

# Experiment Specification



## CONTENTS

Introduction .....	2
ExpPrep .....	2
Data Definitions .....	3
Subjects .....	3

## Introduction

An experiment is run from Control.exe. To start an experiment from Control.exe, either select:

- a binary script (extension .scb) or
- an experiment definition file (extension .ext)

How to make a binary script is defined elsewhere (see **scenario.pdf**). Basically, a SCRIPT file is generated via a text editor as a \*.scn file. To run it in the simulator, it has to be compiled into an \*.scb file, which is a binary script file. The SCRIPT file contains all statements and functions to run an experiment. It also specifies which world database to use. There are a number of ready to use world databases that can be addressed. Databases can be inspected, modified and created in the program RoadDesign.exe (see **RoadDesign.pdf**). A world database consists of 2 files, that are generated when a database is saved from RoadDesign.exe:

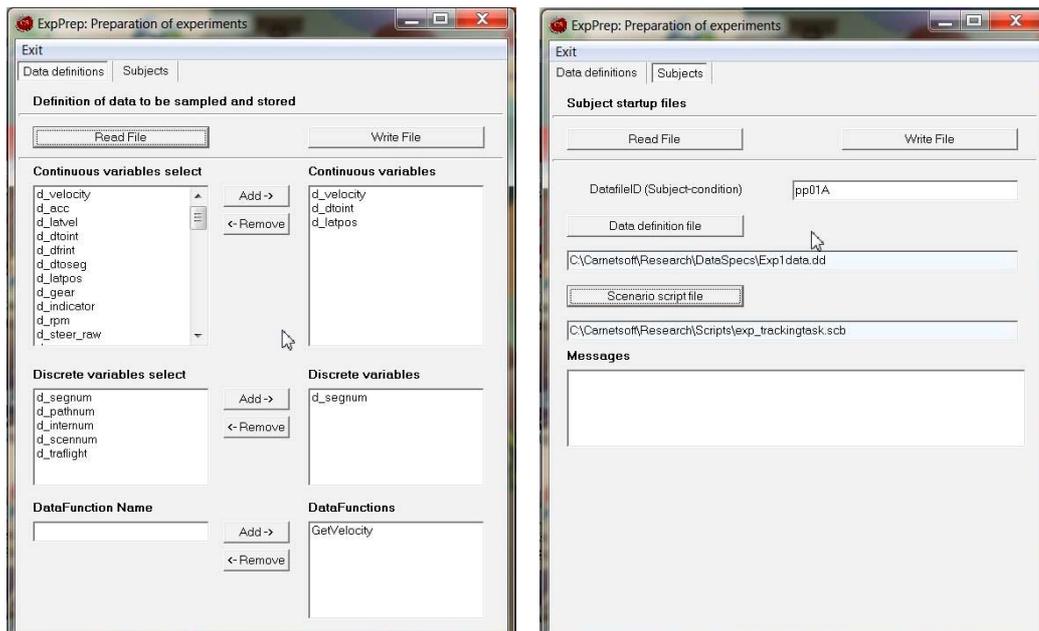
- a logical database, extension \*.net. The simulator loads this from the \SimCarnet\scenegraphs\ folder.
- A graphical database, extension \*.egg. The \*.egg file (output from RoadDesign.exe) must be compiled into a \*.bam file (binary egg) plus a \*.ref file (for position of 3D objects). These graphical database files are loaded from the \models\ folder.

Here we will discuss the experiment definition file (EXP file). An EXP file is simply a text file that contains 3 lines. It can be created manually or with the ExpPrep program. Lets assume that all experiment definition files are located in the \ExpDefinitions\ folder. It is recommended to save an EXP file in this folder from ExpPrep.exe.

## ExpPrep

ExpPrep is an application that creates 2 types of files:

- Data definition files (extension \*.dd). DD files are referenced in the EXP files.
- Experiment definition files (extension \*.exp)



## Data Definitions

A DD file contains code that tells the simulator which data to sample and store. A DD file is created via the Data definitions tab. An example of the contents of a DD file:

```
1      3
1      6
1      9
2      10
3      GetVelocity
```

Each line has items. The first item (1, 2 or 3) is the type of variable:

```
1      Continuous
2      Discrete
3      User defined function name
```

The second item is the symbol for that variable. For example, the symbol 3 means velocity of the simulator car in m/s. All symbols are declared in the include script file DataProc.sci. See also **DataStorage.pdf**.

