OETCS/WP5/M5.1

openETCS

ITEA2 Project
Call 6 11025
2012 – 2015

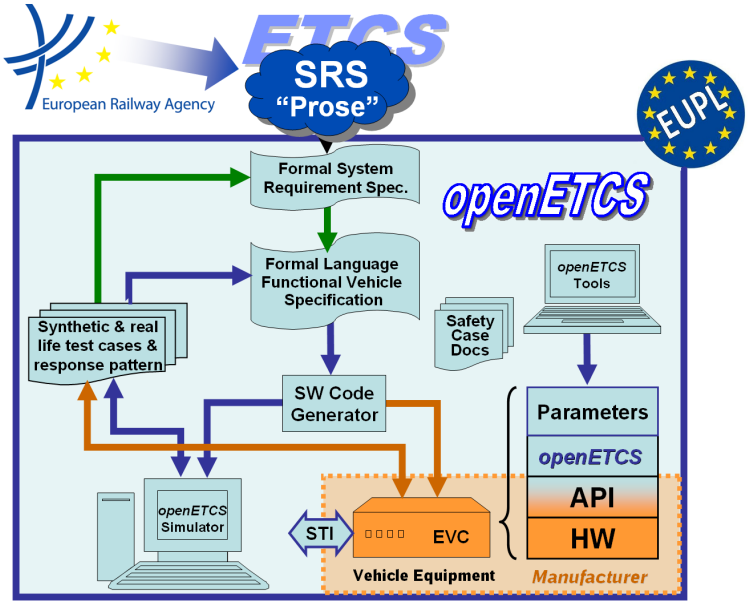
Work-Package 5: “Demonstrator”

Automatic Test Runner User manual

A comprehensive guide for writing and running SRS 330 scenarios.

Alexis Julin, Didier Weckmann, Nicolas Van Landeghem

June 2014



Funded by:

Federal Ministry of Education and Research
Région de Bruxelles-Capitale
GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, ENERGÍA Y TURISMO

This page is intentionally left blank

Work-Package 5: “Demonstrator” **OETCS/WP5/M5.1**
June 2014

Automatic Test Runner User manual

A comprehensive guide for writing and running SRS 330 scenarios.

Document approbation

Lead author:	Technical assessor:	Quality assessor:	Project lead:
location / date	location / date	location / date	location / date
signature Didier Weckmann (ERSA)	signature Didier Weckmann (ERSA)	signature Ainhoa Gracia (SQS)	signature Klaus-Rüdiger Hase (DB Netz)

Alexis Julin, Didier Weckmann, Nicolas Van Landeghem

ERSA
5 Rue Maurice Blin
67500 Haguenau, France

Description of work

Prepared for openETCS@ITEA2 Project

Abstract: This document present how the Automatic Test Runner can be used to execute Baseline 3 scenarios. This document also describe the scenario file format and its syntax.

Modification History

Version	Section	Modification / Description	Author
0.1	All parts	Creation	Christophe Menager
1.0	All parts	Official version	Christophe Menager
1.1	All parts	Add option to set SRS language for message	Eric Schellenberg
2.0	All parts	Update for new version with integrated DMI	Christophe Menager
2.1	All parts	Correction of driver action	Eric Schellenberg
2.2	All parts	Added new WAIT_ICON and BUTTON commands	Flavien Bridault
2.3	All parts	REGENBRK_ON/OFF, EDDYCURRBRK_ON/OFF, MAGNSHOEBRK_ON/OFF descriptions were erroneous and didn't reflect reality in code	Didier Weckmann
2.4	All parts	NTC_MODULE in Config_EVCInit	Flavien Bridault
2.5	All parts	Config_Scenario section	Flavien Bridault
2.6	All parts	Add tests for speed monitoring status	Stéphane Chenevoy
2.7	All parts	Add WAIT_DYNAMIC commands	Didier Weckmann
2.8	All parts	Add EXPECTED_TO_FAIL command	Didier Weckmann
2.9	All parts	Add WAIT_BUTTON ACK_XXX	Didier Weckmann
2.10	All parts	Add EVC_CONFIG, NV_FROM_HEX_BUFFER and COUNTRIES_ID	Alexis Julin
2.11	All parts	Reply to GE comments	Didier Weckmann

Table of Contents

Figures and Tables

Figures

Tables

1 Introduction

This document explains how to use the Automatic Test Runner.

The purpose of the Automatic Test Runner is to provide a tool that allows the automatic tests of the on-board simulator with a set of scenarios during developments in order to validate the changes and perform non regression tests.

The tester is a graphical application integrating an on-board and a simplified DMI. It can simulate all interfaces to the EVC (e.g. balises, radio, loops, odometer, TIU and driver interfaces). It can test the internal state of on-board in order to check its correct behaviour according to an input scenario.

In a first step, this document only concerns the simulated demonstrator. Indeed, the physical demonstrator will be available in a second step. Thus, the document will be updated accordingly.

2 How to create scenarios

2.1 Introduction

A scenario is a text file containing the description of interactions with the on-board and test conditions on the on-board internal states and/or outputs (TIU, radio message) The scenario describes how the Automatic Test Runner has to stimulate the interfaces of the on-board:

- Odometer: by simulating a train movement according to a given speed profile.
- Train interface: by simulating the train device inputs to the on-board.
- Driver: by simulating actions of the driver on a driver machine interface (DMI).
- Balise: by simulating the emission of balise contents to the on-board.
- Loop: by simulating the emission of loop contents to the on-board.
- Radio: by simulating radio communication from RBC/RIU to on-board.

According to the described interactions, a defined behaviour of the on-board is awaited and test conditions can be described in the scenario in order to test the internal state of the on-board and validate the awaited behaviour. These test conditions are defined later in the document and have different trigger: waiting a preset time, waiting a preset location, waiting a predefined output, ...

Note: The file extension for scenario files is ‘.sce’.
Note 2: The STM module is out of the openETCS project scope and will not be handled within this document.

2.2 Scenario file description

A scenario file is composed of several sections which are indicated in the file between brackets: “[<section name>]”.

Here is the list of the different section names:

SCENARIO: Main section for the scenario execution.

SpeedProfile: Description of the speed profile used for train movement simulation.

BaliseTrackside: Description of balise contents to be sent to on-board.

LoopTrackside: Description of loop messages to be sent to on-board.

Config_SRSNationalDefaults: Description of the default national values used by on-board.

Config_EVCInit: Description of starting conditions used by on-board.

Config_TrainData: Description of train data used by on-board.

Config_RBCData1: Description of the first RBC parameters used for testing.

Config_RBCData2: Description of the second RBC parameters used for testing.

Config_Scenario: Specific options for the testrunner about the current scenario.

FixedData refer to SRS section A3.1. To improve the EVC testability, it is also possible to specify other values, e.g. SAFECONNECTION_TIMEOUT or MAX_RECONNECTION_TIME can be shorten.

For data defined within SRS packets, these have to be set using the dedicated SRS message (e.g. gradient profile is defined within packet 21).

The user can also define its own sections for balise, loop and radio message contents and radio execution thread.

Comments lines can be added in the scenario file using "#". If some data of section Config_SRSNationalDefaults, Config_FixedData, Config_EVCInit or Config_TrainData are not defined, default values are used (see the corresponding chapter for more information about the default values).

2.2.1 Scenario inclusion

It is possible to include files in a scenario file when common sections are used. This can be done anywhere in the file by using the 'INCLUDE' keyword as follows:

INCLUDE=<file>

2.2.2 Main section: SCENARIO

This section is composed of a succession of commands. These commands can request different type of actions:

- DRIVER_ACTION:

Table 1. DRIVER_ACTION

Description	Simulates a driver actions interacting with the EVC (DMI or TIU interfaces). Optionally the scenario execution can be suspended a given time delay before executing next command.

