AI enhancements for CSB vehicles

Version 1.5.1 Manual

February 2019

Introduction

Purpose

This modification primarily revolves around enhancing the way in which AI-controlled instances of CSB vehicles behave. More specifically it seeks to enable the AI to:

- Leverage more vehicle functions that it is by default – i.e., under standard M+R-derived scripts – capable, such as heating and cooling.
- *Consistently* vary its preferences and consequent behavior – i.e., what functions to employ under what circumstances – on a per-vehicle and/or per-context basis, such as one vehicle "choosing" to activate its lights earlier than another. Consistency is significant: In spite of choices and actions undertaken by AI vehicles being to some extent subjective, they still ought to consistently reflect the same set of rules and principles, such that actions undertaken by one vehicle at present time do not contradict past actions undertaken by the same vehicle. For example, a vehicle may choose to open some or all of the available folding passenger windows when the temperature of the environment is perceived as "neutral". While both the temperature perception and the choice of which windows to open is subjective and variable, it must hold that under "warm" conditions at least the same or more windows than under "neutral" conditions are to be opened by that same vehicle; behaving the other way around, and/or basing window state on pure chance, while certainly easier to implement, would lack consistency.
- Adjust vehicle state non-instantaneously, so as to – to some extent at least – appear more human-like; for instance by not activating the wipers at the very instant of the very first rain droplet falling down from the sky, but after a variable delay. In other words, AI vehicles shall exhibit traits of *tolerance* and *inertia* ("laziness"), like us humans do. Furthermore, at the degree practically feasible, AI vehicle state adjustment is to occur via adjustment of the very same cockpit controls available to users – not in some magical, *Deus ex machina* kind of way.

Nonetheless, along the way of implementing such AI-dedicated changes, a number of smaller user-level improvements to apposite subsystems made it into this modification as well.

The purpose of this document is, predominantly, to provide a brief, non-technical overview of the most significant among both kinds of changes. The language used to describe individual changes is intentionally somewhat abstract – the intent is to paint the general, vehicle-neutral idea behind each major change / feature; details that are specific to individual vehicles, models, or variants, are generally glossed over. The reader is encouraged to refer to the scripts and in-line commentary therein when greater accuracy on implementation and integration details, and/or rationale behind implementation choices is sought.

Supported CSB vehicles

Currently support is limited to the "CVAG 72" vehicle of the (first-generation) MB O530 CSB module.

Compatibility

This modification was developed in conjunction with, and tested against, version 1.30 of the CSB modification package, and OMSI version 2.3.004.

Roadmap and limitations

Even though as of this release several solid months of effort have already been invested into this project, it is by no means perfect or finished; at least as far as the central theme of simulating human behavior is concerned, sky is the limit. That being said, it should also be stressed that this project does by no means intend to serve as a "silver bullet" to every OMSI-AI-related shortcoming and frustration there is. All it strives to do is more extensively leverage existing tools, and in particular the scripting engine, without otherwise resorting to exotic workarounds. Therefore, inherent limitations of the game – such as AI vehicles being unable to dynamically vary their paths – are and always will be the "hard upper bound" of this project's scope.

That aside, future versions of this modification may bring support to other vehicles of the CSB "family" as well.

Whether and when future versions materialize is a function of time availability on the developers' part and community-received feedback.

Feedback

Users are welcome and encouraged to submit constructive feedback on this modification, via the overarching CSB modification's OOF discussion thread.

Credits

Unorthodox Paradox contributed script and other asset changes, and authored the accompanying documentation.

citaro142 contributed the functional requirements of the lighting module of the AI system. He also assisted with beta-testing.

The following individuals are acknowledged for providing the foundation directly underlying this work:

Morphi has been tirelessly refining the Citaro's script base from its release to this day. His many contributions include significant VDV and transmission enhancements.

alTerr, *wizard*, et al., being the team behind the original OMSI Citaro, performed the groundwork of adapting the M+R MAN script base to the Citaro's needs.

Marcel Kuhnt, *Rüdiger Hülsmann*, et al., authored the original MAN script base, still heavily utilized by virtually all OMSI buses at present.

