Frequently Used SCP Commands (FUSC)

Note: work on this page started in March 2014; with any upcoming user question related to the subject "SCP" we plan to extend this page; also, commands identified as "useful" by us during daily operation will be added.

The indication of the *availability* is given in detail for commands introduced with VTD 1.3.5 or later. Commands available in previous versions do not have a dedicated availability information.

Query <Query>

Query the ID or name of a player or object

purpose: retrieve a player's or object's name from the known ID or vice versa

example 1:
command:

```
<Query entity="player" name="Ego"/>
```

result:

```
<Reply category="vehicle" entity="player" id="1" name="Ego" type="car" />
```

example 2:
command:

```
<Query entity="player" id="2"/>
```

result:

```
<Reply category="obstacle" entity="player" id="2" name="New Object 01" type="other" />
```

stages: INIT_DONE, RUN

remarks: not only the ID but also other information concerning the classification of the entity is returned

availability: VTD 1.3.4+

Query the template characteristics of a player

purpose: retrieve a player's template information (graphics model etc.)

example:
command:

```
<Query entity="player" name="Ego"><Template /></Query>
```

result:

```
<Reply entity="player" id="1" name="Ego"><Template id="104" name="VW Passat Variant 2011 - black" /></Reply>
```

stages: INIT_DONE, RUN

remarks: none

availability: VTD 1.3.4+

Query the inertial position of a player or object

purpose: retrieve a players's or object's inertial position

command:

```
<Query entity="player" name="Ego"><PosInertial /></Query>
```

result:

```
<Reply entity="player" name="Ego"><PosInertial hDeg="60.915" pDeg="0.000" rDeg="0.000" x="-1619.686" y="3529.328" z="0.000" />
```

stages: INIT_DONE, RUN

remarks: none

availability: VTD 1.3.x

Query the list of plug-ins loaded within the ModuleManager

purpose: retrieve a list of plug-ins loaded by the module manager

command:

```
<Query entity="moduleManager"> <Plugin/> </Query>
```

result:

```
<Reply entity="moduleManager">
  <Sensor name="DefaultPerfectSensor"/>
  <DynamicsPlugin name="viTrafficDyn"/>
  <DynamicsPlugin name="viTrafficDynComplex"/>
</Reply>
```

stages: all

remarks: none

availability: VTD 2.0.2+

Query the list of recordings in the current project

purpose: get a list of all recordings in and under the current project's sub-directory *Recordings*

command:

```
<Query entity="taskControl"> <Recordings/> </Query>
```

or

```
<Query entity="taskControl"> <Recordings absolutePath="true"/> </Query>
```

result: (without *absolutePath* being set)

```
<Reply entity="taskControl">
  <Recordings>
    <File name="myRecording_001.dat" />
    <File name="pedRecording_000.dat" />
    <File name="SubDir1/testRec.dat" />
    <File name="testRec.dat" />
    <File name="myRecording_005.dat" />
    <File name="SubDir2/SubSubDir/testRec3.dat" />
    <File name="SubDir2/testRec2.dat" />
    <File name="myRecording_006.dat" />
  </Recordings>
</Reply>
```

stages: all

remarks: if the attribute *absolutePath* is given and *true*, the absolute file paths will be returned.

availability: VTD 2.0.2+

Query camera parameters

purpose: get the position and orientation of the camera

command:

```
<Query entity="imageGenerator" ><Camera /></Query>
```

result:

```
<Reply entity="imageGenerator">
  <Camera>
    <PosInertial x="-442.029" y="2241.18" z="11.5"/>
    <ViewInertial h="-2.12238" p="0.00914571" r="0.014865"/>
  </Camera>
</Reply>
```

stages: RUN

remarks: angles are in radiant

availability: VTD 1.4.3+

Query window parameters

purpose: get information about window size and position

command:

```
<Query entity="imageGenerator" ><Window /></Query>
```

result:

```
<Reply entity="imageGenerator">
  <Camera>
    <Window width="800" height="600" x="3" y="476" />
  </Camera>
</Reply>
```