Syntax	*DRIVER_ACTION = <Action>, <Delay>*	
<Action>	<Action> on the DMI:	
	Level0	ETCS level 0 entry
	Level1	ETCS level 1 entry
	Level2	ETCS level 2 entry
	Level3	ETCS level 3 entry
	DriverID	Driver identifier entry
	TrainRunningNumber	Train running number entry
	TrainData	Train data entry
	TrainData+TRN	Train data entry with train running number entry
	StartOfMission	Start of mission request
	NonLeadingModeEntry	Non leading entry request
	NonLeadingModeExit	Non leading exit request
	ShuntingModeEntry	Shunting entry request
	ShuntingModeExit	Shunting exit request
	OverrideEOA	Request to override EOA
	OverrideUnsuitability	Request to override route unsuitability
	AckBrake	Acknowledgement for brake icon
	AckMessage	Acknowledgement for message
	AckModeOrLevel	Acknowledgement for mode/Level icon
	AckTAF	Acknowledgement for track ahead free
	ConfirmIntegrity	Confirm integrity
	EnterLevel	Open level menu
	MainWindow	Return to main window
	SlipperyTrack	Select slippery track
	NonSlipperyTrack	Select non slippery track
	<Action> on the TIU:	
	MainSwitchOn	Set EVC power TIU input to ‘on’
	MainSwitchOff	Set EVC power TIU input to ‘off’
	TrainIntegrityOK	Set integrity device TIU input to ‘OK’
	TrainIntegrityNOK	Set integrity device TIU input to ‘NOK’
	OpenCabinA	Set cabin status TIU input to ‘cabin A opened’
	OpenCabinB	Set cabin status TIU input to ‘cabin B opened’

	CloseCabin	Set cabin status TIU input to ‘no cabin opened’
	EVCIsolationOn	Swith to Isolation mode
	EVCIsolationReset	v Reset Isolation mode
	EVCSleepingOn	Set Sleeping TIU input to ‘on’
	EVCSleepingOff	Set sleeping TIU input to ‘off’
	ColdMovementDetectOn	Set cold movement detector input to ‘on’
	ColdMovementDetectOff	Set cold movement detector input to ‘off’
	DirectionNominal	Set direction controller TIU input to ’Nominal’ (forward)
	DirectionReverse	Set direction controller TIU input to ’Reverse’ (backward)
	DirectionStandstill	Set direction controller TIU input to ’Standstill’ (neutral)
	DirectionUndefined	Set direction controller TIU input to ’Undefined’
	SBOn	Set service brake TIU input to ’on’
	SBOff	Set service brake TIU input to ’off’
	EBOn	Set emergency brake TIU input to ’on’
	EBOff	Set emergency brake TIU input to ’off’
	SBOutOfOrder	Set service brake out of order
	EBOutOfOrder	Set emergency brake out of order
	MCBOpen	Set main circuit breaker TIU input to ’open’
	MCBClose	Set main circuit breaker TIU input to ’close’
	PantographDown	Set pantograph TIU input to ’down’
	PantographUp	Set pantograph TIU input to ’up’
	PassengerEBOff	Set passenger emergency brake TIU input to ’off’
	PassengerEBOn	Set passenger emergency brake TIU input to ’on’
<Delay>	Time delay in seconds to wait before executing next command (optional)	
Example	*DRIVER_ACTION = DriverID, 1*	

- MOVE_TRAIN:

Table 2. MOVE_TRAIN

Description	Simulates train movement in forward direction according to speed profile.
Syntax	*MOVE_TRAIN*

- MOVE_TRAIN_BACK:

Table 3. MOVE_TRAIN_BACK

Description	Simulates train movement in backward direction according to speed profile.
Syntax	*MOVE_TRAIN_BACK*

- WAIT_TIME:

Table 4. WAIT_TIME

Description	Waits the given time delay before executing next command.
Syntax	*WAIT_TIME = <Delay>*
<Delay>	Time delay in seconds to wait before executing next command
Example	*WAIT_TIME = 1*

- WAIT_SPEED:

Table 5. WAIT_SPEED

Description	Waits the given speed to be reached by the train movement simulation before executing next command.
Syntax	*WAIT_SPEED = <Speed>*
<Speed>	Speed in km/h to be reached before executing next command
Example	*WAIT_SPEED = 101*

- WAIT_LOCATION:

Table 6. WAIT_LOCATION

Description	Waits the given location to be reached by the train movement simulation before executing next command.
Syntax	*WAIT_LOCATION = <Location>*
<Location>	Location in meters to be reached before executing next command
Example	*WAIT_LOCATION = 100*

- WAIT_STANDSTILL:

Table 7. WAIT_STANDSTILL

Description	Waits the train movement simulation to reach standstill before executing next command.
Syntax	*WAIT_STANDSTILL*

- WAIT_STATUS:

Table 8. WAIT_STATUS

Description	Waits the EVC to reach given internal status before executing next command. Several conditions can be requested at the same time. Optionally the scenario execution can be stopped with failure if internal status is not reached in the given time delay.	
Syntax	*WAIT_STATUS = <Condition 1>, ..., <Condition n>, <Delay>, <FATAL>*	
<Condition>	*EB_ON*	Emergency brake intervention request TIU status is 'on'
	EB_OFF	Emergency brake intervention request TIU output is 'off'
	SB_ON	Service brake intervention request TIU output is 'on'
	SB_OFF	Service brake intervention request TIU output is 'off'
	CUTOFF_ON	Cut off traction intervention request TIU output is 'on'
	CUTOFF_OFF	Cut off traction intervention request TIU output is 'off'
	MCB_OPEN	Main circuit breaker request TIU output is 'open'
	MCB_CLOSE	Main circuit breaker request TIU output is 'closed'
	PANTOGRAPH_LOW	pantograph request TIU output is 'low'
	PANTOGRAPH_UP	pantograph request TIU output is 'up'
	AIRTIGHT_ON	Airtight request TIU output is 'on'
	AIRTIGHT_OFF	Airtight request TIU output is 'off'
	PEB_INHIBIT	Passenger emergency brake TIU output is 'inhibited'
	PEB_PERMIT	Passenger emergency brake TIU output is 'permitted'
	REGENBRK_ON	Regenerative brake TIU output is 'permitted'
	REGENBRK_OFF	Regenerative brake TIU output is 'inhibited'
	EDDYCURRBRK_ON	Eddy current brake TIU output is 'permitted'
	EDDYCURRBRK_OFF	Eddy current brake TIU output is 'inhibited'
	MAGNSHOEBRK_ON	Magnetic shoe brake TIU output is 'permitted'
	MAGNSHOEBRK_OFF	Magnetic shoe brake TIU output is 'inhibited'
	LEVEL_0	EVC ETCS is level 0
	LEVEL_1	EVC ETCS is level 1
	LEVEL_2	EVC ETCS is level 2

	LEVEL_3	EVC ETCS is level 3
	RADIOSAFE_ON	Radio safe connection is established
	RADIOSAFE_OFF	Radio safe connection is not established
	RADIOCONN_ON	Radio session is established
	RADIOCONN_OFF	Radio session is not established
	MODE_FS	EVC mode is Full Supervision
	MODE_OS	EVC mode is On Sight
	MODE_SR	EVC mode is Staff Responsible
	MODE_SH	EVC mode is Shunting
	MODE_UN	EVC mode is Unfitted
	MODE_SL	EVC mode is Sleeping
	MODE_SB	EVC mode is Standby
	MODE_TR	EVC mode is Trip
	MODE_PT	EVC mode is Post Trip
	MODE_SF	EVC mode is System Failure
	MODE_IS	EVC mode is Isolation
	MODE_NP	EVC mode is No Power
	MODE_NL	EVC mode Non Leading
	MODE_SE	EVC mode is STM European
	MODE_SN	EVC mode is STM National
	MODE_RV	EVC mode is Reversing
	MONITORING_CSM	Train in ceiling speed monitoring
	MONITORING_PIM	Train in pre-indication monitoring
	MONITORING_TSM	Train in target speed monitoring
	MONITORING_RSM	Train in release speed monitoring
<Delay>	Time delay for condition to be reached (optional)	
<FATAL>	if 'FATAL' keyword is set, the scenario is stopped with FAILURE status if condition is not reached in the given time delay. (optional)	
Example	*WAIT_STATUS = MODE_FS, EB_OFF, LEVEL_1, 5, FATAL*	