Terms of use

Index

1. Installation

1. Extract the `Texture` and `Vehicles` directories of the downloaded compressed archive into your root OMSI 2 installation directory. When prompted whether directory merging is desired, respond “Yes”; no preexisting files will be modified.
2. Copy the following sound files to `<OMSI2>\Vehicles\CSB_J_MB_O530\Sound-Citaro`:
   i. ...from the standard M+R MAN NL 202 / NG 272 (residing, by default, under `<OMSI2>\Vehicles\MAN_NL_NG\Sound):`
      - `D_wischerhebel.wav`
      - `klimator_100.wav`

   ii. ...from Morphi’s MB O530 package, version 4.4 or 4.5 (residing, by default, under `<OMSI2>\Vehicles\MB_O530_Modded\Sound-Citaro`):
      - pedal-bremse.wav
      - pedal-gas.wav
      - pedal-kickdown.wav
      - reversewarn.wav
3. Optionally, if you intend to use the testing / diagnostic function of the AI lighting subsystem (discussed in a later section), you may want to register with the game some additional cloud textures. To do so, copy the contents of the `append_to_clouds.cfg` file, to be found at the root of the downloaded compressed archive, and paste them into `<OMSI2>\Weather\clouds.cfg`.

Upon installation, a second, standalone “(Experimental) CVAG 72” vehicle type will be available for selection in-game, under the “CSB - Chemnitzer Verkehrs-AG” manufacturer. The corresponding vehicle file, for reference by AI-list configuration files, is `<OMSI2>\Vehicles\CSB_J_MB_O530\CSB_CVAG_72_x.bus`.

2. Significant AI-specific changes

2.1. Exterior lighting

Different AI vehicles will employ different kinds of exterior light sources under different conditions and for different reasons.

Conditions are sets of a) solar elevation angle, b) environmental brightness, c) precipitation, d) season, and e) temperature ranges, subjectively interpreted by each vehicle. Depending on the interpretation, each vehicle is said to effectively be in either a) a *seeing*, or b) an *adverse-seeing*, or c) a *reduced-seeing*, or d) a *being-seen*, or e) an *off* context. Depending on the context, different light sources will be used, based on, once again, subjective preferences of each vehicle.

Seeing

A *seeing* context is when a vehicle deems that its environment is too dark to adequately make out unassisted; for example at night, or during the day, when the weather is overcast. Every vehicle will enter a seeing context at some point. Most vehicles will then switch on their headlights. Some will additionally switch on their head fog lights, if the vehicle is equipped with such.

Reduced-seeing

A *reduced-seeing* context is entered when the environment is only deemed borderline too dark; for example a little before sunrise, on a clear day. Most vehicles will not differentiate between seeing and reduced-seeing, and hence their preferred exterior light sources will remain unaffected. A minority of vehicles will however differentiate, as a consequence only activating their standstill lights under such conditions, potentially combined with head fog lights, if available.

Adverse-seeing

The *adverse-seeing* conditions are another subset of the baseline *seeing* ones. Here the environment is considered to be both dark *and risky* in some way – for example when visibility is impaired due to heavy rain at night. Most vehicles *will* differentiate between seeing and adverse-seeing. Those that do will employ head or full (head + tail) fog lights, in addition to headlights, while in this context.

Being-seen

Being-seen conditions start when *seeing* conditions (including their two subsets) end; the two are mutually-exclusive. Such a context is typically entered at times when the sun glare warrants usage of lights for signaling presence to other traffic participants, as opposed to illuminating the environment; e.g. after sunrise and until the sun has ascended above a certain subjective threshold, ceasing to impair the driver's vision. Most AI vehicles, however, particularly those equipped with DRL, will lack such safety-consciousness, and will instead transition to an *off* context once seeing conditions no longer apply. Vehicles that *do* have such sensitivities, on the other hand, will use either a) standstill lights, or

b) standstill and head fog lights, or c) headlights, or d) headlights and head fog lights under being-seen conditions.

Off

This context holds whenever neither *seeing* nor *being-seen* conditions apply. All vehicles will then unconditionally switch off any exterior lights previously in use.

Temporarily-off

This context is entered when vehicles are stopped at prolonged[1], typically terminal stops, where told by OMSI to turn the engine off. Depending on whether the engine is indeed deactivated, as addressed in a later section, two sub-cases can manifest:

- If the engine indeed gets turned off, most vehicles will be satisfied by the automatic reduction of headlights (if previously active) to standstill-only, and will opt not to undertake any further manual exterior lighting reduction. Some more environmentally-friendly simulated drivers will however manually completely deactivate all exterior lights, until departure time.
- Otherwise, that is, if the engine for some reason is left running, vehicles will unconditionally manually reduce exterior lights to standstill (or standstill + fog lights) until departure time. Of course some will still eagerly opt to disable lights altogether.