stages: RUN

remarks: none

availability: VTD 1.4.3+

Query current scenario

purpose: get information about the currently loaded scenario

command:

```
code class="xml"><Query entity="taskControl"><Scenario /></Query>
```

result:

```
<Reply entity="taskControl">
<Scenario contentViaSCP="false" filename="/home/marius/Data/vtd/VTD.2.0/Data/Projects/Current/Scenarios/Demo.xml" />
</Reply>
```

stages: INIT_DONE, RUN

remarks: none

availability: VTD 2.0.3+

Set <Set>

Position an object relative to another object or player

purpose: permanently position an object relative to another object or player, so that it appears "tethered" to the reference object

command:

```
<Set entity="player" name="Object"><PosRelative player="Ego" dx="10.0" persistent="true"/></Set>
```

stages: RUN

remarks: if you omit the attribute "persistent", the relative positioning will only take place upon receipt of the command; afterwards the objects act independently of each other again; it's also possible to permanently position players relative to other players but the connection is rigid, i.e. the tethered player will also rotate with the pivot player (e.g. in case of acceleration).

availability: VTD 1.3.5+

Position a player road relative to another player

purpose: permanently position a player road relative to another player, so that it appears "tethered" to the reference object along the road with the given distance

command:

```
<Set entity="player" name="Player"><PosRelativeRoad player="Ego" distance="10.0" persistent="true"/></Set>
```

stages: RUN

remarks: if you omit the attribute "persistent", the relative positioning will only take place upon receipt of the command; afterwards the objects act independently of each other again.

availability: VTD 1.4.1+

Set the absolute position of a player or object

purpose: place an object or player at an absolute position

command:

```
<Set entity="player" name="SlowCar"><PosInertial x="1850" y="733" hDeg="110.0"/></Set>
```

stages: RUN

remarks: after the re-positioning, the player will move on according to the given driver behavior etc. This works for internally and externally controlled players

availability: VTD 1.3.5+

Camera `<Camera>`

Set intrinsic camera parameters

purpose: set focal and principal points of the camera projection

command:

```
<Camera name="VIEW_CAMERA" showOwner="true">
  <Projection width="800" height="600" near="1" far="1500" focalX="400" focalY="300" principalX="400" principalY="300"/>
  <Set />
</Camera>
```

stages: RUN

remarks: none

availability: VTD 1.4.3+

Simulation Control `<SimCtrl>`

Enter / Control single step mode

purpose: as in replay, one may also specify during standard operation the number of steps that have to be performed until the execution goes into pause mode.

command:

```
<SimCtrl><Step size="3"/></SimCtrl>
```

stages: RUN

remarks: If issued in pause mode, the indicated number of steps will be executed before the system goes back to pause again; if issued in run mode, the indicated number of steps will be executed and then the system will go into pause mode; without the attribute "size", only one simulation frame will be executed

availability: VTD 1.3.5+

Traffic Behavior `<Traffic>`

Show the impact of collisions on other vehicles

purpose: propagate the impulse of a collision to other vehicles, thus affecting their motion vector

command:

```
<Traffic><Collision enable="true" restitutionCoeff="0.2" friction="0.8"/></Traffic>
```

stages: RUN

remarks: the computation is quite simplistic but visually "impressing". You'll have to play around with the coefficients to make the result fit your needs.

availability: VTD 1.x

Send a notification after a vehicle has reached the end of its route

purpose: tell the user that a vehicle has reached the end of its route

command:

```
<Traffic> <Config infoEndAction="true" /> </Traffic>
```

result:

```
<Path name="Path01" shape="road"><Info player="SlowCar" endAction="continue" /></Path>
```

or similar

stages: RUN

remarks: the notification can be turned on/off for all vehicles only, not for individual ones

availability: VTD 1.x

Pause Traffic Module

purpose: keep the simulation running while traffic calculations are paused

command:

```
<Traffic><Pause/></Traffic>
```

and for resuming:

```
<Traffic><Start/></Traffic>
```

stages: RUN

remarks: while paused, the traffic module will also **not** create any DRIVER_CTRL commands. Another config parameter of the TC (`<TaskControl><Traffic maxPlayerAge="-1"/></TaskControl>`)
must be set, so that the TC does not remove the (inactive) traffic players automatically.