- CHECK_PARAM:

Table 9. CHECK_PARAM

Description	Checks an internal EVC value. Optionally the scenario execution can be stopped with a failure return if the condition on the checked parameter is not verified.
Syntax	*CHECK_PARAM = <Condition>, FATAL*
<Condition>	The condition is given with the following syntax: <Parameter> <comparison> <value>:

	EOA_SPEED	End of authority speed
	EOA_LOCATION	End of authority location
	EB_SPEED	Emergency brake intervention speed
	SB_SPEED	Service brake intervention speed
	PERM_SPEED	Permitted speed
	WARN_SPEED	Warning speed
	TARGET_SPEED	Target speed
	RELEASE_SPEED	Release speed
	MRSP	Most restrictive speed
	TARGET_LOCATION	Target location
	ADHESION	Adhesion factor value
	ESTIM_FRONT_LOCATION	Estimated front train position
	The <value> is compared to the internal value according to the comparison symbol '>', '<' or '=', '>=, '<='	
<FATAL>	if 'FATAL' keyword is set, the scenario is stopped with FAILURE status if condition is not reached in the given time delay. (optional)	
Example	*CHECK_PARAM = EOA_LOCATION > 495, FATAL*	

- SET:

Table 10. SET

Description	Modify RBC connection status.	
Syntax	*SET = <Status>*	
<Status>	*RBC_SAFE_OFF*	Shutdown safe connection of radio module 1
	RBC_SAFE_ON	Enable safe connection of radio module 1
	RBC_NET_EMPTY_LIST	radio module 1 returns an empty network list
	RBC_NET_FAIL_REGISTRATION	radio module 1 refuses network registration
	RBC_NET_RESET	radio module 1 accepts network requests normally
	RBC2_SAFE_OFF	Shutdown safe connection of radio module 2
	RBC2_SAFE_ON	Enable safe connection of radio module 2
	RBC2_NET_EMPTY_LIST	radio module 2 returns an empty network list
	RBC2_NET_FAIL_REGISTRATION	radio module 2 refuses network registration
	RBC_NET_RESET	radio module 2 accepts network requests normally
Example	*SET = RBC_NET_RESET*	

- DO_RADIO:

Table 11. DO_RADIO

Description	Executes a user defined section in parallel to the main section. This is useful for radio management that has to be performed in parallel to actions of the main section.
Syntax	*DO_RADIO = <SectionName>*
<SectionName>	It is a user defined section that has to be executed
Example	*DO_RADIO = OnBoardInitSession*

- RBC_RADIO:

  Sends a RBC radio message defined in the given section.

Table 12. RBC_RADIO

Description	Sends a RBC radio message defined in the given section.
Syntax	*RBC_RADIO = <SectionName>*
<SectionName>	It is a user defined section that contains the radio message description that has to be sent to EVC.
Example	*RBC_RADIO = RBCConfiguration*

- WAIT_RADIO_SENT:

Table 13. WAIT_RADIO_SENT

Description	Waits the EVC to send a radio message corresponding to the one described in the given used defined section in the given time delay before executing the next command. Optionally the scenario execution can be stopped with a failure return if the radio message is not sent in the given time delay.
Syntax	*WAIT_RADIO_SENT = <SectionName>, <Delay>, FATAL*
<SectionName>	It is a user defined section that contains the radio message description that is awaited to be sent by EVC.
<Delay>	Time delay in seconds for waiting EVC to send radio message
<FATAL>	If 'FATAL' keyword is set, the scenario is stopped with FAILURE status if radio message is not sent within time given delay. (optional)
Example	*WAIT_RADIO_SENT = RIM_InitCommSession, 10, FATAL*

- CONNECT_RADIO:

Table 14. CONNECT_RADIO

Description	Simulate a connection initiated by RBC for radio module 1.
Syntax	*CONNECT_RADIO*

- CONNECT_RADIO2:

Table 15. CONNECT_RADIO2

Description	Simulate a connection initiated by RBC for radio module 2.
Syntax	*CONNECT_RADIO2*

- WAIT_SYMBOL:

Table 16. WAIT_SYMBOL

Description	Waits the EVC to ask the DMI to display a given icon in a specific area.
Syntax	*WAIT_SYMBOL = <SymbolNumber>, <SymbolArea>, ACK, <Delay>, FATAL*
<SymbolNumber>	Number of the symbol such as found in ERA_ERTMS document. You can also specify NONE if you want to test that nothing is displayed in a specific area.
<SymbolNumber>	Area on the DMI where the symbol is expected to be displayed.
<ACK>	Optional, if present we test that this symbol requires the driver acknowledgement.
<Delay>	Time delay in seconds for waiting EVC to ask display of the symbol.
<FATAL>	If 'FATAL' keyword is set, the scenario is stopped with FAILURE status if the button is not displayed within time given delay. (optional)
Example	*WAIT_SYMBOL = ST02, A4, 2, FATAL*

- WAIT_BUTTON:

Table 17. WAIT_BUTTON

Description	Waits the EVC to request the DMI the availability of one or several menu buttons.
Syntax	*WAIT_BUTTON = <Button1>, <Button2>, ..., <ButtonN>, <Delay>, FATAL*
<Button>	Name of the button to be tested. An exclamation mark can be added as a prefix to the button name in order test that the button is currently not displayed on the DMI. The name can be one of those:
	SOM
	SHUNTING
	EXIT_SHUNTING
	NON_LEADING
	MAINTAIN_SHUNTING
	DRIVER_ID
	TRAIN_RUN_NB
	ETCS_LEVEL
	TRAIN_DATA
	TRAIN_DATA_VIEW
	SR_DATA
	LANGUAGE_SELECT
	OVERRIDE_EOA

	ADHESION_FACTOR
	SYSTEM_VERSION
	SOUND
	BRIGHTNESS
	CONFIRM_INTEG
	ISOLATION
	DMI_WAIT
	USE_SHORT_NUMBER
	ENTER_RBC_DATA
	ENTER_RADIO_ID
	CONTACT_LAST_RBC
	TRAIN_DATA_SWITCH
	FIXED_DATA_ENTRY
	VBC_SET
	VBC_REMOVE
	ACK_MODE
	ACK_MESSAGE
	ACK_BRAKE
<Delay>	Time delay in seconds for waiting EVC to ask display of the button.
<FATAL>	If 'FATAL' keyword is set, the scenario is stopped with FAILURE status if the button(s) availability do(es) not match within time given delay. (optional)
Example	*WAIT_BUTTON = ETCS_LEVEL, DRIVER_ID, !SOM, 1, FATAL*

- WAIT_DYNAMIC:

Table 18. WAIT_DYNAMIC

Description	Waits the EVC to send to the DMI Dynamic data.
Syntax	*WAIT_DYNAMIC = DMI_VARIABLE [=|<|>] z, ..., <Delay>, FATAL*
<DMI_VARIABLE>	Name of the DMI variable to test. The name can be one of those:
	DMI_T_CLOCK
	DMI_V_TRAIN
	DMI_X_VTRAIN_DIGITS
	DMI_O_TRAIN
	DMI_O_BRAKETARGET
	DMI_X_OBRAKETARGET_DIGITS
	DMI_V_TARGET
	DMI_V_PERMITTED
	DMI_V_RELEASE

	DMI_O_BCSP
	DMI_V_INTERVENTION
	DMI_M_MODE
	DMI_M_LEVEL
	DMI_NID_STM
	DMI_NID_C
	DMI_NID_C_UNKNOWN
	DMI_M_WARNING
	DMI_M_SUPSTATUS
	DMI_O_LOA
	DMI_V_LOA
	DMI_O_KP_BALISE_TRACK_KILOMETER
	DMI_O_KP_DIST_TO_BALISE
	DMI_M_KP_FLAG
	DMI_O_DIST_TO_TSA
<Delay>	Time delay in seconds for waiting EVC to the asked value to the dmi.
<FATAL>	If 'FATAL' keyword is set, the scenario is stopped with FAILURE.
Example	*WAIT_BUTTON = ETCS_LEVEL, DRIVER_ID, !SOM, 1, FATAL*

2.2.3 Speed profile section: SpeedProfile

This section describes how the train movement simulation has to behave. The train movement simulation is controlled in the main section with the 'MOVE_TRAIN' and 'MOVE_TRAIN_BACK' commands. The train will start moving and will follow the given speed profile until standstill.

