SOFTWARE USER'S MANUAL (SUM)

FOR THE

“OBA Testbench”

OF CSCI

“OBA Virtual Target Supplies”

for Windows NT

CONTRACT No E011007

CDRL SEQUENCE No 7

Prepared for:
THALES Université Coopération, Buc

en

THALES UNIVERSITE COOPERATION	NUMERO DOCUMENT / DOCUMENT NUMBER	FORMAT / SIZE	PAGE
xxxx	56 699 445-108	A4	1/16
	D C B A -		REV

CHANGES

REVISION	DESCRIPTION
A	Changes from specification review (23/11/01)
B	VDD CTD, Result file names, Driver's window example, Driving controls, separate § added for dashboard, Tuning facilities, Road file format
C	p1 :Thalès Logo All pages : CSCI N° §2 : SPM title & SSS CTD
D	§3.3 : addition of a procedure for Linux OS Appendix B : precisions and change of results file's extension
E	
F	

ind. + Date	– 17/10/2001	A 26/12/2001	B 18/01/2002	C 28/02/2002	D 17/10/2002	E	F
Written by	D. Boucon	D. Boucon	D. Boucon	D. Boucon	M.Sola		
Checked by	D. Jeanjean	D. Jeanjean	D. Jeanjean		C. Mercier		
Approved by	D. Boucon	D. Boucon	D. Boucon		C. Mercier		

TABLE OF CONTENTS

1. SCOPE

1.1 IDENTIFICATION

"OBA Virtual Target Supplies" Configuration Item Number is : 56 699 445.

The OBA Test bench is one of the parts of this CSCI.

1.2 CSCI OVERVIEW

The CSCI "OBA Virtual Target Supplies" (OBA_VTS) is intended to give a simulation environment for a student's development project, named "OBA software".

This complete environment (OBA_VTS) for executing and testing OBA software is a kind of virtual target built upon the operating system of the workstation used for OBA software development. It allows for running the OBA software on a development host machine, instead of the real target.

The OBA Test bench is the user's view of this environment.

The Virtual Target API is the description of the programming interface between the OBA software and the OBA Testbench : its use is described in the *Software Programming Manual of OBA Virtual Target API* [SPM].

1.3 DOCUMENT OVERVIEW

The present document describes the use of the OBA Testbench of CSCI OBA_VTS, for the Windows NT version.

Chapter 2 list the documents referenced in the present document.

Chapter 3 is the main part of the document and describes the operational usage of the OBA Testbench.

Chapter 4 list the error messages of the OBA Test bench (i.e. NOT those of the OBA Software).

Appendix A describes the road definition file format.

Appendix B describes the result file format.

2. REFERENCED DOCUMENTS

[SSS] System/Segment Specification of Safe Drive Subsystem, 31 000 100-305

[IRS] Interfaces Requirements Specification of Safe Drive Subsystem, 31 000 100-506

[SPM] Software Programming Manual of OBA Virtual Target API, 56 699 445-508

[SRS] Software Requirements Specification of OBA Testbench, 56 699 445-306

[VDD] Version Description Document of OBA Virtual Target Supplies, 56 699 445- 498

3. EXECUTION PROCEDURES

3.1 OVERVIEW

The OBA Testbench simulates all of the real environment of the OBA software :

- A car running on a road (the car simulation model),
- Actions of the driver on the car, such as acceleration, brakes, changing gear, etc. (the driving station),
- The Safe Drive control panel.

The underlying virtual target simulates all the interfaces of the OBA software, described in [IRS] and interact with the OBA TestBench.

3.2 VERIFYING INSTALLATION OF THE TEST BENCH

The complete OVTS needs to be installed prior to use. It need at least a Java Virtual Machine 1.3 (see Windows/Linux Help for more details on installation).

The environment variable `OBA_VTS` has to contain the name of the folder where the product is installed.

For windows NT, the following icon for the OBA Testbench is to be found on the Windows desktop :

A sub-menu [OBA Testbench] in the Windows Start/Programs menu can also generally be found.

Note : If OVTS is not installed, refer to your platform administrator (installation procedure is described in [VDD]).

