of the Collision_List_Size is n, only the first n values in this array are valid. Valid values are: 1 – trajectory follower (DDOs) 2 – vehicle (can be ADOs or static objs) 7 – traffic signs 9 – obstacle 13 – walker</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Collision Object CVED Identifiers</td>
<td>The CVED identifiers of the objects that are colliding with the driver’s vehicle. A CVED identifier uniquely identifies the exact object that is colliding with the driver’s vehicle. This is an array of 10 variables. If the value of the Collision_List_Size is n, only the first n values in this array are valid. Each valid value ranges between 2 and n where n is the highest CVED identifier.</td>
<td>60 Hz</td>
</tr>
</tbody>
</table>

Scenario Variables
These variables provide information about the state of the virtual environment and the dynamic objects (other traffic) around the driver.
<table>
<thead>
<tr>
<th>Scenario Variables</th>
<th>Definition and Units</th>
<th>Collection Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>DynObj Data Size</td>
<td>An integer that indicates the number of scenario objects around the driver’s vehicle that information is being logged for.</td>
<td>60 Hz</td>
</tr>
<tr>
<td></td>
<td>The maximum number of objects which data can be logged is 20.</td>
<td></td>
</tr>
</tbody>
</table>
| DynObj SOL Identifiers  | The SOL identifiers of scenario objects around the driver’s vehicle. The SOL identifier identifies the dynamic object from the SOL library. This is an array of 20 integers.  
|                         | If the value of the DataSize is \( n \), only the first \( n \) values in this array are valid. Each valid value ranges between 1 and \( n \) where \( n \) is the highest SOL identifier. | 60 Hz                |
| DynObj CVED Identifiers | The CVED identifiers of scenario objects around the driver’s vehicle. The CVED identifier uniquely identifies an object in the entire simulation. This is an array of 20 integers.  
|                         | If the value of the DataSize is \( n \), only the first \( n \) values in this array are valid. Each valid value ranges between 1 and \( n \) where \( n \) is the highest SOL identifier. | 60 Hz                |
| DynObj Headings         | Headings (in degrees) of scenario objects around the driver’s vehicle. This is an array of 20 floats and objects are sorted with their distance to the driver’s vehicle.                                                                 | 60 Hz                |
| DynObj Names            | Names of scenario objects around the driver’s vehicle. The user may assign names to each object using the ISAT™. This is an array of 20 char arrays. Objects are sorted with their distance to the driver’s vehicle.  
|                         | If the value of the DataSize is \( n \), only the first \( n \) values in this array are valid. Each valid value ranges between 1 and \( n \) where \( n \) is the highest SOL identifier. | 60 Hz                |
| DynObj Positions        | Positions (\( x,y,z \) in scenario coordinates) of scenario objects around the driver’s vehicle. This is an array of 60 floats. Objects are sorted with their distance to the driver’s vehicle.  
<p>|                         | If the value of the DataSize is ( n ), only the first ( n ) values in this array are valid. Each valid value ranges between 1 and ( n ) where ( n ) is the highest SOL identifier. | 60 Hz                |</p>
<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>DynObj Roll and Pitch</td>
<td>Roll and pitch values (i,j,k) of scenario objects around the driver’s vehicle. This is an array of 120 floats. Objects are sorted with their distance to the driver’s vehicle. If the value of the DataSize is n, only the first n values in this array are valid. Each valid value ranges between 1 and n where n is the highest SOL identifier.</td>
<td>60 Hz</td>
</tr>
<tr>
<td>DynObj Audio and Visual States</td>
<td>Indicates the current audio and visual settings for the scenario objects around the driver’s vehicle. This is an array of 40 integers. Objects are sorted with their distance to the driver’s vehicle. If the value of the DataSize is n, only the first n values in this array are valid. Each valid value ranges between 1 and n where n is the highest SOL identifier.</td>
<td>60 Hz</td>
</tr>
<tr>
<td>DynObj Velocity</td>
<td>Velocities (in ft/s) of the scenario objects around the driver’s vehicle. This is an array of 20 floats and objects are sorted with their distance to the driver.</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Lead Vehicle Information</td>
<td>This array provides information about the vehicle in front of the driver. This an array of 9 floats with the following elements: 1st – CVED identifier of object (-1 = no vehicle in front of driver) 2nd - distance to lead vehicle (in feet) 3rd - bumper-to-bumper time to lead vehicle (in seconds) 4th - bumper-to-bumper distance to lead vehicle (in feet) 5th - time-to-collision (in seconds) 6th - lead vehicle velocity (ft/s) 7th – x coordinate of lead vehicle 8th – y coordinate of lead vehicle 9th – z coordinate of lead vehicle</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Lane Deviation</td>
<td>Information about the driver’s vehicle and it’s current lane position. This is array of 4 floats: 1st: -1 (on an intersection corridor), 1 (on a lane) 2nd: lateral offset from the center of lane/corridor 3rd: width of the current lane (corridor’s width is not reported)</td>
<td>60 Hz</td>
</tr>
</tbody>
</table>
Vehicle Dynamics Variables
These variables provide information about the state of the driver’s vehicle during the drive.

