



FS25 Vehicle Assistance System

(Modding) Documentation by SoSi-Modding



Created by: [SoSi] Janni1101

Last update: 2026/02/12



Reverse Camera Control (RCC)

What It Does

- Switches defined nodes/animations on while reversing (e.g., camera screens, reverse work lights).
- Optionally keeps them visible for a short forward roll-out.
- Can disable conflicting vehicle animations (that are e.g. used when VAS Mod is not active) so RCC fully controls them.

How It Behaves

- Active only when the vehicle drives backward; if it was just in reverse, it can stay active while moving forward until maxForwardSpeed is exceeded.
- When speed is above maxForwardSpeed in forward direction, all RCC-controlled nodes/animations are hidden again.

Vehicle XML Entries

Minimal Example:

```
<vehicleAssistanceSystem>
  <reverseCameraActive>
    <cameraVisibility maxForwardSpeed="10" animation="showCameraIcon">
      <node node="0>0|0"/>
    </cameraVisibility>
  </reverseCameraActive>
</vehicleAssistanceSystem>
```

Vehicle Animation Cleanup:

Optional removal of conflicting animations (prevents double-control by base animations):

```
<animations>
  <animation name="reverseCameraDisplay" removeWithRCC="true"/>
</animations>
```



Keys:

reverseCameraActive section:

- cameraVisibility#maxForwardSpeed: Forward speed threshold (km/h) below which RCC keeps visibility after leaving reverse.
- cameraVisibility#animation: Optional animation name to play while active (reset to time 0 when inactive).
- cameraVisibility.node#node: One or more node indices to show/hide with RCC.

animations section:

- animation#removeWithRCC: If true, RCC clears the AnimatedVehicle parts so only RCC drives that animation.



Automatic Emergency Braking (AEB)

What It Does

- Warns driver via sound and brakes automatically to avoid collisions while driving forward.
- Works for static obstacles (walls, buildings, trees) and vehicles/traffic.
- Active when: engine on, moving forward, speed ≥ 10 km/h, AEB enabled in the vehicle XML.

How It Behaves

- Warning distance: 5.0 m. Plays warning sound (always louder indoors than outdoors) and can trigger animations.
- Brake distance: 3.25 m. Increases brake force, cancels throttle, applies full brake, disables cruise control.
- Turn signal effect: With LEFT or RIGHT indicator (not hazard), both distances are halved.
- Vehicle speed scaling: Above ~ 25 km/h, distances grow mildly for vehicle hits; static obstacles keep fixed distances.



Vehicle XML Entries

The Very Least:

```
<vehicleAssistanceSystem>
  <automaticEmergencyBraking active="true"/>
</vehicleAssistanceSystem>
```

Optional Custom Settings Example:

```
<vehicleAssistanceSystem>
  <automaticEmergencyBraking active="true">
    <playAnimation name="showDisplayPopUp" playTime="5"
trigger="warning"/>
    <playAnimation name="showDashboardWarningLights" trigger="brake"/>
  <sounds>
    <sound name="warning">
      <sound file="yourWarningSound.ogg"/>
    </sound>
  </sounds>
</automaticEmergencyBraking>
</vehicleAssistanceSystem>
```

Keys:

custom animations:

- playAnimation#name: Vehicle animation name.
- playAnimation#trigger: When to play "warning" or "brake".
- playAnimation#playTime: OPTIONAL seconds until the animation is reversed (default is 0 so the animation is instantly reversed after trigger isn't active anymore).

custom sounds:

- sounds.sound#name: Name of the sound (only "warning")
- sounds.sound.sound#file: Path to the sound file



Parking Distance Control (PDC)

What It Does

- Provides acoustic feedback (beep/continuous sounds) for obstacles detected by (any number of) sensors at the front and/or rear of the vehicle.
- Supports up to 3 configurable distance zones, each with its own sound and optional animation.
- Can be configured for both front and rear sensors (front & back work independently), and disables rear sensors if a trailer is attached.