Closing notes

- For most vehicles, *seeing* conditions will become effective *before* the time OMSI instructs them to turn their lights on; hardly any vehicles will wait until it gets really dark before deciding lights are needed.
- For most vehicles, *being-seen* conditions, if at all applicable, end at some point before midday[2]. For some vehicles, nonetheless, being-seen conditions will *always* apply when seeing conditions do not. Such vehicles will as a consequence never completely disable their exterior lights (with the potential exception of entering a *temporarily-off* context).
- *Temporarily-off* conditions are more likely to manifest during the day, particularly when the baseline / overarching context is that of *being-seen*.
- Light sources preferred under one context affect those preferred under other contexts; for example, vehicles relying on head fog lights for *being-seen* are more likely to also rely on them for the purpose of *seeing*.
- Perception – the classification of conditions to the aforementioned contexts / labels – and associated light source usage preferences are *static*[3].

---

1 For the purposes of this document, a prolonged stop is either a) a trip’s first, second, or last stop entry, or b) any other stop where OMSI instructs the vehicle to turn its engine off.

2 This does of course vary depending on the season, due to the sun remaining at lower angles most of the day throughout the winter.

3 *Static* AI vehicle attributes are those that yield the same outcome under the same primitive, objective conditions (e.g. temperature). They are initialized once, when the vehicle is spawned or otherwise first assigned to the AI, and remain constant thereafter, until either a) OMSI is terminated or a different map gets loaded, or b) the user takes over control of the vehicle and subsequently re-assigns it to the AI (in which case a new AI vehicle/driver profile gets generated, since it might be a different virtual colleague then taking over). Static attributes are easier to implement and yield better performance, at the cost of reduced realism.

- Perceptual and adjustment “latency” / tolerance – how much time will elapse between a) primitive conditions changing, and b) the vehicle acknowledging the fact that the context changed, and ultimately c) the vehicle reacting to the contextual change by adjusting the state of its lights – is *dynamic*[3]. Generally vehicles will be *more tolerant* of “good” conditions and *less tolerant* of “bad” conditions, that is, they will tend to be *faster* at responding to deteriorating conditions, and *slower* at responding to improving conditions.

2.2. Interior lighting

Activation and deactivation

Most vehicles will only decide to switch their interior lights on when the environment starts becoming *visibly* darker, as opposed to[4] exterior lights, which may be activated well before that time. The conditions for activation of interior lighting are in most cases a “worse” subset of the exterior lighting *seeing* conditions discussed previously. Vehicles that always use exterior lights for *being-seen*, i.e., vehicles that never deactivate their exterior lighting, no matter how favorable the environment, pose a notable exception, however: Some of them will never deactivate their interior lighting either[5].

Dimmed vs full mode

Typical city buses offer two interior lighting modes: A full and a dimmed one, with the former usually translating to all interior light sources being activated, and the latter translating to just some sources (usually excluding those directly behind the driver cubicle) being activated.

Under conditions of low brightness[6] most vehicles will employ the dimmed mode in order to reduce reflection glare on the windshield impairing their vision. Such vehicles will only transition their interior lighting to full mode at times when they would normally disable interior lights, were the weather conditions better; e.g., on cloudy days. As always there are exceptional minorities: Some vehicles will always favor the dimmed mode, and even fewer will always favor the full mode.

Dynamic attributes on the other hand are those that may yield a variety of outcomes under the same primitive objective conditions. Inversely to static attributes, they require greater implementation effort and yield worse performance, but benefit realism.

The question of which approach to favor is dealt with on a case-by-case basis. The general criterion relied on is whether the particular AI perceptual or behavioral aspect only occurs infrequently, and is thus less susceptible to being scrutinized as overly predictable or “mechanical” by the observing user. If the answer is positive, then static attributes are used; otherwise dynamic attributes.

4 Because most real-world drivers appear to value road safety more than passenger comfort.

5 Because an actual driver choosing to unconditionally leave exterior lighting on during the day is somewhat more likely to forget to disable interior lighting on time.

6 Which means “most of the time for most vehicles”, given that the majority of vehicles will not use interior lights unless the environment is to some extent dark.

Temporary deactivation on service trips

A fair percentage of vehicles will opt for disabling interior lighting when undergoing a service trip. Most such vehicles will only do so under really dark conditions though, as well as early[7] in the morning and late in the evening.