availability: VTD 2.0.2+

TaskControl Features `<TaskControl>`

Show information about road position of first external player

purpose: debugging of databases and error reporting may be easier by providing IG snapshots that already include the position of the Ego vehicle in OpenDRIVE co-ordinates

command:

```
<TaskControl><Debug trackPos="true"/></TaskControl>
```

stages: any

remarks: not very sophisticated but quite a good help for debugging

availability: VTD 1.3.5+

Show player names in the IG window

purpose: sometimes it might be good to see the names of all players directly in the IG window

command:

```
<TaskControl><Debug playerNames="true"/></TaskControl>
```

stages: RUN

remarks: none

availability: VTD 1.3.5+

Hide simTime information in the IG window

purpose: hide the running simulation time in the IG window

command:

```
<TaskControl><Debug simTime="false"/></TaskControl>
```

Hide egoSpeed information in the IG window

purpose: hide the running speed of the external vehicle in the IG window

command:

```
<TaskControl><Debug egoSpeed="false"/></TaskControl>
```

Symbols `<Symbol>`

On-screen text

purpose: shows a red text string on the screen

command:

```
<Symbol name="myText">
  <Text data="This is the screen" colorRGB="0xff0000" size="100.0" />
  <PosScreen x="0.2" y="0.25" />
</Symbol>
```

stages: RUN

remarks: screen position of a symbol is provided in normalized co-ordinates;

availability: VTD 1.x

On-screen text, flashing

purpose: shows a flashing green text string on the screen

command:

```
<Symbol name="myText" frequency="5.0" onOffRatio="1.0">
  <Text data="This is the screen" colorRGB="0x00ff00" size="100.0" />
  <PosScreen x="0.8" y="0.25" />
</Symbol>
```

stages: RUN

remarks: frequency is in [Hz]

availability: VTD 1.x

On-screen text, disappearing automatically after 10 seconds

purpose: shows a yellow text string on the screen for a duration of 10s

command:

```
<Symbol name="myText" duration="10.0">
  <Text data="This is the screen" colorRGB="0xffff00" size="100.0" />
  <PosScreen x="0.8" y="0.25" />
</Symbol>
```

stages: RUN

remarks: duration is in [Hz]

availability: VTD 1.x

Text in 3d space at a given inertial position

purpose: shows a text at a fix inertial position

command:

```
<Symbol name="myText" >
  <Text data="Look here!" colorRGB="0x00ff00" size="100.0" />
  <PosInertial x="1795.0" y="795.0" z="5.0"/>
</Symbol>
```

stages: RUN

remarks: The text will always point to the camera

availability: VTD 1.x

Text in 3D space, relative to a known entity (e.g. Ego vehicle)

purpose: shows a text at a fix inertial position

command:

```
<Symbol name="myText">
  <Text data="Hello World!" colorRGB="0xffff00" size="50.0"/>
  <PosPlayer player="FastCar" dx="-5.0" dy="0.0" dz="3.0"/>
</Symbol>
```

stages: RUN

remarks: dx, dy, dz is in player co-ordinates; the text will always point to the camera

availability: VTD 1.x

On-screen text showing an internal parameter

purpose: shows a player's speed as on-screen text with custom formatting

command:

```
<Symbol name="scrSym" >
  <Text data="" colorRGB="0x00ffff" sizeRelative="0.15" />
  <PosScreen x="0.0" y="0.1" />
  <TextSource type="player" entityName="FastCar" parameter="speed" scaleFactor="3.6" stringFormat="FastCar: %.1fkm/h" />
```

```
</Symbol>
```

stages: RUN

remarks: shows the speed of player *FastCar* in kmph with a cyan font in 15% screen size

availability: VTD 1.x

Textured on-screen symbol

purpose: shows a textured symbol in screen-space

command:

```
<Symbol name="scrSymbol" >
  <Overlay id="0" state="2"/>
  <PosScreen x="0.5" y="0.5" />
  <RectSize width="0.2" height="0.2"/>
</Symbol>
```