A speed profile is a succession of positions associated to a speed to reach at this location (<location(m)> = <speed(km/h)>). The locations of the speed profile correspond to the traveled distance of the train and not its absolute position. That's why they are always incrementing. It is the used command in the main section that will determine in which direction the train will move. A constant acceleration/deceleration is used between 2 points of the profile. The following example shows a speed profile with its associated main section and a graphic representing the obtained train movement:

Note: The on-board interventions are not taken into account in the train movement simulation.

2.2.4 Balise section: BaliseTrackside

This section describes the locations at which balise messages have to be sent to the EVC (<location(m)> = <user balise message name>). A user defined section describing the balise message content is associated to each location. As for the speed profile, the locations are given for the travelled distance.

The user defined sections contain the description of the balise message content. It uses the variable names as defined in SRS chapter 7 and 8 (See Erreur : source de la référence non

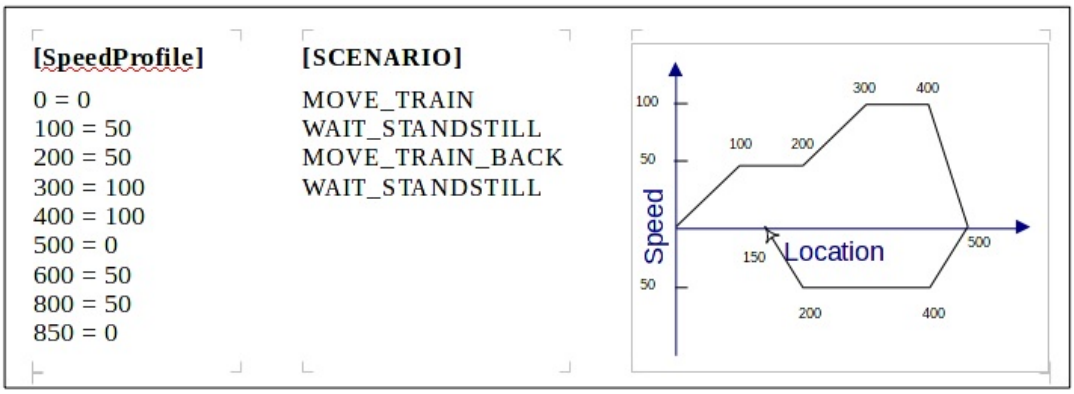


Figure 1. Speed profile

trouvée) Message variables don't need to be defined if the default value is used but they have to be in the right order.

Note: The SRS language can be indicated in option with SRS=<version> Example: <location(m)> = <user balise message name>, SRS=2.3.0

During the execution, balise message contents are automatically sent to EVC when the location is reached.

The following example shows a balise trackside and the associated balise message content sections:

[BaliseTrackside]	**[BG_1_0]**		**[BG_1_1]**	
100 = BG_1_0	*# Header*		*# Header*	
105 = BG_1_1	N_PIG	= 0	N_PIG	= 1
	N_TOTAL	= 1	N_TOTAL	= 1
	NID_C	= 0	NID_C	= 0
	NID_BG	= 1	NID_BG	= 1
	Q_LINK	= 1	Q_LINK	= 1
	# L1 MA packet		*# SSP packet*	
	NID_PACKET	= 12	NID_PACKET	= 27
	Q_DIR	= 1	Q_DIR	= 2
	Q_SCALE	= 1	V_STATIC	= 2
	V_MAIN	= 30		
	L_ENDSECTION	= 500	*# Gradient packet*	
			NID_PACKET	= 21

2.2.5 Loop section: LoopTrackside

This section describes the locations at which loop messages have to be sent to the on-board (<location(m)> = <user loop message name>). A user defined section describing the loop

message content is associated to each location. As for the speed profile, the locations are given for the travelled distance.

The user defined sections contain the description of the loop message content. It uses the variable names as defined in SRS chapter 7 and 8 (See Erreur : source de la référence non trouvée for more details). Message variables don't need to be defined if the default value is used but they have to be in the right order.

Note:

- the SRS language can be indicated in option with SRS=<version>;
- the spread spectrum code value can be indicated in option with SSCode=<SSCode>, if not indicated default value 15 is used.
- Example: <location(m)> = <user loop message name>, SSCode=3, SRS=2.3.0

During the execution, loop message contents are automatically sent to on-board when the location is reached. The following example shows a loop trackside and the associated loop message content sections:

[LoopTrackside]	**[Loop0]**		**[Loop1]**	
100 = Loop0	*# Header*		*# Header*	
105 = Loop1	NID_C	= 0	NID_C	= 0
	NID_LOOP	= 1	NID_LOOP	= 1
	# Infill location reference		*# Infill location reference*	
	NID_PACKET	= 136	NID_PACKET	= 136
	Q_DIR	= 2	Q_DIR	= 2
	NID_BG	= 1	NID_BG	= 1
	# L1 MA packet		*# L1 MA packet*	
	NID_PACKET	= 12	NID_PACKET	= 12
	Q_DIR	= 2	Q_DIR	= 2
	Q_SCALE	= 1	Q_SCALE	= 1
	V_MAIN	= 0	V_MAIN	= 30
	L_ENDSECTION	= 0	L_ENDSECTION	= 500
	# SSP packet		*# SSP packet*	
	NID_PACKET	= 27	NID_PACKET	= 27
	Q_DIR	= 2	Q_DIR	= 2
	V_STATIC	= 30	V_STATIC	= 30
	# Gradient packet		*# Gradient packet*	
	NID_PACKET	= 21	NID_PACKET	= 21

2.2.6 Default national values configuration section: Config_SRSNationalDefaults

This section defines the default national values used by EVC, before Start-of-Mission. These default national values can be overridden when the train receives a packet 3 from trackside. This will allow the Test Environment to be able to test the correct update of the national values data.

- COUNTRY_ID:

Table 19. COUNTRY_ID

Description	Country identifier of the country for which default national values are valid.
Default value	0

- COUNTRIES_ID:

Table 20. COUNTRIES_ID

Description	A list of country identifiers for which default national values are valid.

Default value	0
Example	COUNTRIES_ID = 253;254

- DRIVER_ADHESION:

Table 21. DRIVER_ADHESION

Description	Qualifier for the modification of trackside adhesion factor by driver.
SRS Name	*Q_NVDRIVER_ADHES*
Special values	– 0: not allowed – 1: allowed
Default value	0

- SH_SPEED:

Table 22. SH_SPEED

Description	Shunting mode permitted speed (km/h).
SRS Name	*V_NVSHUNT*
Range	0 km/h – 600 km/h (in 5 km/h step)
Default value	30 km/h

- SR_SPEED:

Table 23. SR_SPEED

Description	Staff Responsible mode permitted speed (km/h).
SRS Name	*V_NVSTFF*
Range	0 km/h – 600 km/h (in 5 km/h step)
Default value	40 km/h

- OS_SPEED:

Table 24. OS_SPEED

Description	On Sight mode permitted speed (km/h).
SRS Name	*V_NVONSIGHT*
Range	0 km/h – 600 km/h (in 5 km/h step)
Default value	30 km/h

- UN_SPEED:

Table 25. UN_SPEED

Description	Unfitted mode permitted speed (km/h).
SRS Name	*V_NVUNFIT*
Range	0 km/h – 600 km/h (in 5 km/h step)

Default value	100 km/h

- RELEASE_SPEED:

Table 26. RELEASE_SPEED

Description	Release Speed permitted speed (km/h).
SRS Name	*V_NVREL*
Range	0 km/h – 600 km/h (in 5 km/h step)
Default value	40 km/h

- ROLLAWAY_DISTANCE:

Table 27. ROLLAWAY_DISTANCE

Description	Distance limit used for roll away and reverse movement protection (meter).
SRS Name	*D_NVROLL*
Range	0 meters – 327 660 meters (in 1 meter step)
Special values	INFINITY: deactivates roll away and reverse movement protection
Default value	2 meters

- SB_USETOTARGET:

Table 28. SB_USETOTARGET

Description	Permission to use service brake when braking to a target is supervised.
SRS Name	*Q_NVSRBKTRG*
Special values	– 0: no – 1: yes
Default value	yes

- EB_RUNRELEASE:

Table 29. EB_RUNRELEASE

Description	Permission to release the emergency brake immediately if the condition why the system has triggered the emergency brake (speed exceeds emergency brake intervention limit, lack of driver reaction) is not fulfilled any more.
SRS Name	*Q_NVEMRRLS*
Special values	– 0: only at standstill – 1: immediate release possible
Default value	only at standstill

- OVERRIDEEOA_ENTRYSPEED:

Table 30. OVERRIDEEOA_ENTRYSPEED

Description	Maximum speed limit allowing the driver to select the ”override EOA” function (km/h).
SRS Name	*V_NVALLOWOVTRP*
Range	0 km/h – 600 km/h (in 5 km/h step)
Default value	0 km/h

- OVERRIDEEOA_MAXSPEED:

Table 31. OVERRIDEEOA_MAXSPEED

Description	Permitted speed limit to be supervised when the ”override EOA” function is active (km/h).
SRS Name	*V_NVSUPOVTRP*
Range	0 km/h – 600 km/h (in 5 km/h step)
Default value	30 km/h

- OVERRIDEEOA_MAXDISTANCE:

Table 32. OVERRIDEEOA_MAXDISTANCE

Description	Maximum distance for overriding the train trip (meter).
SRS Name	*D_NVOVTRP*
Range	0 meters – 327 670 meters (in 1 meter step)
Default value	200 meters

- OVERRIDEEOA_MAXTIME:

Table 33. OVERRIDEEOA_MAXTIME

Description	Maximum time for overriding the train trip (second).
Range	0 second – 255 seconds (in 1 second step)
Default value	60 seconds

- DRIVERID_RUNCHANGE:

Table 34. DRIVERID_RUNCHANGE

Description	Entry of Driver ID permitted while running.
SRS Name	*M_NVDERUN*
Special values	– 0: no – 1: yes
Default value	yes

- PT_MAXDISTANCE:

Table 35. PT_MAXDISTANCE

Description	Maximum distance for reversing in Post Trip mode (meter).
SRS Name	*D_NVPOTRP*
Range	0 meters – 327 670 meters (in 1 meter step)
Default value	200 meters

- CONTACT_TIME:

Table 36. CONTACT_TIME

Description	Maximal time without new ”safe” radio message (second).
SRS Name	*T_NVCONTACT*
Range	0 second – 254 seconds (in 1 second step)
Special values	INFINITY: deactivates supervision of radio link
Default value	∞ seconds

- SR_MAXDISTANCE:

Table 37. SR_MAXDISTANCE

Description	Maximum distance for running in Staff Responsible mode (meter).
SRS Name	*D_NVSTFF*
Range	0 meters – 327 660 meters (in 1 meter step)
Special values	INFINITY: deactivates distance supervision in Staff Responsible mode
Default value	∞ meters

- NOCONTACT_REACTION:

Table 38. NOCONTACT_REACTION

Description	Indicates the reaction to be performed when T_NVCONTACT timer elapses.
SRS Name	*M_NVCONTACT*
Special values	– NONE: no reaction – TRIP: train trip – SB: service brake application
Default value	no reaction

- NV_FROM_HEX_BUFFER:

Table 39. NV_FROM_HEX_BUFFER

Description	Allows to initialize EVC with a general radio message (24) containing packet 3.
Format	A list of hexadecimal values splited by comma that represents general radio message

Example	NV_FROM_HEX_BUFFER = 18, 08, 80, 00, 00, 00, 00, 00, 00, 00, 68, 31, 10, 00, 02, 3f, 4f, f0, c1, 83, 20, 0a, 00, 28, 06, 18, 03, 27, f8, 0c, 84, 69, ff, fd
Default value	none
Warning	If NV_FROM_HEX_BUFFER is defined all other values define in this section are removed

2.2.7 EVC configuration section: Config_EVCInit

This section allow definition of values of the EVC at power on, i.e. at initialization state.

- LINE_LEVEL:

Table 40. LINE_LEVEL

Description	Default ETCS level.
Special values	– 0: level 0 – 1: level 1 – 2: level 2 – 3: level 3
Default value	level 3

- RBC_ID:

Table 41. RBC_ID

Description	Default RBC identifier.
Range	$$RBC\_ID = NID\_C * 2^{14} + NID\_RBC$$ – NID_C: 0 – 1024 – NID_RBC: 0 – 16384
Default value	789

- RBC_PHONE:

Table 42. RBC_PHONE

Description	Default RBC phone number.
Range	1 to 16 digits value
Default value	1234123412341234

- NETWORK_ID:

Table 43. NETWORK_ID

Description	Default radio network identifier.

Range	1 to 6 digits value
Default value	123456

- COUNTRY_ID:

Table 44. COUNTRY_ID

Description	Country identifier of the last relevant balise group (LRBG).
Range	0 – 1024
Default value	0

- GROUP_ID:

Table 45. GROUP_ID

Description	Identifier of the last relevant balise group (LRBG).
Range	0 – 16384
Default value	4522

- DISTANCE:

Table 46. DISTANCE

Description	Distance to the reference last relevant balise group from train front.
Range	0 meter – 327 670 meters
Default value	50

- DIRECTION:

Table 47. DIRECTION

Description	Validity direction for the reference balise group (LRBG).
Special values	– NOMINAL – REVERSE – UNDEFINED
Default value	NOMINAL

- VALIDITY:

Table 48. VALIDITY

Description	Validity of the last relevant balise group (LRBG).
Special values	– VALID – INVALID – UNKNOWN
Default value	VALID

- NTC_MODULE:

Table 49. NTC_MODULE

Description	Name of the NTC module to instantiate (only one currently).
Special values	– GENERIC – GENERICSS – TEST
Default value	GENERIC

- EVC_CONFIG:

Table 50. EVC_CONFIG

Description	Allows to activate EVC configuration.
Values	– CFG_RADIO_INTERNAL_TIME_STAMP : Internal time stamping (not use T_TRAIN) – CFG_USE_JRU : Use JRU (generates JRU data file) – CFG_RECORD_TO_CSV_FILE : Record supervision data to CSV file – CFG_BAL_WITH_ODO_STAMP : Balise are received with odo stamp – CFG_LOOP_WITH_SSCODE : Loop are received with spread spectrum code – CFG_LOCAL_TIME_STAMP : Request local time stamp otherwise GMT time in log & JRU record – CFG_RECORDER_LOG_ADD_FULL_TIME_STAMP : add time stamp in EuroCabLog.dat like 2009-05-29/08:15:21.29
Example	EVC_CONFIG = CFG_LOOP_WITH_SSCODE|CFG_USE_JRU
Warning	CFG_RADIO_INTERNAL_TIME_STAMP and CFG_BAL_WITH_ODO_STAMP are mandatories for EVC then EVC set these configurations automatically and they can be removed

2.2.8 Train data configuration section : Config_TrainData

This section defines a default train data set but also allows test automation by simulating user actions (e.g. from DMI).