Temporary deactivation at prolonged stops

A very small vehicle minority will disable[8] interior lights at stops when the engine is to be turned off, according to OMSI, as long as no passengers remain on board.

Closing notes

- For most vehicles, the activation conditions for interior lights will become effective at *roughly the same* time OMSI instructs them to turn them on. In contrast to exterior lighting seeing conditions, however, it is not entirely uncommon to also witness vehicles choosing not to activate interior lights before *actual sunset*.
- As with exterior lighting, perception and preferences are static, while adjustment delays are dynamic. Furthermore, adjustment delays are longer when interior lights are to be switched off[9] than when they are to be switched on.

2.3. Doors

Opening

Doors can be opened *lazily* or *eagerly*.

Lazy opening of a particular door occurs when at least one AI passenger requests entry or exit via that door.

7 Early vs. late determined on the basis of actual 24-hour time, not solar elevation angle. During those periods, on most maps, service trips will tend to last longer and be more frequent.

8 Perhaps for the driver to take a nap or simply enjoy some relaxing, undisturbed alone-time; or maybe to just conserve battery load.

9 Exception: *Dimming* (full→dimmed) occurs faster than *un-dimming* (dimmed→full), due to the glare irritating the driver.

Eager opening of one or multiple doors occurs when the vehicle a) is at a prolonged stop and OMSI instructs it to turn its engine off, and b) deems the environment as appropriate[10, 11] for door opening.

Closing

Doors can likewise be closed *lazily* or *eagerly*.

Lazy closing of all doors occurs when it is time to depart.

Eager closing of one or multiple doors occurs when either a) the environment is no longer perceived as appropriate[12] for those doors to be open, or b[13]) the current stop is non-prolonged and the doors appear not to be (anymore) in use by passengers boarding or disembarking.

Open front door departure

Rarely, a certain minority[14] of vehicles may depart from a stop without[15] closing the front door. This is most likely to happen when:

- The environmental temperature is borderline uncomfortably warm, yet not warm enough to use the A/C – according to the vehicle's perception.
- The weather is perceived as dry and clear, and it is daytime.
- The vehicle is undergoing a service trip or parking maneuver (see next section), or is at the last (pre-terminus) or first few stops of its trip.

---

10 Environmental perception influencing door and door wing locking, heating / cooling, window and hatch, and at some degree sun blind and wiper state, is driven as follows:
- The weather temperature is subjectively interpreted as being *coldest*, *cold*, *chilly*, *ideal*, *warm*, *hot*, or *hottest*.
- The solar elevation angle is subjectively interpreted as *day* or *night*.
- The precipitation rate, if any, is subjectively interpreted as *dry* or *wet*.

The interpretation is static, while the acknowledgment delay of perceptual change is dynamic. Conditions that are "worse than before" tend to be registered faster than conditions that are "better than before". For the algorithm of what constitutes a "better" vs. a "worse" transition, refer to the implementation of `ai_pre_cockpit.osc:ai_pre_cockpit_env`.

11 In general the likelihood of eager door opening occurring is proportional to temperature, daytime, and dryness. There is however an implicit personality trait involved as well, resulting in some vehicles being overall more inclined to unconditionally open their doors than others.

12 Note that environmental "inappropriateness" here does not necessarily translate to the complement of "appropriateness"; the distinction is asymmetrical. A vehicle may, upon *lazy* opening, and under certain conditions that are neither deemed as "really warm" nor "really cold", choose the middle ground of "lazily-doing-nothing", i.e., leaving the door open, even though it wouldn't have opened it in the first place if no boarding or disembarking passengers had requested its opening.

13 This allows vehicles to appear to close doors "asynchronously", as if in a hurry to depart, when no longer needed open, prior to OMSI telling them it is time to depart (e.g. closing front door while back door(s) still used by disembarking passengers). Due to OMSI limitations this feature may not always work at intended though; refer to the implementation of `ai_pre_cockpit.osc:ai_pre_cockpit_stop` for a detailed view of the pros and cons.

14 And even for that minority this is a rare sight, due to generally being illegal and considered "a thing of the past".

15 This case is distinct from the case of departing (releasing the stop brake) prior to the front door having (fully) closed.

- There are hardly any passengers on board (static criterion), and in particular none that are either standing or sitting[16] in the first row.