stages: RUN

remarks: overlay and state must be defined in the IG configuration files; size and position are given in normalized screen co-ordinates

availability: VTD 1.x

Textured symbol in 3d space, relative to a player

purpose: shows a flashing textured symbol for 10s, relative to player "Ego", heading offset 60deg

command:

```
<Symbol name="mySymbol" frequency="2.0" onOffRatio="1.0" duration="10.0">
  <Overlay id="0" state="2"/>
  <PosPlayer player="Ego" dx="20.0" dy="0.0" dz="1.0"/>
  <Orientation type="inertial" dhDeg="60" dpDeg="0" drDeg="0"/>
  <RectSize width="3.0" height="2.0"/>
</Symbol>
```

stages: RUN

remarks: overlay and state must be defined in the IG configuration files; size is given in [m], position is given in player co-ordinates;

availability: VTD 1.x

Textured symbol in 3d space, located at a given inertial position, inertial orientation

purpose: shows a textured symbol at a given inertial position with 60deg inertial heading

command:

```
<Symbol name="mySymbol" >
  <Overlay id="0" state="2"/>
  <PosInertial x="1795.0" y="795.0" z="5.0"/>
  <RectSize width="6.0" height="4.0"/>
  <Orientation type="inertial" dhDeg="60" dpDeg="0" drDeg="0"/>
</Symbol>
```

stages: RUN

remarks: overlay and state must be defined in the IG configuration files; position and size are given in [m]

availability: VTD 1.x

Textured symbol in 3d space, located at a given road position, inertial orientation

purpose: shows a textured symbol at a given road position with 60deg inertial heading

command:

```
<Symbol name="newSymbol" >
  <Overlay id="0" state="1"/>
  <PosRoad roadId="5" s="40.0" t="-2.0" z="2.0" />
  <RectSize width="3.0" height="2.0"/>
  <Orientation type="inertial" dhDeg="60" dpDeg="0" drDeg="0"/>
</Symbol>
```

stages: RUN

remarks: overlay and state must be defined in the IG configuration files; road position must exist; symbol size is given in [m]

availability: VTD 1.x

Textured symbol in 3d space, located at a given road position and relative orientation

purpose: shows a textured symbol at a given road position with 60deg relative heading

command:

```
<Symbol name="newSymbol" >
  <Overlay id="0" state="1"/>
  <PosRoad roadId="5" s="40.0" t="-2.0" z="2.0" dhDeg="60"/>
  <RectSize width="3.0" height="2.0"/>
</Symbol>
```

stages: RUN

remarks: overlay and state must be defined in the IG configuration files; road position must exist; symbol size is given in [m]

availability: VTD 1.x

Lines in 3d space, relative to an entity

purpose: draws a flashing symbol composed of three lines relative to a known entity

command:

```
<Symbol name="lineSymbol3" frequency="1.0" onOffRatio="1.0">
  <Orientation type="relative"/> <PosPlayer player="FastCar" dx="-0.8" dy="0.7" dz="0.9"/>
  <Line width="4.0" fromX="0.0" fromY="0.0" fromZ="0.0" toX="0.0" toY="0.0" toZ="1.0" colorRGB="0xff0000"/>
  <Line width="4.0" fromX="0.0" fromY="0.1" fromZ="0.2" toX="0.0" toY="0.0" toZ="0.0" />
  <Line width="4.0" fromX="0.0" fromY="-0.1" fromZ="0.2" toX="0.0" toY="0.0" toZ="0.0" />
</Symbol>
```

stages: RUN

remarks: the first line's color entry will be applied to all lines within the same symbol

availability: VTD 1.x

Connecting arc between two entities (special case)

purpose: shows an arc between two moving objects

command:

```
<Symbol name=dyade0 colorRGB=0xffee00a0>
  <Connection shape=circular lineWidth=4.0 heightDeg="1.5" deltaAngleDeg=0.5 lineStyle=solid />
  <PosPlayer id=1 dx=2.5 dy=0.20 dz=0.6 />
  <PosPlayer id=2 dx=0 dy=0 dz=1.2 />
  <Orientation type=inertial />
  <Scale x=1.0 y=1.0 z=1.0 />
</Symbol>
```

stages: RUN

remarks: the maximum height of the symbol will be determined by *heightDeg*;

availability: VTD 1.x

Environment Configuration `<Environment>`

Change environment settings (e.g. weather)