- BALISE_COM_AVAILABLE:

Table 51. BALISE_COM_AVAILABLE

Description	Indicates if balise communication is available on-board.
Special values	– 0: not available – 1: available

Default value	available

- LOOP_COM_AVAILABLE:

Table 52. LOOP_COM_AVAILABLE

Description	Indicates if loop communication is available onboard.
Special values	– 0: not available – 1: available
Default value	available

- RADIO_COM_AVAILABLE:

Table 53. RADIO_COM_AVAILABLE

Description	Indicates the number of radio equipments available onboard.
Range	0 – 2
Default value	1

- INTEGRITY_DEVICE_AVAILABLE:

Table 54. INTEGRITY_DEVICE_AVAILABLE

Description	Indicates if integrity detection device is available.
Special values	– 0: not available – 1: available
Default value	available

- SERVICE_BRAKE_AVAILABLE:

Table 55. SERVICE_BRAKE_AVAILABLE

Description	Indicates if service brakes are available.
Special values	– 0: not available – 1: available
Default value	available

- ETCS_PHONE1:

Table 56. ETCS_PHONE1

Description	Phone number of first radio equipment.
Range	1 to 16 digits value

- ETCS_PHONE2:

Table 57. ETCS_PHONE2

Description	Phone number of second radio equipment.
Range	1 to 16 digits value

- TCO_AVAILABLE:

Table 58. TCO_AVAILABLE

Description	Indicates if traction cut off is available.
Special values	– 0: not available – 1: available
Default value	available

- USE_BRK_FEEDBACK:

Table 59. USE_BRK_FEEDBACK

Description	Indicate if brake feedback is available.
Special values	– 0: not available – 1: available
Default value	not available

- BRK_PERCENTAGE:

Table 60. BRK_PERCENTAGE

Description	Brake percentage for train data conversion model.
Range	Integer value
Default value	135

- ETCS_ID:

Table 61. ETCS_ID

Description	ETCS identifier of the on-board.
Range	0 – 16777215
Default value	4554

- BALISEANTENNA_OFFSET:

Table 62. BALISEANTENNA_OFFSET

Description	Offset of balise antenna relative to train front.
Range	Integer value (in meters)
Default value	0 meter

- TRAIN_CATEGORY:

Table 63. TRAIN_CATEGORY

Description	Train category of the train.
Range	Integer value. See (NC_TRAIN)
Default value	1

- CUTOFF_TIME:

Table 64. CUTOFF_TIME

Description	Traction cut off time.
Range	Double value (in seconds)
Default value	1.0 second

- SPEED_MAX:

Table 65. SPEED_MAX

Description	Maximum train speed.
Range	Double value (in km/h)
Default value	300 km/h (83.3 m/s)

- TRAIN_LENGTH:

Table 66. TRAIN_LENGTH

Description	Train length.
Range	Double value (in meters)
Default value	100 meters

- TRAIN_MAXACCEL:

Table 67. TRAIN_MAXACCEL

Description	Train maximum acceleration.
Range	Double value (in m/s^2)
Default value	0.5 m/s^2

- LOADING_GAUGE_MASK:

Table 68. LOADING_GAUGE_MASK

Description	Loading gauge type of the train.
Range	Integer value. See (M_LOADINGGAUGE)
Default value	1

- AXLE_LOAD:

Table 69. AXLE_LOAD

Description	Axle load of the train.
Range	Double value (in kg)
Default value	23 000 kg

- ODO_FIXED_ERROR:

Table 70. ODO_FIXED_ERROR

Description	Fixed error on odometric data.
Range	Double value (in meter)
Default value	5 meters

- TRACTION_POWERS:

Table 71. TRACTION_POWERS

Description	List of traction power types equipped by the train.
Range	Space separated integer value list. See (M_TRACTION)
Default value	11 48

- COLD_MOVEMENT_DETECTOR_AVAILABLE:

Table 72. COLD_MOVEMENT_DETECTOR_AVAILABLE

Description	Indicate if cold movement detector is available.
Special values	– 0: not available – 1: available
Default value	not available

2.2.9 Train data configuration section: Config_FixedData

- SAFECONNECTION_TIMEOUT:

Table 73. SAFECONNECTION_TIMEOUT

Description	Safe connection repeat timeout.
Range	Double value (in seconds)
Default value	20 seconds

- MAX_RECONNECTION_TIME:

Table 74. MAX_RECONNECTION_TIME

Description	Maximum time to maintain a communication session in case of failed re-connection attempts.
Range	Double value (in seconds)

Default value	300 seconds

2.2.10 Train data configuration section: Config_RBCData

- RBC_OFF_DISCONNECT_TIMEOUT:

Table 75. RBC_OFF_DISCONNECT_TIMEOUT

Description	When the RBC is off, (after calling SET=RBC_SAFE_OFF) and a safe connection is requested, it replies by a disconnection request. When we need to test the absence of a reply, we may want to delay this behaviour.
Range	Double value (in seconds)
Default value	0.5 seconds

2.2.11 Scenario configuration section: Config_Scenario

- DMI_SIMPLIFIED_SUPPORTED:

Table 76. DMI_SIMPLIFIED_SUPPORTED

Description	Tells if this scenario can be executed when the simplified DMI is used.
Range	0 or 1
Default value	1 (True)

- EXPECTED_TO_FAIL:

Table 77. EXPECTED_TO_FAIL

Description	Tells if this scenario will fail because the fix is not yet done and should not be counted as a regression.
Range	Not available
Default value	Not available

3 How to run the Automatic Test Runner

First of all, to run the Automatic Test Runner (preliminary Test Environment), a terminal has to be opened and the current directory changed to the Automatic Test Runner working directory:

```
>cd /usr/local/openETCS/test_runner
```

3.1 Single scenario execution

It is possible to run one single scenario by calling the application directly with the scenario as argument in the Automatic Test Runner working directory:

```
>./test_runner <ScenarioFileName>
```

3.2 Parameters

Parameters can be displayed by launching the software without any of them:

```
>./test_runner
Usage: test_runner <FILE> [OPTIONS]
Options:

  -m : manual mode, disable DMI driver action (automated clicks)

  -j : enable JRU recording

  -l : generate logs when scenario fail

  -a : autoclose application at the end of scenario
```

3.3 Graphical user interface

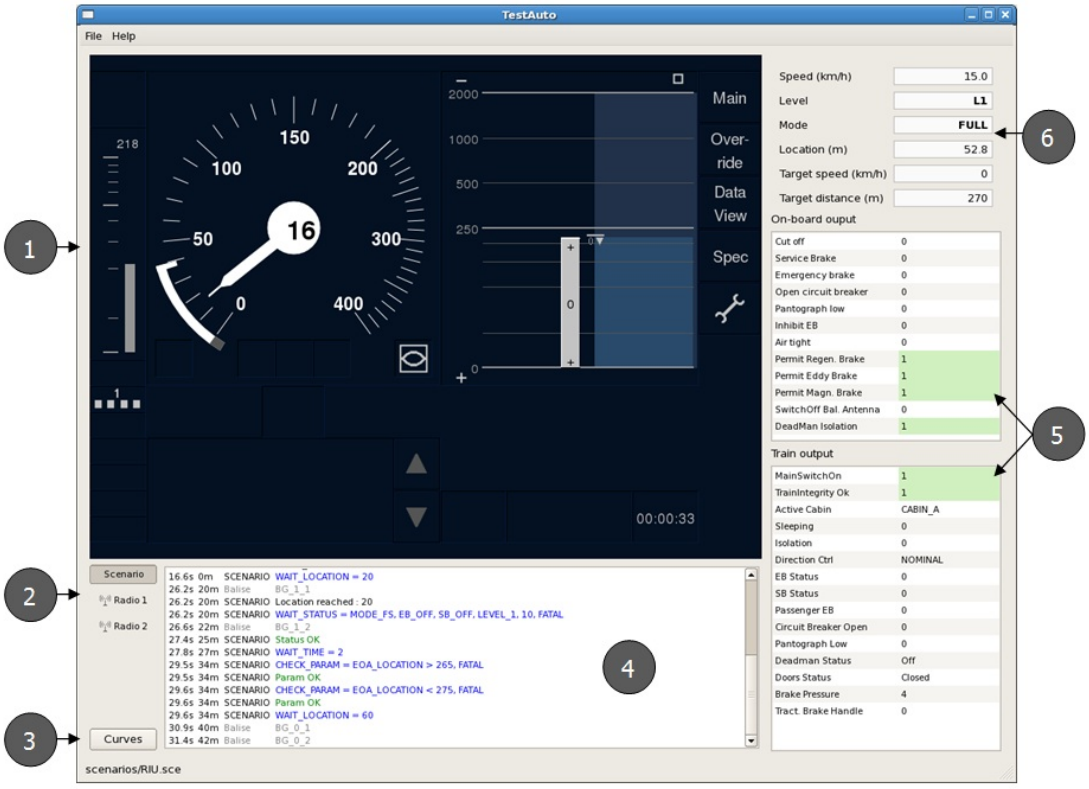


Figure 2. Main window

Description :