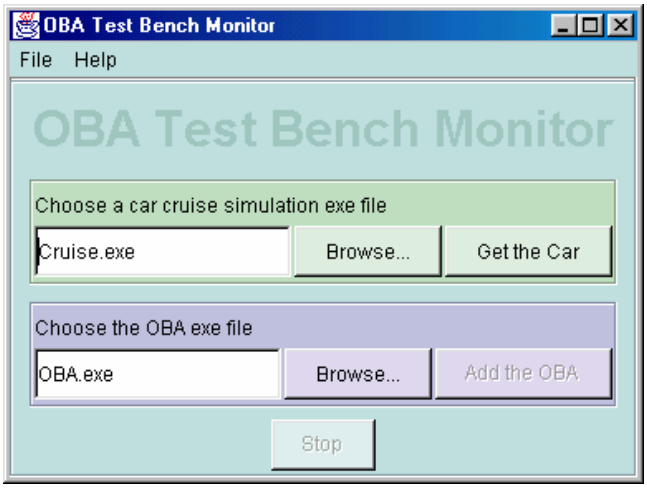
3.3 STARTING THE TEST BENCH

Starting the OBA Testbench could be done in different ways :

- For Windows OS :
  - Double-click on OBA Testbench desktop icon
  - Run the command script `%OBA_VTS%\RunOTB.bat`
- For Linux OS :
  - Set the environment variable `OBA_IDENT` with a value as `OBA1` or `OBA2` or ... and run the following command script `$OBA_VTS/RunOTB`

A console window start : execution trace messages of the testbench can appears in this windows (for testbench maintenance only).

The first window of the OBA Testbench is the testbench manager window, looking like this :



3.4 CREATING AND DESTROYING A VIRTUAL OBA-AUTOMOBILE

A testbench session means creating and destroying a virtual OBA-automobile couple running on a road.

A testbench session is composed of, in sequence :

- starting a car simulation model with its road definition file and an OBA software,
- executing sequence of driver actions on the car and on the control panel,
- stopping the car simulation model and the OBA Software.

Each testbench session will produce a result file named `Cruise_<execution number>.rec` in the OBA_VTS folder . The format of result files is explained in Appendix B.

3.4.1 CHOOSING THE AUTOMOBILE

Prior to start a testbench session, there is a needs to choose which car simulation model will be used. The car simulation model is a separate executable file whose name is given by the testbench user (a default name is given by the testbench).

This could be done by two different ways :

- typing its name in the appropriate text pane (*Choose a car cruise simulation exe file*),
- clicking on the near [Browse] button and select in the browse windows.

3.4.2 CHOOSING THE OBA SOFTWARE

Prior to start a testbench session, there is a needs to choose which OBA software will be run.

This could be done by two different ways :

- typing its name in the appropriate text pane (*Choose the OBA exe file*),
- clicking on the near [Browse] button and select in the browse window.

3.4.3 CREATING A VIRTUAL OBA-AUTOMOBILE ON A ROAD

Having selected the car simulation model file name, the car simulation executable can be started by clicking on the [Start] button near the car simulation model file name, inside the testbench manager window. This will give the driving station and the dashboard of the car accessible to the user.

Having selected the OBA Software executable name, OBA software can be started by clicking on the [Start] button near the OBA executable name, inside the testbench manager window. This will give the SD control panel accessible to the user (as if a SD sub-system have been mounted in the car)

Note : The OBA software can only be started AFTER the car simulation have been started (it make no sense trying to test an OBA software, even with the SD control panel, without any automobile !)

3.4.4 CHOOSING THE ROAD

Having selected the car simulation model file name, there is a needs to choose the definition of the road the car will runs on. The road definition is done in a road definition file (files with `.road` extension). Some examples are given in `OBA_VTS` folder. The file format for road definition is explained in Appendix A.

Choosing the road definition file could be done by :

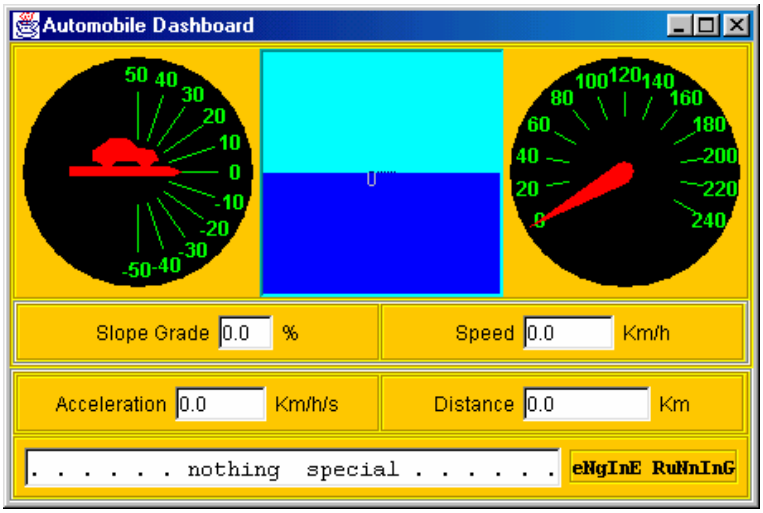
- typing its name in the appropriate text pane (*Select and choose a road*)
- clicking on the near [Browse] button and select in the browse window.