Sun Parameters `<Sun>`

purpose: influence position and appearance of the sun in the scene; overwrite the automatic positioning of the sun by the image generator

command:

```
<Environment>
  <Sun override="true" azimuthDeg="135" elevationDeg="6"
       ambientIntensityScale="1.0" diffuseIntensityScale="1.0" specularIntensityScale="1.0"
       modelSizeScale="1.0" modelIntensityScale="1.0"/>
</Environment>
```

result: the sun will be positioned at the inertial direction `azimuthDeg` / `elevationDeg`; its intensity for ambient, diffuse and specular components will be scaled according to the factors given; the appearance of the sun model (disc) will be scaled in size and intensity

stages: RUN

remarks: any attribute that is not provided will cause the sun's appearance to remain unchanged with respect to the attribute's scope; sending `<Environment><Sun override="false"/></Environment>` will cause the sun model to behave again according to the internal mechanisms of the image generator.

availability: VTD 2.0.3+

Precipitation Parameters `<Precipitation>`

purpose: influence appearance of the precipitation effects in the image generator

command:

```
<Environment>
    <Precipitation type="rain" intensity="2.0" particleSpeed="8" particleSize="0.06" particleColor="0.8 0.8 0.8 1"/>
</Environment>
```

result: the precipitation will be modified according to the parameters provided; you may even configure red (or also purple) rain:

stages: RUN

remarks: any attribute that is not provided will cause the precipitation's appearance to remain unchanged with respect to the attribute's scope

availability: VTD 2.0.3+

Contact Point `<ContactPoint>`

purpose: initiate sending of road information via RDB at given points relative to a player (typically: four contact points around Ego vehicle)

command:

```
<ContactPoint id="1" > <PosPlayer player="Ego" dx="2.0" dy="0.7"/> </ContactPoint>
<ContactPoint id="2" > <PosPlayer player="Ego" dx="2.0" dy="-0.7"/> </ContactPoint>
<ContactPoint id="3" > <PosPlayer player="Ego" dx="0.0" dy="-0.7"/> </ContactPoint>
<ContactPoint id="4" > <PosPlayer player="Ego" dx="0.0" dy="0.7"/> </ContactPoint>
```

result: RDB messages of type `RDB_PKG_ID_CONTACT_POINT` will be sent continuously

stages: INIT_DONE, RUN

remarks: for high frequency road contact calculation, use the OdrGateway or the OdrManager

availability: VTD 1.x

Headlight Control `<LightSource>`

purpose: define and modify headlights of a vehicle

command:

```
<LightSource name="New Lightsource">
<On /><PosRelative dx="1.800000" dy="0.700000" dz="0.400000" player="Ego" />
<ViewRelative dhDeg="0.000000" dpDeg="0.000000" drDeg="0.000000" />
<Config autoMode="false" state="2" template="0" />
<Frustum bottomDeg="50.000000" far="100.000000" leftDeg="50.000000" near="1.000000" rightDeg="50.000000" topDeg="50.000000" />
<Intensity ambient="1.600000" diffuse="0.400000" specular="0.000000" />
<Attenuation constant="0.000000" linear="0.000000" quadratic="0.100000" />
</LightSource>
```

stages: RUN

remarks: you may add multiple headlights, of course, just make sure that each has a unique ID; for interactive control of a headlight, consider sending appropriate messages via RDB (RDB_PKG_ID_LIGHT_SOURCE)

availability: VTD.1.x

further information: see also VTD_FAQ_ImageGenerator

Traffic Light Control `<TrafficLight>`

Set the state mask directly

purpose: set the state mask of a given traffic light (or all traffic lights at once)

command:

```
<TrafficLight id="221"><Mask value="0x01000000"/></TrafficLight>
```

result: the mask (32bit integer) will be set to the given value and changes the appearance of the indicated traffic light accordingly (only works permanently if no traffic light program is assigned to the respective controller); if the attribute "id" is missing, then all traffic lights will be set to the identical value

stages: RUN

remarks: for the standard VIRES traffic lights, the following mask logics apply (masks may also be combined)