How It Behaves

- Only active when engine is running and vehicle speed is below the configured maxSpeed (default: 8 km/h).
- Sensors cast rays forward/backward to detect obstacles within maxDistance (default: 1.5 m).
- For each zone, if an obstacle is detected within the zone's distance, the corresponding sound (always louder indoors than outdoors) and (optionally) animation are triggered.
- If a zone is left (e.g. by moving away from the obstacle), the animation is immediately reversed.
- If PDC is deactivated (engine off, speed too high, etc.), all zone animations are reversed and sounds are stopped.

Vehicle XML Entries

The Very Least:

```
<vehicleAssistanceSystem>
  <parkingDistanceControl>
    <sensors>
      <sensor node="0>0|0" type="back"/>
      <!-- OR <sensor node="0>0|0" type="front"/> -->
    </sensors>
  </parkingDistanceControl>
</vehicleAssistanceSystem>
```

**Optional Custom Settings Example:**

```
<vehicleAssistanceSystem>
  <parkingDistanceControl maxDistance="1.5" maxSpeed="8">
    <sensors>
      <sensor node="0>0|0" type="back"/>
      <sensor node="0>0|1" type="back"/>
      <sensor node="0>0|2" type="front"/>
      <sensor node="0>0|3" type="front"/>
    </sensors>
    <zones>
      <zone distance="1.2" interval="900" volume="0.35"
pitch="0.95 playAnimation="anim1"/>
      <zone distance="0.7" interval="450" volume="0.6" pitch="1.0"
playAnimation="anim2"/>
      <zone distance="0.35" continuous="true" volume="0.9" pitch="1.0"
playAnimation="anim3"/>
    </zones>
    <sounds>
      <sound name="beep">
        <sound file="yourBeepSound.ogg"/>
      </sound>
      <sound name="continuous">
        <sound file="yourContinuousSound.ogg"/>
      </sound>
    </sounds>
  </parkingDistanceControl>
</vehicleAssistanceSystem>
```



Keys:

custom base settings:

- `maxDistance`: Maximum detection distance in meters (default: 1.5)
- `maxSpeed`: Maximum speed for PDC to be active (default: 8 km/h)

custom sensors:

- `sensors.sensor#node`: Node index (empty transform group) for the sensor
- `sensors.sensor#type`: Sensor type ("front" or "back")

custom zones:

- `zones.zone#distance`: Distance threshold for this zone (meters)
- `zones.zone#interval`: Beep interval in ms (ignored for continuous zones)
- `zones.zone#volume`: Sound volume for this zone
- `zones.zone#pitch`: Sound pitch for this zone
- `zones.zone#continuous`: If true, plays continuous sound instead of beeping
- `zones.zone#playAnimation`: OPTIONAL Animation name to play when this zone is triggered

custom sounds:

- `sounds.sound#name`: Name of the sound ("beep" or "continuous")
- `sounds.sound.sound#file`: Path to the sound file



Animation Action Control (AAC)

What It Does

- Maps custom animations and sounds to input actions with optional configuration filters.
- Supports multi-group sound controls (toggle + switch) and individual simple sounds (HOLD or TOGGLE).
- Persists states in savegames and syncs them in multiplayer.

How It Behaves

- Animations: toggle animation state on input; can show custom toggle texts.
- Sound control configurations: toggle a sound group, optionally cycle through multiple groups via a second input; can trigger animations per group.
- Simple sounds: one-off or looping sounds that may play an animation while active.
- Configuration filters restrict availability to specific design/config variants.