When the variable type is 'User defined function name', the actual function name is referred to, in this example GetVelocity. This must be a function that is specified in the script and always returns a value. As an example it could be defined as follows, to return the speed in km/h of the simulator car (which is Part[MainTarget]):

```
Define Function GetVelocity() {
  GetVelocity := 3.6*Part[MainTarget].Velocity;
}
```

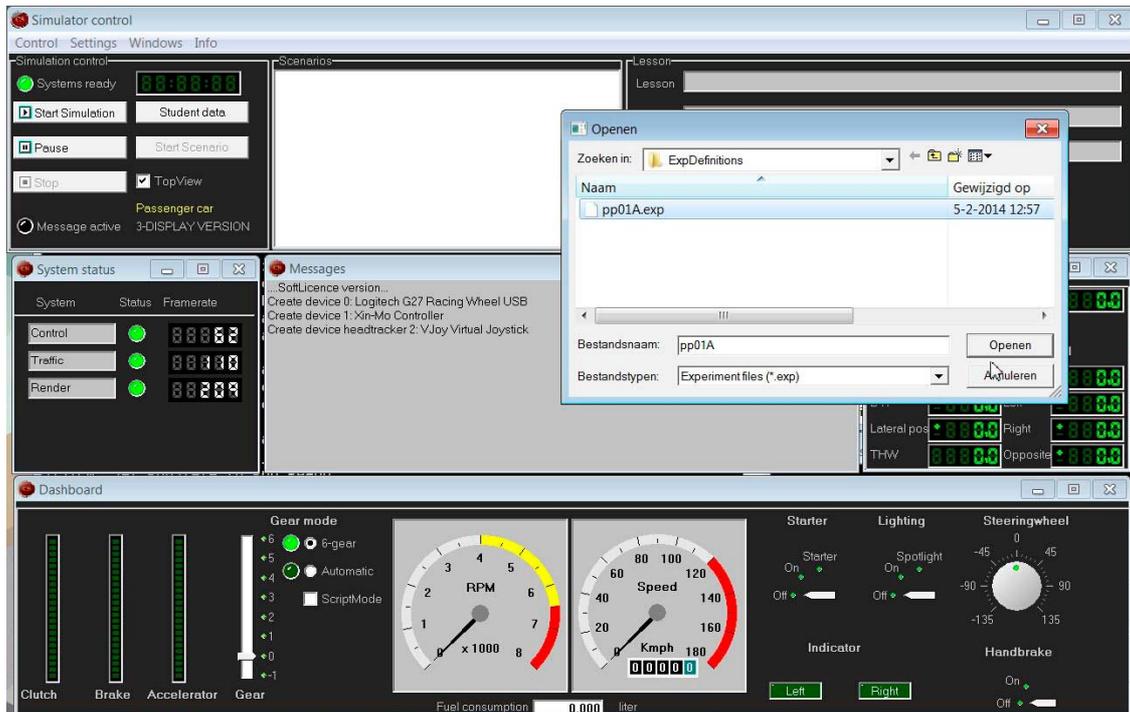
In ExpPrep, all variables to be sampled with a fixed frequency (usually 10 Hz) and stored can be selected as continuous, discrete or user defined function variables. The maximum number of data variables to be stored in this way is 32.

## Subjects

The experiment data of each subject are stored in a separate EXP file. The EXP file is created via the Subjects tab. The EXP file has 3 lines:

- Name of the data file (Datafile ID), for example PP01A. This can be the combination of a subject ID (PP01) plus a condition (A).
- Name and location (full path) of the DD file (data definitions), for example:  
c:\DriveSim3\Research\DataSpecs\Exp1data.dd
- Name and location (full path) of the SCRIPT file (binary version), for example:  
c:\DriveSim3\Research\scripts\exp\_TrackingTask.scb

To run this file, start the simulator, and in control.exe press <Start Simulation>. Then search for the appropriate .exp file and start it, see next figure. More information on control.exe and how to start a simulation can be found in **CarnetManualEN30.pdf**.



When an EXP file is read (instead of a SCRIPT file, extension \*.scb), data definition and subject names are extracted and read in the utility include script DataProc.sci, in the function StartDataSampling():

```

a := ExpDataDefined();
If ( a = 1 ) {
  D_BaseFName := SubjectIdent(); // get the subjectid as send from the experiment specification file
  EXT_DataDef := DataDef(); // get the data specification filename
  Proc( OpenFileRead, EXT_DataDef );
  DataType := ReadNumber( EXT_DataDef );
  While ( DataType >= 0 ) {
    If ( DataType = 1 or DataType = 2 ) {
      DataCode := ReadNumber( EXT_DataDef );
      Proc( AddDataVariable, DataCode );
    }
    Else {
      FuncName := ReadString( EXT_DataDef );
      Proc( AddDataFunction, FuncName );
    }
    DataType := ReadNumber( EXT_DataDef );
  }
}
Else {
  // No exp file found
  Proc( Print, "No Exp data defined...");
  D_BaseFName := "testpp";
  Proc( AddDataVariable, d_velocity );
}
// open binaire file
D_BinName := strcat( D_BaseFName, D_ExpCondition );
Proc( OpenData, D_BinName, D_FileText ); // open binary file and store a string to the header
D_DataSamplingOn := True;

```