`0x10000000`

`0x01000000`

`0x00100000`

availability: VTD 1.3+

ImageGenerator Control `<ImageGenerator>`

Operate a generic switch

purpose: operate (generic) switches in the scene graph.

command:

```
<ImageGenerator><Switch objectId="2" switchId="1" scopeId="3" state="2"/></ImageGenerator>
```

with

- `scopeId`: category of affected element in the scene graph
  - 3 = dynamic object (player)
- `objectId`: the actual numeric ID of the object (in example: playerId)
- `state`: the numeric state (range depending on modeling)

result: the switch in the scene graph will be set to the respective state

stages: RUN

availability: VTD 1.4.3+

example: see here

ImageGenerator PostProcessing `<ImageGenerator>`

Change values of uniform variables

purpose: Change values of uniform variables at runtime

command:

```
<ImageGenerator>
  <PostProcessing>
    <Step name="StepName">
      <Uniform type="float" name="u_scalefactor" value="1.0805" />
    </Step>
  </PostProcessing>
</ImageGenerator>
```

result: The value will be updated for the shader of the step, if it is defined.

stages: RUN

Record and Playback

Recording to internal buffer

Instead of recording data to a file, an internal buffer may be used. It will be available for instantaneous replay. The buffer is a ringbuffer holding the tbd. last seconds of simulation data. The configuration is achieved as follows:

```
<code class="xml">
<Record>
  <Config writeMem="true" recBufferSize="20"/>
  <Start/>
</Record>
</code>
```

The attribute *writeMem* activates buffer recording and *recBufferSize* determines the size of the buffer in [s].

The recording is stopped by

```
<code class="xml">
<Record>
  <Stop/>
</Record>
</code>
```

For replaying the buffer, use the following command sequence:

```
<code class="xml">
<Replay>
  <Start/>
</Replay>
</code>
```

The playback is stopped by

```
<code class="xml">
<Replay>
  <Stop/>
</Replay>
</code>
```

The buffer content may also be dumped to a file using the following syntax

```
<code class="xml">
<Record>
  <File path="/home/vtd/VTD.2.0/Data/Projects/Current/Recordings" name="testRec.dat" overwrite="true"/>
  <Dump/>
</Record>
</code>
```

availability: VTD 2.0.2+

Recording to file

This is the standard (i.e. default) way of recording. Unless you have been recording to memory before, all you have to provide are the filename and the start/stop command.

Example:

```
<code class="xml">
<Record>
  <Config writeFile="true"/>
  <File path="/home/vtd/VTD.2.0/Data/Projects/Current/Recordings" name="testRec.dat" overwrite="true"/>
  <Start/>
</Record>
</code>
```

Note: the <Config> entry may be omitted if you have **not** been recording to memory before.