Vehicle XML Entries

Minimal structure (pick what you need, e.g. decide between toggleText general text or separate on / off text; add an extra animation when switching to a specific siren; ...):

```
<vehicleAssistanceSystem>
  <animationActionControl>
    <animations>
      <animation name="toggleRWS" animationInputButton="VAS_TOGGLE_RWS"
actionTextPriority="GS_PRIO_NORMAL">
        <toggleText text="$l10n_input_VAS_TOGGLE_RWS"
on="$l10n_vas_notification_RWS_enabled"
off="$l10n_vas_notification_RWS_disabled"/>
        <configurationFilter name="design" index="1 2 3" />
      </animation>
    </animations>

    <soundControlConfigurations>
      <soundControlConfiguration
name="$l10n_vas_configuration_soundControl"
toggleSoundGroupButton="VAS_TOGGLE_SOUNDGROUP"
switchSoundGroupButton="VAS_SWITCH_SOUNDGROUP" inputMode="TOGGLE"
actionTextPriority="GS_PRIO_HIGH">
        <soundGroup name="City Siren" animation="sirenCity">
          <sound file="sounds/sirenCity.ogg" pitch="1.0"
volume="1.0" isLooping="true" />
        </soundGroup>
        <soundGroup name="Country Siren" animation="sirenCountry">
          <sound file="sounds/sirenCountry.ogg" pitch="1.0"
volume="1.0" isLooping="false" />
        </soundGroup>
      </soundControlConfiguration>
    </soundControlConfigurations>

    <sounds>
      <sound name="AIRHORN" soundInputButton="VAS_AIR_HORN"
inputMode="HOLD" actionTextPriority="GS_PRIO_NORMAL" animation="airHorn">
        <configurationFilter name="design" index="2 3 4" />
        <sound file="sounds/airHorn.ogg" pitch="1.0" volume="0.6"
isLooping="true" />
      </sound>
    </sounds>
  </animationActionControl>
</vehicleAssistanceSystem>
```

**Keys:**

animations (per key):

- `animations.animation#name`: Animation name; must exist on the vehicle.
- `animations.animation#animationInputButton`: InputAction to toggle the animation.
- `animations.animation#actionTextPriority`: GS_PRIO_NORMAL | GS_PRIO_HIGH | GS_PRIO_LOW (default GS_PRIO_NORMAL).
- `animations.animation.toggleText#text|on|off`: Optional texts shown for the action state (supports \$l10n_ prefix).
- `animations.animation.configurationFilter#name / #index`: Optional config filter (space-separated indices, 1-based).

soundControlConfigurations:

- `soundControlConfiguration#name`: Display name (supports \$l10n_ prefix).
- `soundControlConfiguration#toggleSoundGroupButton`: Input to toggle the first/active sound group.
- `soundControlConfiguration#switchSoundGroupButton`: Optional input to cycle to the next sound group.
- `soundControlConfiguration#inputMode`: HOLD or TOGGLE (default TOGGLE).
- `soundControlConfiguration#actionTextPriority`: Priority for the action text (default GS_PRIO_NORMAL).
- `soundControlConfiguration.configurationFilter#name|index`: Optional config filter.
- `soundGroup#name`: Label for the group (supports \$l10n_ prefix).
- `soundGroup#animation`: Optional animation while this group is active.
- `soundGroup.sound`: Standard SoundManager paths (file/pitch/volume/isLooping, etc.).

sounds (per key):

- `sounds.sound#name`: Name for a simple sound (supports \$l10n_ prefix).
- `sounds.sound#soundInputButton`: InputAction for this sound.
- `sounds.sound#inputMode`: HOLD or TOGGLE (default TOGGLE).
- `sounds.sound#actionTextPriority`: Priority for the action text (default GS_PRIO_NORMAL).
- `sounds.sound#animation`: Optional animation while the sound is active.
- `sounds.sound.configurationFilter#name / #index`: Optional config filter (space-separated indices, 1-based).
- `sounds.sound.sound`: Standard SoundManager paths (file/pitch/volume/isLooping, etc.).



Beacon Light Control (BLC)

What It Does

- Manages toggleable beacon light groups (e.g., warning lights, strobe lights, work lights) via input actions.
- Groups can contain multiple individual beacon lights with configurable blinking patterns.
- Supports optional animations that play when the light group is active.
- Persists states in savegames and syncs them in multiplayer.

How It Behaves

- Each beacon light group is toggled on/off by a dedicated input action button.
- When activated, all beacon lights in the group are rendered; when deactivated, they are stopped.
- Optional animation plays forward when the group is activated and backward when deactivated.
- Configuration filters restrict availability to specific design/color variants.
- Toggle texts can display dynamic state information (e.g., "Lights ON"/"Lights OFF") with localization support.