Naturally the front door must also be decoupled[17, 18] from the stop brake for the preference to be applicable. The front door is then left open until a) the vehicle arrives at the next scheduled stop, or b) the environmental constraints no longer hold.

Final notes

- Eager closing due to environmental constraints is static behavior.
- Eager opening due to environmental conditions, as well as eager closing due to lack of passenger flow and timetable constraints, are dynamic.
- Lazy opening and closing are inherently dynamic, since they are mandated by the game.
- Door adjustment delays are as usual dynamic.
- Lazy opening occurs (much) faster than eager opening.
- Closing generally takes longer at prolonged stops. It also takes longer depending on how full the vehicle is (evaluated statically by each vehicle), whether there are standing passengers or passengers seated close to the front door[16, 19], or passengers still walking toward their seat of preference at the original time of departure.
- The decision to leave the front door open at departure is dynamic.

2.4. Stop brake switch

Activation at scheduled stops

Upon arrival at a scheduled stop, vehicles will manually engage the stop brake via the stop brake ("20-h") switch, if either:

- They intend to open their front, yet not their back door(s), and the front door is *not coupled* to the stop brake; or
- The stop is non-prolonged, and/or the engine is, according to OMSI, *not*[20] to be turned off, and the vehicle remains stopped for longer than a few seconds.

Activation at unscheduled stops

A vehicle is said to be at an *unscheduled stop* whenever, for whatever reason, it has to slow down to a standstill without being located within the docking distance of a stop cube on its schedule; e.g. when waiting on a red traffic light. The vehicle may then toggle its stop brake switch if stopped for long enough. How long that might take, and with what frequency the stop brake will be employed, depends

16 Only for *user-focused* AI vehicles; OMSI does not communicate seat occupation status to *unfocused* AI vehicles, and hence this feature will not work as intended there.

17 That is, opening of the front door will not automatically cause stop brake engagement.

18 See also section on .cti-variables.

19 And hence greater care is needed by the driver to ensure the front door has fully closed prior to departure.

20 Otherwise the handbrake would too ultimately become engaged, mitigating the need to also engage the stop brake.

on subjective preference and random variation. Regardless the significance of the preference, there will be times when a vehicle does not engage the stop brake, no matter the unscheduled stop's duration.

Switch deactivation

Vehicles will reset the stop brake switch, if previously engaged, just before departing. At unscheduled stops they may also reset it earlier (they may quickly toggle it from "off" to "on" and back to "off").

Closing notes

Stop brake switch activation constitutes static behavior when occurring in the context of front door opening; no vehicle will ever open its front door without engaging its stop brake. All other cases of behavior are dynamic, as are adjustment delays.

2.5. Scheduled stop departure

Stop overlap pseudo-departure

On some maps there are stop cubes that are so close to one another that their configured docking distances overlap, effectively causing the same path segment to technically "belong to" two or more cubes. This manifests more frequently on terminal/initial cube pairs, such as at U-Bhf. Ruhleben on the standard Spandau map. Standard AI vehicles will in such cases "appear to depart"[21], only to instantly "appear to arrive" the very next instant, without actually having covered any distance in between.

Vehicles of this modification will a) detect this case, and b) communicate departure to OMSI, without c) altering their visible state[22].

Parking maneuvers

Sometimes a bus arrives at a (typically prolonged) stop, only to find part of its path blocked (typically by a preceding vehicle), causing it to stop (and "appear to arrive") prior to reaching the center of the cube's docking distance. When the path finally clears, yet departure time has not yet arrived, OMSI instructs that the vehicle depart *without* actually advancing its timetable to the next stop entry, causing the vehicle to merely correct its parking position (hence the naming of such behavior). Modified vehicles will attempt to detect this case and depart much faster under such conditions, to avoid blocking succeeding traffic for too long.

21 By closing doors, turning their engine back on, etc..

22 Put otherwise, whenever the vehicle arrives at stop *n*, and that stop's corresponding cube overlaps with stops *(n+1)*, *(n+2)*, and *(n+3)*, the vehicle will not "appear to depart" until told by OMSI that departure time from stop *(n+3)* has arrived.

Other factors affecting departure delay

Under regular, non-parking-maneuver-style departures, a number of factors influence the overall delay:

- The time of departure is significantly delayed at prolonged[23] stops.
- The time of departure is significantly delayed at stops when the engine was turned off[24].
- The longer doors take to close (see above section), the longer departure gets delayed.
- The departure delay is lower when doors were not opened at all at the particular stop, or when all were closed eagerly before the time of departure.