<table>
<thead>
<tr>
<th>Vehicle Dynamics Variables</th>
<th>Definition and Units</th>
<th>Collection Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surface Tire Index</td>
<td>0=Intersections and drivable off-road, 14=Road, 20=Shoulder</td>
<td>Differential</td>
</tr>
<tr>
<td>Chassis CG Acceleration</td>
<td>The driver vehicle’s acceleration in ft/seconds$^2$</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Chassis CG angular velocity</td>
<td>The driver vehicle’s angular velocity in degrees/seconds</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Chassis CG Orientation</td>
<td>The driver vehicle’s orientation in degrees</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Chassis CG Position</td>
<td>The driver vehicle’s position (in feet)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Chassis CG Velocity</td>
<td>The driver vehicle velocity (in mph)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Eye-point Orientation</td>
<td>The driver’s eye-point orientation (in degrees) in global coordinates</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Eye-point Position</td>
<td>The driver’s eye-point position (in degrees) in global coordinates</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Head-point Angular Velocity</td>
<td>The head-point’s angular velocity (in degrees/second)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Head-point Specific Forces</td>
<td>Forces being felt at the head-point (in G’s)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Load Torque</td>
<td>Wheel torque due to external forces (in foot lbs.)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Num Grids</td>
<td>Number of grids used for each contact patch</td>
<td>Differential</td>
</tr>
<tr>
<td>Num Tires</td>
<td>Number of tires on the vehicle (0 -10)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Steering Torque Backdrive</td>
<td>Commanded steering wheel torque (in foot-lbs.)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Tire Ground Contact</td>
<td>The tire/terrain contact location (in feet)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Tire Rotational Velocity</td>
<td>Tire rotational velocity (in degrees/second)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Tire Slip Angle</td>
<td>Tire slip angle (in degrees)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Tire Slip Ratio</td>
<td>Tire slip ratio (0.0 – 1.0 normalized)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Tire Weight on Wheels</td>
<td>Tire weight on wheels (in lb. force)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Vehicle Engine RPM</td>
<td>Vehicle engine revolutions per minute</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Engine Torque</td>
<td>Vehicle engine torque (in foot-pounds)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Vehicle Heading</td>
<td>Vehicle heading (in degrees)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Vehicle Speed</td>
<td>Vehicle speed (in mph)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Vehicle Transmission</td>
<td>Vehicle transmission revolutions per minute</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Frequency</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>RPM Vibration Forces</td>
<td>Commanded vibration forces (in G's)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Vehicle Center Heading</td>
<td>Heading angle of wheel (in degrees)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Wheel Center Velocity</td>
<td>Translational velocity of wheel center (in feet/seconds)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Wheel Spin</td>
<td>Wheel spin (in radians/second)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Wheel Spin Angle</td>
<td>Rotational position of tire (in radians)</td>
<td>60 Hz</td>
</tr>
<tr>
<td>Wheel Steer Angle</td>
<td>Road wheel angle (in radians)</td>
<td>60 Hz</td>
</tr>
</tbody>
</table>
ISAT™ (Interactive Scenario Authoring Tool)

The NADS ISAT™ is one of the most full-featured scenario authoring tools available anywhere in the world. This tool, originally built in 1996, is currently in its 4th major revision. For the last 11 years, this tool has been successfully used for both research in the NADS family of simulators and training in the Swiss FATRAN Military Truck Driving Simulator.

The ISAT™ provides a user-friendly graphical interface for developing training scenarios. ISAT features include the following:

- Easy-to-use fully GUI interface, no programming or script writing is required
- Robust control set which allows the creation of training scenarios with a high-level of repeatability and control
- A.I. (artificial intelligence) driven debugging mode for quick testing and off-line perfection of scenarios
- Graphical playback of data files collected from drives of the miniSim™
- Production of media content, including pictures and videos
- Compatibility with the NHTSA scenarios

Please refer to the ISAT™ Users Guide for more information about the countless features of this powerful tool.
TMT™ (Tile Mosaic Tool)

The National Advanced Driving Simulator relies upon the TMT software for construction of correlated synthetic simulation environments. The software and graphics tile libraries are supported and maintained by a team of staff with over 40 years of combined simulation experience.

The TMT uses a modular, extendable library of visual and correlated logical or virtual components. A library of approximately 50 tiles will be included. Each tile is constructed as a self-contained set of files that function as a single entity. Each file set contains visual features, and may be populated with some or all of the following: roads, intersections, signs, collision elements, terrain objects and location markers called placeholders. The set of files is commonly referred to as a singular entity called a Database Tile. The library of modular component Tiles is called a Tile Library. The Tile Mosaic Tool (TMT) is an easy to use, graphical user interface application used to assemble previously created terrain and correlated databases, and publish them into a simulation environment. Simulation rendering performance can be adjusted through the TMT – there is no need to re-work source Tile database files.

Figure 20 - Geo-typical Tile database assembly within the Tile Mosaic Tool
Relationship Between Software Modules

The following diagram shows the relationship between the functionality provided by the miniSim, ISAT, TMT and Matlab. In general, the workflow is as follows:

- The Tile Mosaic Tool (TMT) is used to assemble a road network out of pre-existing tile assets. The TMT creates a visual database and a logical database (the road network) that are used by the miniSim.
- The Interactive Scenario Authoring Tool (ISAT) is used to create a scenario that is specific to the road network. The ISAT can also play back a miniSim data file and display and plot data for scenario testing and debugging purposes.
- The miniSim driving simulator then runs the scenario on the road network using the driver’s inputs and produces a time-history data acquisition file containing all the simulator variables sampled at 60 Hz.
- To reduce the time-history data into final measures, Matlab is typically used along with NADS-supplied graphical user interfaces and tools.
- The miniSim will also produce basic driver performance measures in text format, with no post processing required.
Virtual Environments

The two main components of the miniSim™ virtual environments are the roadmap and model libraries. The combination of these two libraries results in the most realistic driving environment. An example of a scene is presented in Figure 21. The following sub-sections describe the contents of the roadmap and model libraries in more detail.