Vehicle XML Entries

The Very Least:

```
<vehicleAssistanceSystem>
  <beaconLightControl>
    <beaconLightGroups>
      <beaconLightGroup name="warningLights"
inputButton="VAS_TOGGLE_RWS" animation="toggleWarningLights"
actionTextPriority="GS_PRIO_HIGH">
        <toggleText on="$l10n_vas_notification_RWS_enabled"
off="$l10n_vas_notification_RWS_disabled"/>
        <configurationFilter name="design" index="1 2 3"/>
        <configurationFilter name="color" index="1"/>
        <beaconLight>
          <staticLight node="0>0|1" intensity="500"
multiBlink="true" multiBlinkParameters="100 300 100 0"/>
        </beaconLight>
      </beaconLightGroup>
    </beaconLightGroups>
  </beaconLightControl>
</vehicleAssistanceSystem>
```

**Optional Custom Settings Example:**

```
<vehicleAssistanceSystem>
  <beaconLightControl>
    <beaconLightGroups>
      <beaconLightGroup name="warningLights"
inputButton="VAS_TOGGLE_RWS" animation="toggleWarningLights"
actionTextPriority="GS_PRIO_HIGH">
        <toggleText on="$l10n_vas_notification_RWS_enabled"
off="$l10n_vas_notification_RWS_disabled"/>
        <configurationFilter name="design" index="1 2 3"/>
        <configurationFilter name="color" index="1"/>
        <beaconLight>
          <staticLight node="0>0|1" intensity="500"
multiBlink="true" multiBlinkParameters="100 300 100 0"/>
        </beaconLight>
      </beaconLightGroup>
    </beaconLightGroups>
  </beaconLightControl>
</vehicleAssistanceSystem>
```

Keys:

- beaconLightGroup#name: Unique identifier for this beacon light group.
- beaconLightGroup#inputButton: InputAction to toggle this beacon light group on/off.
- beaconLightGroup#animation: Optional animation name to play when the group is activated/deactivated.
- beaconLightGroup#actionTextPriority: GS_PRIO_NORMAL | GS_PRIO_HIGH | GS_PRIO_LOW (default GS_PRIO_NORMAL).
- toggleText#text: Static text shown for both states; overrides on/off if present (supports \$l10n_ prefix).
- toggleText#on: Text shown when lights are ON (supports \$l10n_ prefix).
- toggleText#off: Text shown when lights are OFF (supports \$l10n_ prefix).
- configurationFilter#name|index: Optional config filter (space-separated indices, 1-based); multiple filters all must match.
- beaconLight: Standard vanilla BeaconLight definition (staticLight or animatedLight with blinking parameters, intensity, node, etc.).



Fold / Cover / TurnOnVehicle Disabler

What It Does

- Disables the base foldable, cover and/or turnOnVehicle specializations for a vehicle.
- Hides the corresponding input action events on the client.

How It Behaves

- When disabled, fold/cover/turnOnVehicle actions always return false and cannot be triggered by player or AI.
- If only one feature is disabled, the other keeps its normal behavior.

Vehicle XML Entries

Minimal example with all functions disabled:

```
<vehicleAssistanceSystem disableFoldable="true" disableCover="true"  
disableTurnOnVehicle="true" />
```

Keys:

- vehicleAssistanceSystem#disableFoldable: true/false, blocks all fold actions.
- vehicleAssistanceSystem#disableCover: true/false, blocks all cover actions.



Intelligent Cruise Control (ICC)

How It Behaves

- Active only when driving forward with cruise control and speed ≥ 10 km/h.
- Casts multiple forward rays to detect vehicles/traffic and computes a dynamic follow distance based on speed (approx. 2.0 s gap, clamped).
- If another vehicle is closer than the follow distance, cruise speed is reduced/held to maintain gap; once clear, original cruise speed is restored.
- Toggleable via action event; state is saved and synced in multiplayer.

Automatic Light Control (ALC)

How It Behaves

- Low-beam control: turns on automatically at night or in bad weather (rain/snow/fog/hail) when electrics are on; can be toggled per vehicle.
- High-beam control: optional; blocks or turns off high beam when oncoming/preceding traffic is detected or in foggy conditions using forward raycasts similar to ICC.
- Remembers user toggles per vehicle and syncs states in multiplayer.