3.4.5 DESTROYING A VIRTUAL OBA-AUTOMOBILE ON A ROAD

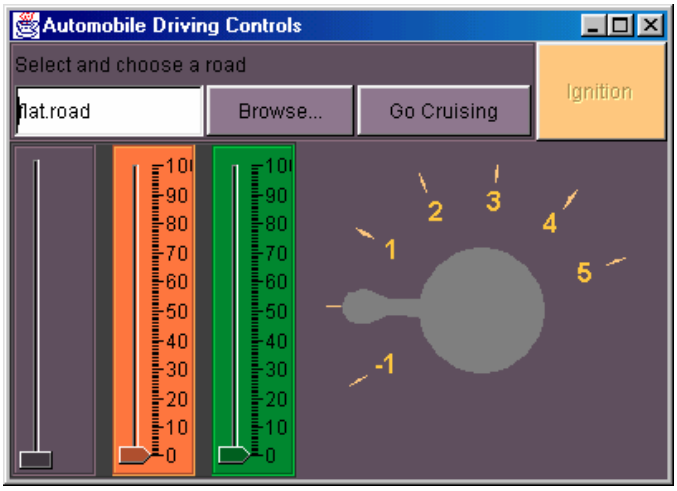
Stopping a testbench session could be done by clicking on the [Stop] button, inside the testbench monitor window. This will stop the OBA software process, stop the car simulation model process and close the result file. This could be considered like having destroyed the automobile used for the test and destroy the SD system put in the automobile (i.e. its OBA software and its control panel).

3.5 DRIVING THE VIRTUAL AUTOMOBILE

When having started a test bench session, the car dashboard windows is started, looking like this :



At the same time, the car driving station windows is started, looking like this :



Note : The car can be used without the OBA software started
(i.e. like without having a SD sub-system mounted in it).

3.5.1 DRIVING CONTROLS

3.5.1.1 Road Selection

This set of graphics elements (road name text pane, browse button and go cruising button) is the way to choose the type of road the simulated car will run on, and to start the simulation.

3.5.1.2 Ignition

This graphic element simulates the contact key of the automobile. The user can act on this element to put the engine of the car ON or OFF.

Note : This button will only be accessible when the simulation is started (see before).

3.5.1.3 Gearbox shaft

This element simulates the gearbox shaft of the automobile. The user can act on this element to change gear of the car (an automatic clutch pedal pressed and depressed is done by the testbench when changing gear: it is not mandatory for the driver to do it by itself).

3.5.1.4 Pedals

The three gliders simulates drivers pedals (clutch, brakes, acceleration). The user can act on these elements, as the driver acting on the pedals. The clutch pedal (on left) is not progressive : it can only be in open or close position.

3.5.2 CAR DASHBOARD

3.5.2.1 Engine running

This element indicate the running state of the engine :

- being stable means that the engine is NOT running.
- moving means that the engine is running,

3.5.2.2 Distance

This element indicates the mileage calculated by the testbench (supposed to be a mechanical one, that cannot be changed).

3.5.2.3 Speed

This element indicate the speed calculated by the testbench (supposed to be a mechanical one, that cannot be changed).

3.5.2.4 Slope

This element indicate the upward or downward slope of the road at the current car location.

3.5.2.5 Acceleration

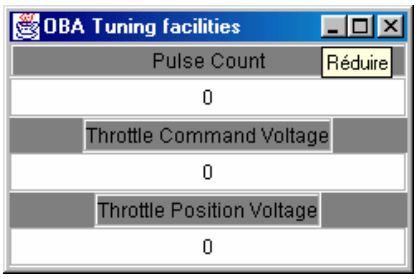
This element simulates the acceleration feeling of the driver by moving down the horizon line (accelerating) or moving up (braking). This acceleration value (calculated by the car simulation model) is shown in another field as a numeric value.

3.5.3 USING OBA

When having started an OBA software test bench session, a window that simulates the SD control panel is started. This window look like below and all the elements inside follow the semantic given in [SSS] :

3.6 TUNING FACILITIES

An additional window can be opened from the [Additional Tools] menu bar of the SD control panel. This window act as a spy on the interfaces between the OBA software and its environment and shows detailed information about these interfaces.