Figure 21 - A high resolution model and driving environment from the miniSim™ driving simulator

NADS also has proven capabilities in the development of geo-specific virtual environments. These are virtual environments that recreate actual locations in the world to a very high degree of accuracy (within an inch). An example of a geo-specific virtual environment is shown in the following figure. This is a virtual version of an actual dirt racetrack from West Liberty, Iowa.

Figure 22 - The West Liberty raceway (left is the actual location, right is the virtual environment)
NADS miniSim™ Included Tiles and Databases *(Right-Hand-Traffic Only)*

There are approximately **190** database tiles that are included with the miniSim. These represent road segments that can be assembled into custom databases of your own design using the NADS Tile Mosaic Tool (TMT).

Also included are the following databases that have already been assembled and tested:

**NADS TMT v1.2 installed databases**
- **demo**: Databases developed for demonstration purposes
  - **nadsdemo_dsc**: DSC conference demo. Contains urban, rural, and mountain/snow.
  - **nadsdemo_geospecific**: Geo-specific capabilities demo (West Liberty Racetrack)
  - **nadsdemo_kiosk**: Outreach database; 3 separated databases on one terrain, cannot drive from one to the other
- **ESC_Dry_05**: Rural 2 lane database developed to test electronic stability control; open-loop database with decreasing radius curve at north end of terrain
- **ESC_HvyT**: Interstate and Rural 2-lane database developed to test electronic stability control.
- **multipleEnv_01**: Combination city, suburb, freeway, rural database developed to design and evaluate levels of impairment
  - **day_night**: Original study was night only, this version is not the same as the study version due to changing the night version tiles with day tiles.
  - **snow**: This version is not the same as the study version, the rural tiles are snow covered, otherwise the same as the day_night version
- **rural_long**: Rural 2 lane, open-loop database developed to test antihistamine response, repurposed in other studies as well.

---

*Nadsdemo_dsc database*
Nadsdemo_geospecific

Nadsdemo_kiosk

ESC_Dry_05
NADS Demo Database Details
A short roadmap that combines a number of different types of driving environments. Urban, rural scenes are followed a mountainous scene with switchbacks. Day and night-time driving are also featured.
Scenario Object Library (SOL)

The NADS Scenario Object Library (SOL) features a comprehensive library of vehicle models, signs and obstacles. Each vehicle has the capability to exhibit intelligent traffic behavior. Most signs feature a number of options to allow the user to customize the training environment. In total, the model library features over 300 unique elements. The model library is included with the miniSim™.

Vehicles

<table>
<thead>
<tr>
<th>Ambulance</th>
<th>Audi A8</th>
<th>Chevy Blazer</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMW</td>
<td>Cadillac Escalade</td>
<td>Cement Truck</td>
</tr>
<tr>
<td>Beverage Truck</td>
<td>John Deere Combine</td>
<td>Cadillac Deville</td>
</tr>
</tbody>
</table>
Dump Truck
Ford Escape
Freightliner Semi Truck
Ford F150
Ford Expedition
Garbage Truck
Landrover LR2
US Mail Truck
Dodge Neon
Control Signs

Warning Signs

Construction Signs
Route Direction Signs

Speed Limit Signs

Speed limits signs for both posting speed limits on roads and on ramps.
Interstate and Highway Signs
### Objects