2.6. Electrics, engine, transmission, and handbrake

All three of these functions are handled by the enhanced buses of this modification in more or less the same fashion as standard AI buses. The primary difference is proper sequencing and random delay of adjustment. Note that all of the behaviors described in this section are static.

Electrics

Electrics are generally dependent on whether the engine is running – when it is (ignition key in position 2 or (momentarily) 3), the electrics are fully on, otherwise (ignition key in position 1) they are fully or partially on[25]. When the engine is not running, however, the AI may need to temporarily fully activate the electrics (by rotating the ignition key to position 2, without proceeding to position 3 to ignite the engine), if the driver window or the hatches require adjustment[26] while stopped. After adjustment of the driver window and/or the hatches has completed, electrics are reset to partial / constrained operational state (ignition key rotated back to position 1).

Engine

The engine is turned off only when so instructed by the game[27]; it is then not reignited until actual departure[28]. Furthermore, the engine is not turned off if the weather is perceived by the vehicle either as extremely[29] cold or extremely hot.

23 Particularly in the case of initial stops, passengers might not be able to reach the vehicle in time otherwise (due to its destination display failing to update soon enough, and/or the docking distance being too great and passengers walking at snail’s pace, and/or OMSI notoriously tending to spawn additional, “late-arriving” passengers, while at the same time instructing the vehicle to depart).

24 To allow non-instant and hence more realistic adjustment of engine, transmission mode, and handbrake.

25 Depending on vehicle configuration, in terms of individual functions’ constraints on power availability.

26 The driver window and/or the hatches may require that the ignition key be at position 2 or greater in order to become operable (refer to .cti-variable reference). Only then will the AI rotate the ignition key prior to using those functions.

27 Generally this is the case when the effective (scheduled time of departure - actual time of arrival) stop duration is greater than 1 minute.

28 Accounting for stop overlapping, (explained previously).

29 “Extremely” according to the vehicle’s viewpoint, as always; one vehicle might perceive a temperature of 28 degrees as severely hot, while another might perceive the same value as comfortably warm.

Transmission

Transmission is set to “N” whenever the game instructs that the engine be turned off, regardless of whether the engine will be turned off in practice; it is then not reset to “D” until actual departure, and not before the engine has been turned back on[30].

Handbrake

The handbrake is trivially engaged whenever the vehicle arrives at a prolonged stop and is told to turn its engine off, and disengaged at departure time.

Adjustment sequences

Typically, upon prolonged stop arrival, vehicles will first a) engage the handbrake, then b) switch transmission mode to “N”, and finally c) turn the engine off, if applicable. At departure the typical adjustment order is the inverse, i.e., a) reigniting the engine, then b) switching transmission to “D”, and finally c) disengaging the handbrake. Besides the aforementioned constraints governing engine and transmission dependency, however, adjustment delays are orthogonal to one another, and hence it is not uncommon to witness vehicles performing these three actions in any otherwise permissible order.

2.7. Front door wing locking

A fair amount of vehicles that have a wing locking function (switch) installed for the front door, will employ that function when the environmental conditions are perceived as uncomfortably cold (particularly at nighttime, as well as under wet conditions) or uncomfortably warm[31] (particularly at daytime). Furthermore, most of these vehicles will be inclined to lock the front, rather than the rear wing.

Under hot conditions and during prolonged stops when the engine is turned off, the wing lock will be reset, such that both wings of the front door can (eagerly) be opened for the duration of the stop.

Door wing locking preferences are static. Adjustment delays are dynamic, for the most part[32].

---

30 This is significant when the need for an unanticipated parking maneuver adjustment arises: Even if the vehicle has not had enough time to shut its engine down and switch transmission mode to “N” since its arrival, it will always gracefully ensure the proper, in terms of ordering, adjustment of both functions back to their initial state, that is, a) if the engine was turned off, but transmission is still in “D”, switching transmission to “N” first, then b) reigniting the engine, and lastly c) re-switching transmission to “D”.

31 Provided the A/C is in use for cooling.

32 Exception: When the front door is to be opened at a time when adjustment of the wing locking switch is pending, the latter is “lazily” rescheduled to occur *exactly* before the former (the driver wants to open the door and just so “remembers” they ought to flip the wing locking switch first).