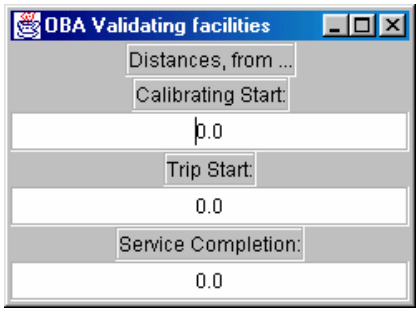


Data presented in this windows are those in the target ports :

- Pulse count (from the wheel),
- Throttle Command Voltage (as applied by OBA),
- Throttle Position Voltage (current effective value)

3.7 VALIDATING FACILITIES

An additional window can be opened from the [Additional Tools] menu bar of the SD control panel. This window show data calculated by the testbench, that are useful for validation purposes of the OBA software.



Data presented in this windows are :

- Calibration distance (i.e. the distance calculated by the testbench during a calibration sequence),
- Trip distance (i.e. the distance calculated by the testbench since beginning of the trip),
- Distance from last service completion (i.e. the distance calculated by the testbench since last service completion).

Note : All these values are calculated by the testbench (as if they were done from a mechanical equipment of the car) and may differ from OBA ones, depending on calibration values.

4. ERROR MESSAGES

Msg ID	Text	Description
NS	. . . . . nothing special	The testbench is functioning correctly. This message serves only to indicates that message panel is working.
ES	~ Engine STALLED ~~~~~~ Restart please ~	The engine of the simulated car is considered stalled but the simulation is not stopped. The car can be restarted with the [Ignition] button.
UH	~Something unexpected occured:~harmless~	An underlying error have arise in the testbench, considered harmless (the testbench is still able to perform). In some cases, it could signal that unexpected values can follow.
UE	*UNEXPECTED hard fortune: End of Cruise.	An unexpected error have arise in the testbench : the car cruise simulation is stopped and the testbench has to be restarted. This situation should not normally appears : ***Contact Technical Support.***
EB	* Engine has been BROKEN: End of Cruise.	The engine of the simulated car is considered broken : the car simulation is stopped and the testbench has to be restarted. *This situation arise when there is an excessive engine rate (i.e. accelerates hard and not changing gear for a long time).*
RF	== Cruise completed. END of the ROAD.	The road end have been reached : the road profile have been completely used by the car cruise simulation. The car cruise simulation exe is stopped.
UE	** Java/C++ Cruise Fortune Flags misfit.	An internal error have been detected in the testbench : unpredictable behavior may arise. This situation should normally never appears : ***Contact Technical Support.***

5. NOTES

5.1 DEFAUT CAR SIMULATION MODEL

The car simulation model provided by default, uses the following constant values :

- Time interval between two outputs : 50 milliseconds
- Wheel diameter : 0.60 m
- Number of Pulse per wheel revolution : 10

APPENDIX A : Road Definition File Format

Road definition files are ASCI text file that shall be written according to the following syntax :

```
Road ::= { RoadSection } + ( EndRoad )

RoadSection ::= Slope SlopeLength

EndRoad ::=   -1.0   -1.0
```

Commentaire : This will probably disappear : it seem to be obvious to catch the end-of –file !! (but maybe some strange trick stays in simu !!)

in which

Slope is a real value beyond –100.0 (vertical drop) and under 100.0 (vertical wall), representing the percentage of slope. A positive value indicates a climb and a negative one indicates a descent.

SlopeLength is an integer value corresponding to a number of meters (of the road) during which the slope is valid.

EndRoad is a special kind of *RoadSection*, with special values. This element is optional (i.e. can be omitted) and indicates explicitly the end of the road.

Example :

0.0 7000

5 2000

10.0 1000

-10 1000

-1.0 -1.0

APPENDIX B : Results File Format

Results are periodic samples of the virtual target ports and testbench data, made at the periodic rate of the OBA Testbench.

They are archived in binary files, named `Cruise_<execution number>.rec`.

These files must be converted in text file by an appropriate program, outside the OBA Testbench execution.

This conversion program is to be run, at the MS-DOS prompt, by typing the command :

```
CruiseRecordScrutiny.exe <binary_file_name.rec> <separator>
```

That will produce a `.rpt` file with the same prefix name.

The converted file is an ASCII text file, with one line per sample. Each line contains values separated by the separator above whereas the first one gives the names of each field.