<table>
<thead>
<tr>
<th>Image</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Barricade" /></td>
<td>Barricade</td>
</tr>
<tr>
<td><img src="image2" alt="Bump Hump" /></td>
<td>Bump Hump</td>
</tr>
<tr>
<td><img src="image3" alt="Bump with Large Plateau" /></td>
<td>Bump with Large Plateau</td>
</tr>
<tr>
<td><img src="image4" alt="Bump with Small Plateau" /></td>
<td>Bump with Small Plateau</td>
</tr>
<tr>
<td><img src="image5" alt="Pole 1" /></td>
<td>Pole 1</td>
</tr>
<tr>
<td><img src="image6" alt="Pole 2" /></td>
<td>Pole 2</td>
</tr>
<tr>
<td><img src="image7" alt="Concrete Barrier (24')" /></td>
<td>Concrete Barrier (24')</td>
</tr>
<tr>
<td><img src="image8" alt="Concrete Barrier (100')" /></td>
<td>Concrete Barrier (100')</td>
</tr>
<tr>
<td><img src="image9" alt="Cone" /></td>
<td>Cone</td>
</tr>
<tr>
<td>Image</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td><img src="image" alt="Crash Barrel" /></td>
<td>Crash Barrel</td>
</tr>
<tr>
<td><img src="image" alt="Diagonal Stripes" /></td>
<td>Diagonal Stripes</td>
</tr>
<tr>
<td><img src="image" alt="Drum" /></td>
<td>Drum</td>
</tr>
<tr>
<td><img src="image" alt="Halt Line with White/Red" /></td>
<td>Halt Line with White/Red</td>
</tr>
<tr>
<td><img src="image" alt="Bouncing Ball" /></td>
<td>Bouncing Ball</td>
</tr>
<tr>
<td><img src="image" alt="Desk (falling off Vehicles)" /></td>
<td>Desk (falling off Vehicles)</td>
</tr>
<tr>
<td><img src="image" alt="Foam Pad" /></td>
<td>Foam Pad</td>
</tr>
<tr>
<td><img src="image" alt="Suitcase" /></td>
<td>Suitcase</td>
</tr>
<tr>
<td><img src="image" alt="Reflector Pole" /></td>
<td>Reflector Pole</td>
</tr>
<tr>
<td><img src="image" alt="Parking Meter" /></td>
<td>Parking Meter</td>
</tr>
<tr>
<td><img src="image" alt="Pedestrian Clutter" /></td>
<td>Pedestrian Clutter</td>
</tr>
<tr>
<td><img src="image" alt="Pedestrian Clutter 2" /></td>
<td>Pedestrian Clutter 2</td>
</tr>
<tr>
<td>Smoke</td>
<td>Street Light</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td><img src="Image" alt="Smoke" /></td>
<td><img src="Image" alt="Street Light" /></td>
</tr>
<tr>
<td>Dethatch</td>
<td>Stone rake</td>
</tr>
<tr>
<td><img src="Image" alt="Dethatch" /></td>
<td><img src="Image" alt="Stone rake" /></td>
</tr>
<tr>
<td>Deer</td>
<td>Dog</td>
</tr>
<tr>
<td><img src="Image" alt="Deer" /></td>
<td><img src="Image" alt="Dog" /></td>
</tr>
<tr>
<td>Walker 2</td>
<td>Walker 3</td>
</tr>
<tr>
<td><img src="Image" alt="Walker 2" /></td>
<td><img src="Image" alt="Walker 3" /></td>
</tr>
<tr>
<td>Ped Group 2</td>
<td>Ped Group 3</td>
</tr>
<tr>
<td><img src="Image" alt="Ped Group 2" /></td>
<td><img src="Image" alt="Ped Group 3" /></td>
</tr>
</tbody>
</table>
Walker 5

Phone Truck

Road Construction 1

Road Construction 2

Road Construction 3

Road Construction 4

Road Construction 5

Taxi

Road Construction 6

Road Construction 7

Road Construction 8

Road Construction 9
<table>
<thead>
<tr>
<th>Quigley</th>
<th>San Ruis</th>
<th>Trayer</th>
<th>Westchester</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rapid City</td>
<td>Shelby</td>
<td>Underwood</td>
<td>Ypsilanti</td>
</tr>
<tr>
<td>Ringgold</td>
<td>Streeter</td>
<td>Vancouver</td>
<td>Zurich</td>
</tr>
<tr>
<td>Riverside</td>
<td>Taylor</td>
<td>Vinton</td>
<td></td>
</tr>
<tr>
<td>Sagatauk</td>
<td>Traverse City</td>
<td>Warren</td>
<td></td>
</tr>
</tbody>
</table>

### Street Signs

<table>
<thead>
<tr>
<th>Adams Ave</th>
<th>Fox Ln</th>
<th>Martin St</th>
<th>Queen Ave</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adams Dr</td>
<td>Fox Rd</td>
<td>Martin Way</td>
<td>Queen Dr</td>
</tr>
<tr>
<td>Adams Ln</td>
<td>Fox St</td>
<td>Neff Ave</td>
<td>Queen Ln</td>
</tr>
<tr>
<td>Adams Rd</td>
<td>Fox Way</td>
<td>Neff Dr</td>
<td>Queen Rd</td>
</tr>
<tr>
<td>Adams St</td>
<td>Holt Ave</td>
<td>Neff Ln</td>
<td>Queen St</td>
</tr>
<tr>
<td>Adams Way</td>
<td>Holt Dr</td>
<td>Neff Rd</td>
<td>Queen Way</td>
</tr>
<tr>
<td>Cobb Ave</td>
<td>Holt Ln</td>
<td>Neff St</td>
<td>Taylor Ave</td>
</tr>
<tr>
<td>Cobb Dr</td>
<td>Holt Rd</td>
<td>Neff Way</td>
<td>Taylor Dr</td>
</tr>
<tr>
<td>Cobb Ln</td>
<td>Holt St</td>
<td>Park Ave</td>
<td>Taylor Ln</td>
</tr>
<tr>
<td>Cobb Rd</td>
<td>Holt Way</td>
<td>Park Dr</td>
<td>Taylor Rd</td>
</tr>
<tr>
<td>Cobb St</td>
<td>Martin Ave</td>
<td>Park Ln</td>
<td>Taylor St</td>
</tr>
<tr>
<td>Cobb Way</td>
<td>Martin Dr</td>
<td>Park Rd</td>
<td>Taylor Way</td>
</tr>
<tr>
<td>Fox Ave</td>
<td>Martin Ln</td>
<td>Park St</td>
<td></td>
</tr>
<tr>
<td>Fox Dr</td>
<td>Martin Rd</td>
<td>Park Way</td>
<td></td>
</tr>
</tbody>
</table>
Guide Signs

Adel 38  Ames 33  Club St  Detroit 20  Dike 26

EXIT 23  High St  Main Street 3  Olin 20  Park St

Scott County Park  Vine St  Weigh Station
**Rail Functionality**

New functionality has been developed to support railroad crossings. This functionality includes new railroad vehicles, objects, tiles and operational capability. This implementation includes typical railroad crossing objects, terrain rail objects and associated signs.