Summary

The following SCP script illustrates the different ways of recording / playing back data:

```
<code class="xml">
# prepare the scenario
0 "<SimCtrl><LoadScenario filename="crosssing8Demo.xml"/><Init mode="operation"/></SimCtrl>"

# recording to memory only
+5s  "<Record><Config writeMem="true" recBufferSize="20"/><Start/></Record>"
+1   "<SimCtrl><Start mode="operation"/></SimCtrl>"
+1s   "<Symbol name="expl01" > <Text data="Recording to memory" colorRGB="0xffff00" size="50.0" /> <PosScreen x="0.01" y="0.05
+20s "<Record><Stop/></Record>"
+1   "<SimCtrl><Stop/></SimCtrl>"
+1   "<Replay><Stop/></Replay>"
#replay
+1   "<Replay><Start/></Replay>"
+1s  "<Symbol name="expl02" > <Text data="Replay from memory" colorRGB="0xffff00" size="50.0" /> <PosScreen x="0.01" y="0.10" /
+10s "<Replay><Stop/></Replay>"
#replay again
+1   "<Replay><Start/></Replay>"
+1s  "<Symbol name="expl02" > <Text data="Replay from memory again" colorRGB="0xffff00" size="50.0" /> <PosScreen x="0.01" y="(
+10s "<Replay><Stop/></Replay>"
#
# case 2: recording to file only
+1   "<Record><Config writeFile="true"/><File path="/home/vtd/VTD.2.0/Data/Projects/Current/Recordings" name="testRec.dat" ove
+1   "<SimCtrl><Start mode="operation"/></SimCtrl>"
+1s  "<Symbol name="expl01" > <Text data="Recording to file" colorRGB="0xff0000" size="50.0" /> <PosScreen x="0.01" y="0.05" /
+20s "<Record><Stop/></Record>"
+1   "<SimCtrl><Stop/></SimCtrl>"
#replay
+1   "<Replay><File path="/home/vtd/VTD.2.0/Data/Projects/Current/Recordings" name="testRec.dat"/><Start/></Replay>"
+1s  "<Symbol name="expl02" > <Text data="Replay from file" colorRGB="0xff0000" size="50.0" /> <PosScreen x="0.01" y="0.10" /><
+10s "<Replay><Stop/></Replay>"
#replay again
+1   "<Replay><File path="/home/vtd/VTD.2.0/Data/Projects/Current/Recordings" name="testRec.dat"/><Start/></Replay>"
+1s  "<Symbol name="expl02" > <Text data="Replay from file again" colorRGB="0xff0000" size="50.0" /> <PosScreen x="0.01" y="0.1
+10s "<Replay><Stop/></Replay>"

# case 3: recording to both
+1s   "<Record><Config writeMem="true" writeFile="true" recBufferSize="20"/><File path="/home/vtd/VTD.2.0/Data/Projects/Curren
+1   "<SimCtrl><Start mode="operation"/></SimCtrl>"
+1s  "<Symbol name="expl01" > <Text data="Recording to both" colorRGB="0x0000ff" size="50.0" /> <PosScreen x="0.01" y="0.05" /
+20s "<Record><Stop/></Record>"
+1   "<SimCtrl><Stop/></SimCtrl>"
#
# replay mem
+1   "<Record><Config writeMem="true" recBufferSize="20"/></Record>"
+1   "<Replay><Start/></Replay>"
+1s  "<Symbol name="expl02" > <Text data="Replay from memory" colorRGB="0x0000ff" size="50.0" /> <PosScreen x="0.01" y="0.10" /
+10s "<Replay><Stop/></Replay>"
# replay file
+1  "<Record><Config writeMem="false" writeFile="true"/></Record>"
+1  "<Replay><File path="/home/vtd/VTD.2.0/Data/Projects/Current/Recordings" name="testRec.dat"/><Start/></Replay>"
+1s "<Symbol name="expl02" > <Text data="Replay from file" colorRGB="0x0000ff" size="50.0" /> <PosScreen x="0.01" y="0.10" /><
+9s "<Symbol name="expl02" > <Text data="END OF TEST" colorRGB="0xffff00" size="50.0" /> <PosScreen x="0.01" y="0.05" /></Symb
+1s "<Replay><Stop/></Replay>"
</code>
```

vtd64.png (97.8 KB) Marius Dupuis, 11.04.2014 16:24

vtd143.png (102 KB) Marius Dupuis, 02.06.2014 20:15

vtd144.png (89.7 KB) Marius Dupuis, 02.06.2014 20:19

vtd188.png (44.3 KB) Marius Dupuis, 30.12.2014 20:04

vtd187.png (44.4 KB) Marius Dupuis, 30.12.2014 20:04

vtd186.png (44.4 KB) Marius Dupuis, 30.12.2014 20:04

relativePersistentPlayer.png (121 KB) Andreas Biehn, 18.02.2015 15:33

relativeRoadPersistentPlayer.png (132 KB) Andreas Biehn, 18.02.2015 15:33

vtd457s.png (155 KB) Marius Dupuis, 03.01.2017 18:26

vtd458s.png (118 KB) Marius Dupuis, 03.01.2017 18:26