1. Integrated ERA DMI
2. Logs selector. Default is “Scenario”.
3. Curve window button
4. Logs display
5. TIU : Output of train and on-board
6. Odometric data and internal status

3.3.1 DMI

Driver actions of a scenario are simulated on the DMI by automated clicks. For each driver action the sequence of clicks are saved in a configuration file .testrunner_rc located in the folder of the test runner application.

Example:

```
DriverID = "click(600,350);wait(200);click(500,110);wait(850);"
```

The command wait(time) permit the software to update content (display, data) before the next click. All commands shall be separated by a semicolon. Display of automated clicks is represented by a red circle on the DMI.

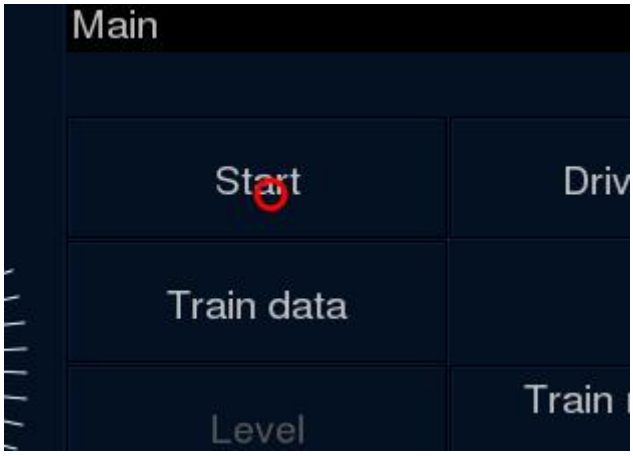


Figure 3. An automated click on Start

Warning : User actions on DMI are not inhibited. Therefore, it is not recommended to interact with the DMI during a test execution.

3.3.2 Logs view

The logs views informs user about:

- Executed commands
- Results of tests
- Trackside messages
- On-board messages

Messages can be displayed in a separated window by double-clicking on them.

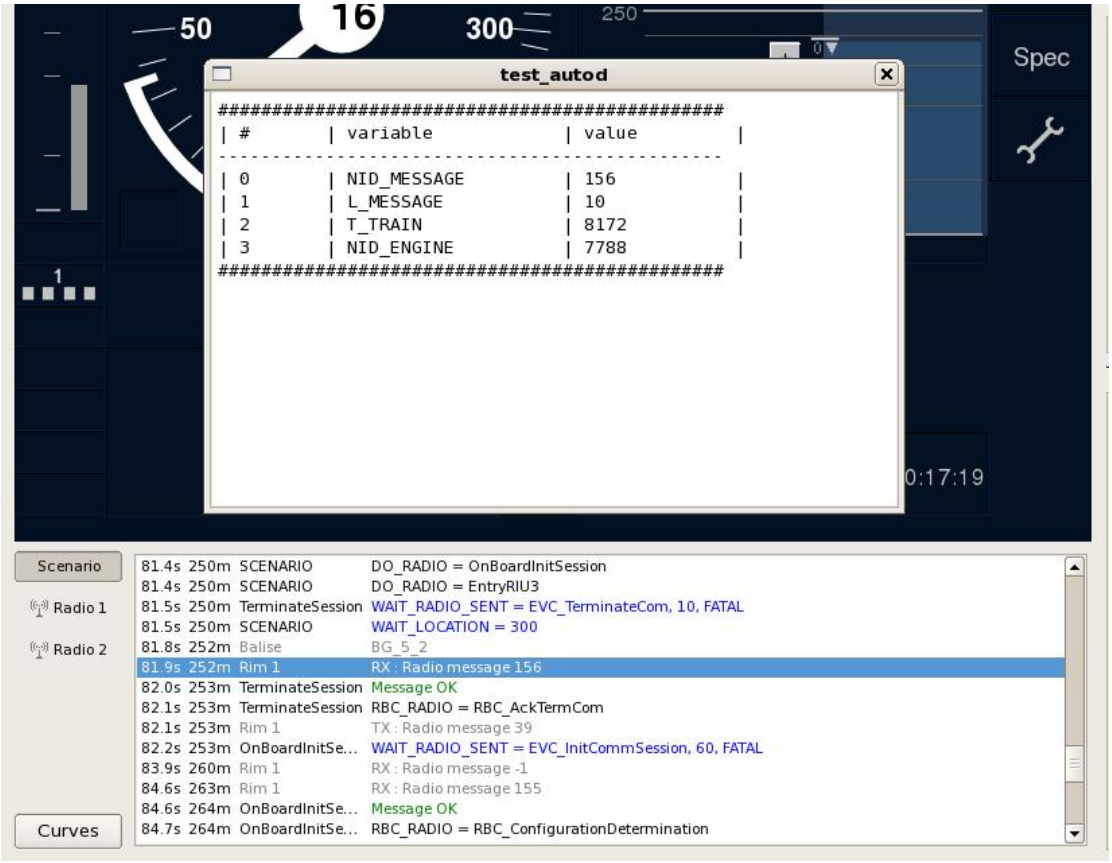


Figure 4. Message window

Radio 1 and *Radio 2* logs give status of the connection with the on-board. A green icon is displayed when connection is established with on-board. At the end of the scenario execution, the result is displayed:

- SUCCESS: the scenario has been executed until end successfully
- FAILURE: the scenario execution has been interrupted during execution due to an error in the scenario or a 'FATAL' argument in a condition test that has not been fulfilled.

3.3.3 Curve supervision display

Curve window display supervision curve computed by on-board. This window can be switched on/off by clicking Curve button on the main window. Options on the bottom right of the window