**Rail Vehicles**

A limited set of rail vehicles has been implemented. These models are designed to operate along straight tracks only; the train model which consists of several cars does not articulate as the train traverses curves. The rail vehicles may be used as static, stationary objects as would be found on a railroad siding or as deterministic dependent objects (DDOs). DDOs require a travel path to be defined. This mechanism permits the rail vehicle to travel both directions along what is essentially a one-way road (track). Without the path, vehicles will always travel downstream, from the origin of the track towards the end of the track.

![railveh_train_01](image1)

![railveh_boxcar](image2)
railveh_flatcar_steel_rolls

railveh_tanker_Gold_West

railveh_Union_Pacific
Railroad Tiles
The following railroad tiles are available: General-purpose rail tiles: 1X straight (single track), 3X straight (triple track), and at-grade crossings for rural and city environments. There are single and triple at-grade crossings, and one blank 1X and 3X rural crossing to permit total control over the crossing environment.
miniSim Options

Custom Virtual Environment Development

NADS excels in creating virtual environments (roadmaps and objects) custom-designed to suit client specifications. There are several components that determine the realism or fidelity of a graphical virtual environment. There are:

- A **geometry** consists of the vertices and triangles used to define polygon triangles which create the visual environment. Objects that are visible within the simulation environment are created with geometry or particle systems.

- A **texture** refers to bitmap imagery applied to geometry to provide detail. Texture is applied to geometry to give it color, depth and resolution. Texture is generally required to be within specific size constraints per image, and to fit within the overall texture budget of target simulator for each database.

- **Shading** is the application of vertex normals to geometry. When used in conjunction with light sources within the simulation, shaded surfaces appear to contain crisp surface definition (i.e., two walls at a corner). Shading may also refer to application of shaders.

- A **model** is a visual representation of something - a car, sign, streetlight, etc. Models may contain additional attributes or characteristics available for control during simulation, or may be limited to a visual representation within the environment.

- **Feature density** refers to the amount or frequency of features represented within the simulation environment. Generally the environment is divided into terrain (this includes terrain and roadways) and features. Trees, signs, traffic lights, buildings are examples of features.

- It is possible to utilize multiple **data sources** for simulation features and environments. Data sources may consist of one or more of the following: specifications, 3D models, 2D textures (i.e., Google Earth™), GIS data (including GPS coordinates, ARC™ coverages, digital elevation data), LIDAR samples, etc. Multiple data sources frequently require calibration and registration between the different sources.

- From the standpoint of simulation environments, **resolution** and **fidelity** are used to refer to the accuracy of the environment reproducing real-world features or locations.

- A **correlated database** is another representation of or additional, non-visual characteristics and attributes of a specific simulation environment.

Taking these into account, NADS offers 3 levels of realism when it develops graphical virtual environments. The realism increases with higher levels. Figure 23 presents an example of two similar virtual environments at levels 1 and 3.
Figure 23 - Examples of different texture types, feature density, overall appearance between Levels 1 and 3.

Table 1 identifies the overall graphics components associated with each level of a virtual environment.

<table>
<thead>
<tr>
<th>Level</th>
<th>Texture</th>
<th>Shading</th>
<th>Geometry</th>
<th>Features/Density</th>
<th>Data Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Simple pattern</td>
<td>None</td>
<td>Simple</td>
<td>Simple/Low</td>
<td>Low</td>
</tr>
<tr>
<td>2</td>
<td>Generally simple pattern, limited complexity, generic representation</td>
<td>Yes</td>
<td>Simple, generally geo-typical with some geo-specific representation</td>
<td>Simple/Low to medium</td>
<td>Low</td>
</tr>
<tr>
<td>3</td>
<td>Generally complex, photographic, specific representation</td>
<td>Yes</td>
<td>Complex; generally geo-specific with some repeated elements generalized</td>
<td>Complex/High</td>
<td>High</td>
</tr>
</tbody>
</table>

Table 1 - Characteristic of graphical virtual environments at Levels 1, 2 and 3.

Whereas Level 1 represents virtual environments that are simpler, Level 3 represents virtual environment that are very rich and full of features. Level 3 also consists of virtual environments that feature areas so realistic that they are indistinguishable from actual world locations. Examples of virtual environments at Levels 1 (Figure 24), 2 (Figure 25) and 3 (Figure 26) are presented below.
Figure 24 - An example of a Level 1 virtual environment

Figure 25 - An example of a Level 2 virtual environment
Figure 26 - Examples of Level 3 virtual environments

Custom development of a graphical virtual environment is heavily dependent upon the geographic region of interest (ROI), the quantity and quality of features to be developed, and the amount of individuality/specificity to be utilized (to what degree may elements be recycled within the environment). In general, detailed, highly representative ROI results in longer development efforts and greater cost.
Examples of Geo-Specific Database Development

Here, the objective was to develop a highly accurate representation of a specific location. A geo-specific database is intended to re-create a virtual space that is recognizable by persons familiar with the real world location. In extreme cases the majority of surfaces present in the virtual environment are built 1:1 to match the real world.