2.8. Driver window

The degree at which the driver window is opened is dependent on random, per-vehicle preference, for each of the sets of common perception criteria[10] discussed previously for door and door wing locking. Generally the window will be most lowered under a perception matching { “warm”, “day”, and “dry” }. Under wet conditions, the window will usually be closed, as too will be the case under conditions warranting usage of the A/C for cooling.

As with door wing locking, under conditions perceived as uncomfortably warm, at prolonged stops when the engine is to be turned off, the window will be opened significantly and remain open until the time of departure.

The preferred window state for each set of conditions is static. The adjustment delay is dynamic.

2.9. Hatches

Hatch state[33] is driven in exactly the same manner as driver window state. The condition - preference pairs, however, are orthogonal to those governing the driver window. Most vehicles will be more conservative[34] with their use of hatches than their use of the driver window. Additionally, as opposed to the driver window, hatches are not temporarily opened under uncomfortably warm conditions, at prolonged stops when the engine is turned off.

2.10. Passenger windows

Both the simulated AI driver, as well as AI passengers[35] can manipulate[36] the state of passenger windows. The preferred setting per window is determined similarly (albeit once again remaining orthogonal) to the manner in which the preferred state of the driver window and the hatches are. It is important to note that there are two preferences involved – that of the driver and that of the passenger(s), if any, seated close to each window – which need not necessarily align.

33 Due to the range of currently supported vehicles being limited to the “CVAG 72” (a Citaro), all hatches can at present only be in two states – open (backwardly) or closed. This function may be further refined, should support for different vehicle types be introduced in the future.

34 That is, fewer vehicles will open hatches under sub-optimal conditions, as opposed to the driver window, which may well remain at least somewhat open under even relatively cold or hot conditions.

35 Only in user-focused vehicles, whether AI- or user-controlled, for reasons explained previously.

36 Assuming the vehicle *does have* folding passenger windows that can be opened. See also section on .cti-variables.

AI driver preferences

AI drivers will typically want one or two windows in the back[37] of the vehicle to be open under neutral-to-cold temperature ranges, gradually increasing their number until the temperature reaches borderline uncomfortable warm levels. Under uncomfortably warm or wet conditions, they will usually want the windows shut. As an additional measure, most drivers will tend to lock the passenger windows under the extreme ends (cold or hot) of the temperature spectrum; some drivers will however always lock them when not in use.

AI driver-conducted window adjustment

AI drivers will only ever alter the state of passenger windows when a) at a prolonged stop, when the engine ought to be turned off according to OMSI, and b) no passengers are on board[38]. This case of vehicle adjustment is special, in the sense that there is no single switch that can just be flipped to instantly carry out the adjustment; instead, it must appear[39] as if the driver were to exit their cubicle, walk to each window, and adjust its state.

AI passenger preferences

Passengers act in groups, each group comprising the occupied seats[40] that are adjacent to each folding window. A group[41] is formed when the first passenger arrives. Subsequently-arriving passengers attach themselves to previously-formed groups. A group dissolves and instantly reassembles[42] when either a) too many newcomers joined in, or b) when too many of the group's founding passengers have disembarked. A group simply dissolves when all its members have left.

In terms of actual preference, passengers desire the same[43] thing as the driver: Open the window when the environment is perceived as "better-outside-than-inside", and close it otherwise. Unlike the driver, passengers have a third option – to "do nothing", implicitly adopting the driver's preference regarding their window as "fair-enough". Also, unlike the driver, passengers will want to open the windows when outside conditions are uncomfortably warm and the engine is not running (and hence nor the A/C). Of course passengers' behavior is also constrained by the driver having potentially previously decided to lock their window.

---

37 Likely due to the temperature in the rear being higher (due to engine conduction); and/or to limit draft between passenger windows and their own window, under lower temperatures.

38 This ensures there is no "war" between conflicting driver - passenger viewpoints resulting in endless "cycling" – one party opening the window and the other closing it shortly thereafter.

39 Temporally-wise at least, since we can't animate the actual driver figure.

40 Sitting or standing places declared at the `passengercabin.cfg` level, that is.

41 Refer to commentary in `ai_pre_cockpit.osc:ai_pre_cockpit_passengers` for a more detailed discussion on group dynamics.

42 With potentially different opinions, i.e., preferred window state.

43 That is to say, the logic is the same, while the concrete perception may differ.

Closing notes

The driver's preferences are static. Group preferences are static throughout each group's lifetime. Adjustment delays are dynamic.