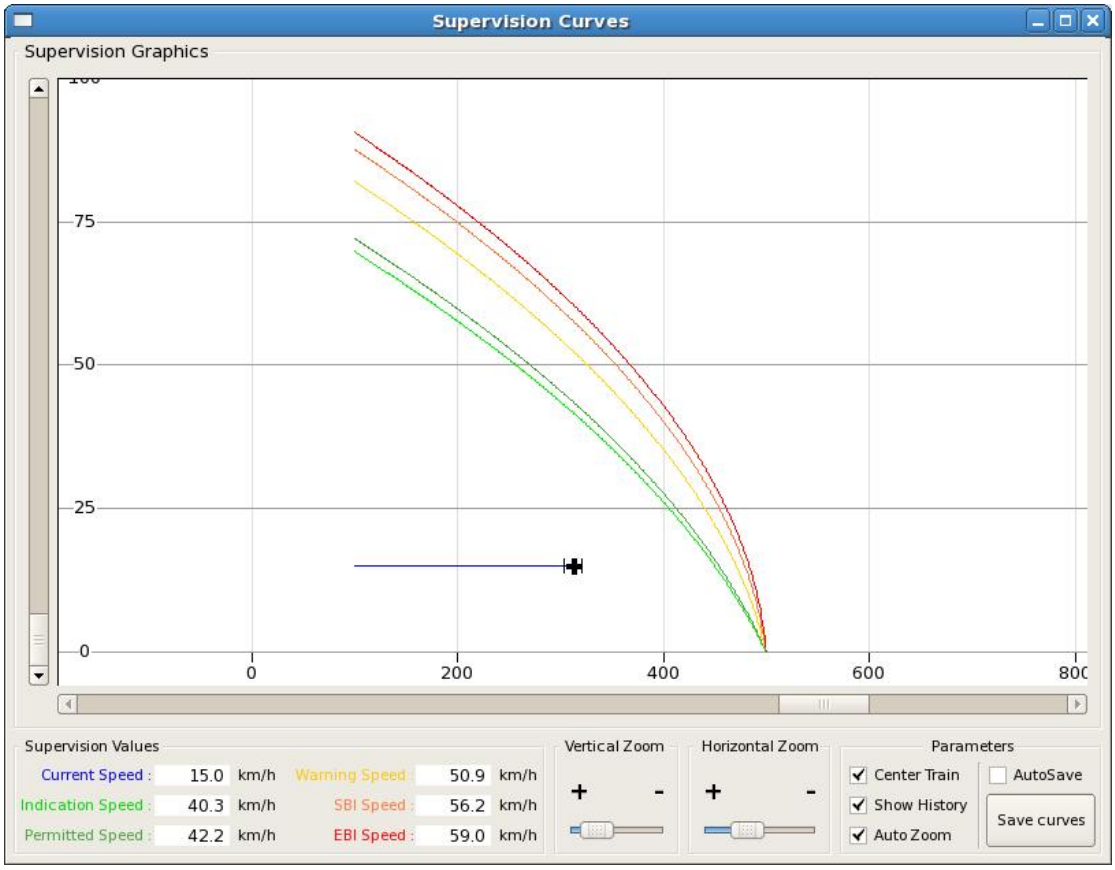


Figure 5. Curve window

allow user to adjust the view. A save function is available to record all points in a single CSV file in the data/curve/ directory.

4 Example of scenario file

```
# test scenario definition
# start in level 1, move and transition to level 2


[SCENARIO]

# SoM in level 1

DRIVER_ACTION    = MainSwitchOn
WAIT_STATUS              = MODE_SB, 2, FATAL
DRIVER_ACTION    = OpenCabinA, 2
DRIVER_ACTION    = DriverID, 2
WAIT_TIME                = 1
DRIVER_ACTION    = Level1
WAIT_TIME                = 1
DRIVER_ACTION    = TrainData
WAIT_TIME                = 1
DRIVER_ACTION    = StartOfMission
WAIT_TIME                = 1
DRIVER_ACTION    = ACK

WAIT_STATUS      = LEVEL_1, EB_OFF, SB_OFF, MODE_SR, 5, FATAL

DRIVER_ACTION = DirectionNominal
MOVE_TRAIN

WAIT_LOCATION = 30

DO_RADIO = Connection+MA


# wait TAF to level border
WAIT_LOCATION = 100
WAIT_RADIO_SENT = MArequestFree, 5, FATAL


# wait level transition location
WAIT_LOCATION = 130

WAIT_STATUS = LEVEL_2, EB_OFF, SB_OFF, MODE_FS, 5, FATAL

WAIT_LOCATION = 530
WAIT_STATUS = LEVEL_2, EB_OFF, SB_OFF, MODE_FS, 5, FATAL


# ---------------------- Radio Procedures ---------------------

[Connection+MA]
WAIT_RADIO_SENT = InitCommSession, 10, FATAL
RBC_RADIO                = ConfigurationDetermination
```

```
WAIT_RADIO_SENT = SessionEstablished, 5, FATAL
WAIT_RADIO_SENT = ValidatedTrainData, 5, FATAL
RBC_RADIO                = AckTrainData
RBC_RADIO                = MA



# ---------------------- RBC telegrams -------------------------

[ConfigurationDetermination]
NID_MESSAGE = 32
NID_LRBG          = 1
M_VERSION = 16d

[AckTrainData]
NID_MESSAGE = 8
NID_LRBG          = 1

[MA]
NID_MESSAGE = 3
NID_LRBG          = 1
# MA data
NID_PACKET        = 15
Q_DIR             = 1
Q_SCALE           = 1
L_ENDSECTION = 5000
# SSP
NID_PACKET        = 27
Q_DIR             = 1
Q_SCALE           = 1
V_STATIC          = 60
# Gradient
NID_PACKET        = 21

# ------------------------ EVC telegrams ----------------------

[InitCommSession]
NID_MESSAGE = 155

[SessionEstablished]
NID_MESSAGE = 159

[ValidatedTrainData]
NID_MESSAGE = 129
NID_PACKET = 0
NID_PACKET = 11


[MArequestFree]
NID_MESSAGE = 132
NID_PACKET = 9
```

```
NID_LTRBG   = 11


# ------------------------- Balises -------------------------

[BaliseTrackside]
# announcement of level transition
30 = BG1_1
35 = BG1_2

# TAF free to L2 border
100 = BG2_1
105 = BG2_2

[BG1_1]
N_PIG    = 0
N_TOTAL = 1
NID_C    = 0
NID_BG   = 1
Q_LINK   = 1
# level transition packet
NID_PACKET                = 41
Q_DIR                     = 1
Q_SCALE                   = 1
D_LEVELTR                 = 100
M_LEVELTR                 = 3
L_ACKLEVELTR     = 50
# order to contact RBC
NID_PACKET                = 42
Q_DIR                     = 1
Q_RBC                     = 1
# Ending packet
NID_PACKET = 255

[BG1_2]
N_PIG    = 1
N_TOTAL = 1
NID_C    = 0
NID_BG   = 1
Q_LINK   = 1
# Ending packet
NID_PACKET = 255


[BG2_1]
N_PIG    = 0
N_TOTAL = 1
NID_C    = 0
NID_BG   = 2
Q_LINK   = 1
# TAF free to level 2 border
```

```
NID_PACKET                  = 90
Q_DIR                       = 1
NID_BG                      = 11
# Ending packet
NID_PACKET = 255

[BG2_2]
N_PIG    = 1
N_TOTAL = 1
NID_C    = 0
NID_BG   = 2
Q_LINK   = 1
# Ending packet
NID_PACKET = 255




# ------------------------ Speed Profile ----------------------

[SpeedProfile]
0 = 0
500 = 40
1000 = 0

# ------------------------ configuration ----------------------


[Config_EVCInit]
LINE_LEVEL        = 1
COUNTRY_ID        = 0
GROUP_ID          = 4522
DISTANCE          = 50
DIRECTION         = NOMINAL
VALIDITY          = VALID

[Config_TrainData]
ETCS_ID           = 4554
BALISEANTENNA_OFFSET = 10
TRAIN_CATEGORY = 1
LOADING_GAUGE_MASK = 1
AXLE_LOAD = 237000
TRACTION_POWERS = 0, 1, 5, 41, 78, 11
TRAIN_LENGTH = 100
TRAIN_MASS = 142000.0
TRAIN_MAXACCEL = 0.5
SPEED_MAX = 300
EB_TIME = 1.0
EB_TIME = 1.0
CUTOFF_TIME = 1.0
BALISE_COM_AVAILABLE              = 1
```

```
LOOP_COM_AVAILABLE                  = 1
RADIO_COM_AVAILABLE                 = 1
INTEGRITY_DEVICE_AVAILABLE          = 1
SERVICE_BRAKE_AVAILABLE             = 1
ETCS_PHONE1                         = 1234123412341234
ETCS_PHONE2                         = 1234123412341235


[Config_SRSNationalDefaults]
COUNTRY_ID = 0
DRIVER_ADHESION = 0
SH_SPEED = 30
SR_SPEED = 40
OS_SPEED = 30
UN_SPEED = 100
RELEASE_SPEED = 40
ROLLAWAY_DISTANCE = 2
SB_USETOTARGET = 1
EB_RUNRELEASE = 0
OVERRIDEEOA_ENTRYSPEED = 0
OVERRIDEEOA_MAXSPEED = 30
OVERRIDEEOA_MAXDISTANCE = 200
OVERRIDEEOA_MAXTIME = 60
DRIVERID_RUNCHANGE = 1
```