Examples of geo-specific environments

Real world locations: Iowa City, Iowa and West Liberty Raceway, Iowa, Churchville at Aberdeen Proving Grounds, Munson Test Center at Aberdeen Proving Grounds
Examples of Geo-Typical Database Development
For a geo-typical environment, the research or training team determines which characteristics are necessary and those specific characteristics are created for use within the simulation environment. This effort may be primarily generic in nature, with the goal of providing not a specific location but a broad range of potential locations intended to address desired training needs or research goals. The biggest consideration here in addition to cost is these environments are designed to present static and unchanging road networks and terrain, although it is possible to intermingle them with both geo-specific and standard tiles and extend their functionality.

Town and Hill1 geo-typical macro tiles

Geo-typical mountain macro tile and Tunnel tile
Examples of geo-typical macro environments built to specification

Mobile Trailer Installation Option
NADS engineers have experience in installing the miniSim™ inside mobile platforms. This optional service is also available for an additional fee. Below is example of the Verasity™ driving simulator (based on miniSim™ technology) which was built and installed by NADS engineers. In additional to trailers, the miniSim™ can be installed in larger vans.

Figure 27 - The Verasity driving simulator based on miniSim(TM) components (photo courtesy of Tom Zwica).

Figure 28 - Interior view of miniSim(TM) driving simulator installed in the back of a mobile trailer (photo courtesy of Tom Zwica).
Another mobile application was the installation of a ¼ cab miniSim in a 16ft trailer for the Center for Transportation Research and Engineering (CTRE) at Iowa State University. NADS developed the artwork for the trailer and engineered and installed the simulator.

Figure 29 – Mobile Driving Simulator
4-Channel Video Capture System Option
An optional accessory of the miniSim is a 4-Channel video-capture system or VidCap™ for the NADS miniSim driving Simulator. The system will record up to 4 SDI HD (1920x1080) Video Inputs that are combined into a single, HD channel with audio. Real-time simulator variables such as frame number and driver speed can be super-imposed over the recorded channel.

The VidCap PC is connected to the miniSim via an Ethernet connection. The purpose of this is to allow the VidCap machine to start and stop recording video in sync with the miniSim start/run states and to provide the DAQ file name to the Vidcap, which will use the same and file name location for the video files. An application is provided to re-encode the video to save space. A sample of the quad-split output is below.

VidCap System Output with Quad Split

Typical SDI HD (1920x1080) Video Cameras used for VidCap
Eye Tracking Integration Options

The miniSim is currently compatible with several Eye Tracking systems:

- **Ergoneers** [www.ergoneers.com](http://www.ergoneers.com)
  - Dikablis and Tobii Eye Trackers
  - D-LAB Analysis workspace
  - NADS D-LINK to stream sim data to D-LAB
- **EyeTracking Inc.** [www.eyetracking.com](http://www.eyetracking.com)
  - FOVIO™ single and multi-camera systems (other listed on website)
  - EYEWORKS™ analysis tool
  - Cognitive Workload
- **TOBII** [www.tobii.com](http://www.tobii.com)
  - Head-mounted eyetracker systems
- **SmartEye** [smarteye.se](http://smarteye.se)
  - Multi-Camera Eye Tracking Systems
  - MAPPS
NADS Springfield 2014 Road Network Option
The Springfield Road Network consists of a large Interstate Highway loop, rural roads, and urban and residential areas. NADS will supply this road network with sample scenarios and external reference scenarios that set the traffic, signs, and signals. Ambient oncoming traffic is provided via the ISAT traffic manager function. More complex custom scenarios can be delivered to meet customer specifications at additional cost.

- Built, Tested, and Ready to Go!
- Ambient Traffic
- Large, Diverse Environment
  - 285 square miles
  - 230 miles of roadway
  - 178 intersections
  - 143 traffic signals
  - 1362 signs
- Supports many applications
  - Automation Development
  - UI Testing
  - Distraction
  - Outreach, Education

Springfield Road Network Map (approx. 19 x 15 miles)
Springfield Road Network Screenshots
**Logical Database API**

This Logical Database API is to provide a simplified programmer interface into CVED, to serve as a method to extract road information at run time on the MiniSim. This API will not directly serve as an interface into the MiniSim, it will however be able to open the BLI file and run parallel with the MiniSim. As part of the delivery an example program that does provide a communication interface to the MiniSim, and utilizes this API will be delivered. The below figure shows the data-flow, where the example program receives the current state information including the position of the driver, the example program then uses this information to query the API.

Definitions:

<table>
<thead>
<tr>
<th>CVED: Correlated Virtual Environment Database is an API used internally at NADS to read the bli, and query the road network.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLI: Binary Logical road Interface, this is a file that contains all of the logical road information.</td>
</tr>
<tr>
<td>SOL: The sol file provides static information about static or dynamic objects, such as the bounding box size</td>
</tr>
<tr>
<td>Corridor: A corridor is a defined path through an intersection</td>
</tr>
<tr>
<td>Static Object: Static objects include signs, and non-moving objects that are either part of the BLI, or inserted into ISAT by the scenario author.</td>
</tr>
<tr>
<td>Dynamic Object: Dynamic objects are those objects that when created by running the scenario, have a physical presence in the simulation, and are updated by scenario control. These are either ADOs or DDOs.</td>
</tr>
</tbody>
</table>

![Figure 30 MiniSim Data Flow](image-url)
The API is developed in C++; it will be delivered as a library file built in VS 2010, with a header files. The API is object oriented in design and will use Hungarian notation (classes begin with C, member variable will begin with m_). The example program will be provided as a source code distribution, with a project and solution file for VS 2010. This program will be developed with C++. The operation of the example program will require network connectivity with the MiniSim. The API will require in the installation directory of the executable:

- a copy of the BLI
- a copy of the SOL file
- a copy of the scenario file

Description of API Functionality:
The API provides an interface that will allow a user to query logical information about the driving environment, such as lane width, distance to next intersection. The API will accept x,y,z locations and provide road related data about said location. This shall have real-time level performance. It also shall be possible to use this API in a system that is not directly connected to the MiniSim.

The example program will communicate with the MiniSim, and will utilize at least one version of every API available, duplicate calls that only vary in call type (i.e. float vs. double) may only have 1 version called. This program will provided as an example of the API, and will provide validation of the API.

Road Information
The API will provide a series of functions that allow the user to input an X,Y,Z location and get logical information at that location, including lane width, road marking type, surface type. Only a minority of tiles have tags (or attributes) that allow road marking information to be queried by CVED, hence this API will only provide road marking information from BLIs that a marked with tags for road markings. Should a road position not have valid markings tag, a return code will indicate no information was available.

GetLaneWidth()
This will return the current width of the current lane at a given XYZ location.
GetRoadMarkings()
This will return lane marking type, on the right and left of the lane. This shall also provide
distance from center line to inside of lane markings at a given XYZ location.

GetLaneLayout()
This function shall return the number of lanes, the direction of each lane, and meta-data
about the lane, given a XYZ position; this will not function in intersection.

GetLaneSurfaceType()
This function will return the road surface type at the current location. Documentation will be
provided as to what the surface type codes indicate.

GetRoadTrace()
This function will take the current position, and a distance, and return an array of x,y,z points
and lane width, following road. A version of this function will use the driver’s path as specified
by the scenario, and use that to navigate any intersections that the function may encounter.
Another form of the function will take a direction in, and will navigate the intersection using a
given direction (right, left, straight). Another version will take a corridor id, and provide a
trace through the corridor.

GetOncomingIntersection()
Forms of this function will return: the distance to the next intersection 0, if currently located
on an intersection, and list of connected corridors, and their respective turn directions.

GetOncomingHaltLine()
This function will return the distance and x,y,z, location of the next oncoming halt line,
version of this function will take the drivers path to determine what halt line to stop at.

GetOncomingMergePoints()
This function will take a distance, and position, it will project forward, and return distances to,
and x,y,z location for points where two corridors begin to overlap.

GetLightState()
This function will take in location information and the traffic light state data past from the
MiniSim, and will indicate what the traffic light state is for said location.

GetCurvature()
This function will take xyz location, and return the radius of curvature for said location.

GetGrade()
Given an x,y,z get the road grade of the road in degrees.

ChangeLanes()
Given a position (XYZ) on a road segment (not in an intersection) and a lane offset(+
meaning lanes to the right, negative numbers mean to the left, this function will return center
of the lane indicated by the lane offset or an error code indicating why it cannot changes
lanes( i.e. the lane does not exist, or is an oncoming lane).
Static Object Functions

GetStaticObjects()
Given a bounding box, this function will return a list of static objects, this will include x,y,z information and sol ID. Most signs will provide a x,y,z location, where this position will provide the bottom of the post of the sign (for signs with a single post). The height of the sign will be derivable from GetSolInfo. However not every sign’s position is directly related to its position in the BLI (some signs such as interstate with many components do not match the actual position as to ease authoring and viewing the sign in ISAT). Some guidance will be provided as what signs can have their positions directly derived from the bli.

Dynamic Object Functions

UpdateDynamicObjects()
This function takes in the dynamic object data outputted from the MiniSim, and adds entries into CVED for each object.

GetDynamicObjects()
Given a bounding box, this function will return a list of dynamic objects, this will include x,y,z information and sol ID.

GetTerrainObjects()
Given a bounding box, this function will return a list of terrain objects; terrain objects provide information about the location of additional terrain information, such as parking lots, or buildings. This tends to only be a much smaller subset of what is in the visual database, this API will only allow access to terrain objects that have been placed in the BLI file.

GetSolInfo ()
Given sol ID (that which is part of the dynamic object data), this function will return information from the SOL file, including the minimum bounding box for the object.

Example program
An example program is provided, this program:

Utilizes every query available to the API
Every query function accessible to the API shall be called at least one

MiniSim Connectivity
The sample program shall receive all data that the query API may utilize, and properly load the data into the API’s objects.

Full Source code and project file will be provided
The source code and project files will be provided for the sample program, the end user shall be able to build this program using Visual Studio 2010.

Vehicle Automation Option
The NADS team have developed a set of low-level behaviors that can be activated by the scenario, by manual action, or by a user-designed external subsystem. There are also functions for transfer of control, high-level driving style parameters, and support for existing scenario control triggering systems. Along with the logical database API, it provides the user with the tools to develop their own
high-level systems for executive control. The functions are as follows (excerpts from the
development document):

Lane Following
A low-level controller has been implemented to allow the ownship vehicle to following the lane in
which it is traveling. The controller works over a range of speeds from urban to highway driving,
and is able to work on curved roads. The controller does not manage changing lanes, making
turns, etc. These functions are covered in new functions described below. The control system
that has been implemented in Simulink shall be integrated into the MiniSim software.

Speed Following
An adaptive cruise control (ACC) system has been implemented in vehicle dynamics. This
system was designed for an ACC study and modeled after a Toyota Avalon tested at that
time. There is a deceleration limit on the ACC authority; and the system does not work below 25
mph or respond to objects that are moving less than about 5mph. This system shall be
enhanced/modified as needed to work smoothly with the other automated functions.

Lane Change
A lane change maneuver shall be implemented that is distinct from turns and exits, such that it
may occur on any section of roadway, including curved roads. The lane change may occur to the
left or to the right, but shall not intrude into an oncoming lane, and thus shall not accommodate
some passing maneuvers. The lane change maneuver shall be incorporated into high level
control logic as needed to accommodate other maneuvers. For example, a left turn may be
specified that would require a lane change into a left turn lane. The system shall be able to
recognize this and request a lane change preceding the left turn.

Brake to Stop
Most intersections have stop lines at which vehicles are expected to stop in the presence of a
stop sign or a red light. The automation controller shall recognize stop lines and be able to bring
the vehicle to a complete stop at such locations. The ability to cause the ownship vehicle to
brake to a stop at an arbitrary location shall also be provided. This function may be used to
simulate a system failure or some other experimental manipulation. The high level logic around
this maneuver shall not, at this time, recognize the presence of stop signs or the state of traffic
signals; however, that capability may be added at a later time.

Merge / Exit
The ownship vehicle shall be able to autonomously merge onto highways and exit off them. The
controller in these situations shall be able to accommodate the interchange types most often
encountered in NADS databases (ex: overpasses and cloverleafs). If there is a choice of turn
direction after exiting a highway, then the operator must specify a turn as they normally would at
any other intersection.

Turning
The automation shall be able to make the ownship vehicle turn left or right at an intersection,
whether that intersection is a 4-way, 3-way, or Y intersection. Two variations of turning shall be
provided. The first is a stop, turn, go maneuver in which the vehicle comes to a complete stop
before proceeding. The second is a slow, turn, go maneuver in which the vehicle rolls into the
intersection and navigates through it at a comfortable speed. If there is a choice of destination
lanes, then the controller shall choose the rightmost lane on a right turn and the leftmost lane on
a left turn.
Pull Over
The automation shall be able to make the ownship pull over to the shoulder or curb of the road to a stop. The purpose of this behavior is to provide an acceptable ‘default’ behavior in cases where the driver is disabled or unable to act.

Abort
The automation shall allow the occupant to abort an action and revert to the previous valid action. For example, if the car is attempting a lane change and the passenger wishes to abort, the car will remain doing the last valid behavior (in this case, lane following).

Transfer of Control
Mechanisms shall be provided to allow the driver to transfer control to the automation as well as take back control from it. A driver interface shall be provided in some form to notify the driver of the current state of the automation. The operator shall be able to specify a total or partial transfer of control, where partial refers to transferring one axis of control while leaving the other in its current state. Transfer from the automated to manual control shall be able to be triggered by the operator to simulate a system failure of some kind.

Driving Style
The automation controllers shall be able to perform all the maneuvers described above in a consistent manner, similar to how a human driver would perform them. Moreover, the operator (and perhaps the driver) shall be able to set the driving style from among a limited set of options ranging from more conservative to more aggressive, in order to better match driver’s preferences.

Triggers
The operator shall be able to issue ‘turn-by-turn’ instructions to the automation in order to simulate an automated route navigation feature. If the route is fixed and pre-determined, then these instructions may be embedded in the scenario as triggered behaviors. However, they may also be issued manually by the operator as the drive is progressing. For example, if the operator issues a command to turn right, it shall be understood by the navigation controller that a right turn is requested at the next available opportunity. An operator interface of some kind shall be provided to allow the operator to interact with the navigation controller.
**User-Defined Simulator Subsystem**

The miniSim subsystems are state machines, whose current states are controlled through the user interface. A user can create a custom subsystem to run with the rest of the MiniSim simulator. The ability to link user developed functions to the subsystem has many applications, including the incorporation of active vehicle safety systems or automation functions into the miniSim driver in the loop simulation environment. It is also possible to build custom MatLab functions into static library files, include their accompanying C++ header files in the source code for the custom subsystem, and link the static libraries into the project the same way the libraries for the subsystem base class are linked in.

Custom subsystems can be added to the miniSim with little change to the rest parts of the system. The custom subsystem can be built from the subsystem base class which provides an inter-subsystem data communication interface and the functionalities of the real-time scheduling core that monitors state changes. The user layer registers input and output data elements, and executes custom functions that based on the current state of the simulator, interpret the input data and generate outputs.

NADS will provide code for a functioning example subsystem. Included in the package are the required header and library files to write the code for a subsystem, as well the sample source code for a simple subsystem which does data input and output via the element registration scheme and handles state changes. The MiniSim software was written in C++ and built with Microsoft Visual Studio.Net solution and project files for the sample subsystem are included.