2.11. Heating and cooling

For heating purposes under a somewhat cold environment, vehicles will rely on either the A/C or the cabin heaters. For heating purposes under a significantly cold environment, vehicles will always use their more effective cabin heaters. Regardless this distinction, vehicles will also increase the driver-specific output temperature and fan speed, while leaving the driver A/C function in passive, engine-assisted mode, while in a heating context.

For cooling purposes under hot, as well as environmental conditions that are both warm and wet, vehicles will rely on the A/C (both passenger and driver functions), adjusting the driver-specific output temperature to a medium-to-cool setting, as well as increasing the driver-specific fan speed.

Lastly, under mostly-comfortable conditions, some vehicles may choose to use the driver A/C in "plain fan" mode, increasing the driver-specific fan speed, while retaining the lowest permissible output temperature, and without otherwise activating either of the two A/C functions.

Heating and cooling preferences[44] are static. Adjustment delays are dynamic.

2.12. Sun blinds

Sun blind state depends on the perceived solar elevation angle[45], as well as the common temperature ranges, the common dryness/wetness dichotomy, and the primitive level of environmental brightness[46]. The probability that a vehicle drag their sun blind(s) down is greatest around midday, particularly when the environment is additionally perceived as warm, dry, and bright. Those vehicles that choose to employ their sun blind(s) in the morning or evening hours, however, will tend to also drag it down further during those periods than at midday. Furthermore, the likelihood of a vehicle using its main (windshield) sun blind is greater than the likelihood of it using its side (driver window) sun blind (where installed).

44 Statically-variable preferences: A/C vs. cabin heaters for heating purposes; driver A/C temperature and fan speed combinations for heating and cooling; usage of driver A/C as "plain fan" (and, if so, the fan speed) under generally favorable conditions. The remainder of the aspects discussed are not really preferences, in the sense that they do not vary between vehicles.

45 Here the angle is subjectively classified as either *night*, *morning/evening*, or *midday*.

46 Here brightness is reduced down to a simple dichotomy of *(not) bright*.

Dragging either sun blind down is only permissible[47] when the vehicle is halted at either a scheduled or unscheduled stop, and no other adjustments are scheduled to concurrently occur. Sun blind retraction is allowed as long as no other adjustments are pending.

The preferred position of each sun blind under each set of conditions is static. Adjustment delays are dynamic.

2.13. Wipers

Wiper state depends on the precipitation rate[48] (obviously). Furthermore, it depends on the precipitation type, with rain generally being perceived as “more intense” / “more severe” than snowfall. Lastly, it depends on the common day/night dichotomy, along with primitive environmental brightness, with *brighter* conditions generally translating to precipitation being perceived as “*more* intense”[49].

Only some vehicles will ever employ the “fast” wiper operational mode; most will stop at the “continuously on” speed, regardless the precipitation intensity.

When at prolonged stops, when told by the game to turn their engine off, most vehicles will deactivate their wipers until departure. Some will instead leave them running[50], possibly at a lower mode of operation.

The preferred wiper mode under each set of conditions is static. Adjustment delays are dynamic, and shorter when wipers are to be enabled or transitioned to a greater speed, than when they are to be transitioned to a lower speed or disabled.

2.14. Indicators

The main contexts of indicator adjustment[51] currently implemented are: a) Adjustment when arriving at a stop; b) adjustment when departing from a stop; and c) adjustment en route. There are also the

47 Because the driver would generally need both their hands free to drag down a blind, as well as be able to momentarily leave their seat (mandating that at least the stop brake be active). Electric sun blinds are currently unsupported.

48 For wiper-centric perception, precipitation specifically is classified as *very light* (wipers only manually toggled on/off when windshield gets wet enough), *light* (wipers set to “interval”), *moderate* (wipers set to “continuously on”), and *high* (wipers set to “continuously on” or “fast”, depending on per-vehicle preference).

49 Because particularly under rainfall, a wet windshield is more “annoying” / vision-impairing when reflecting ambient light, which is of course of greater intensity during the day.

50 Whether the wipers actually stay on then depends on whether a) the engine happens to remain on, or otherwise b) whether the wipers remain operable under reduced electrics state (ignition key at position 1). Currently no special accommodations are made for (b), i.e., electrics will not be turned fully on (ignition key at position 2) solely to enable wiper operation in AI vehicles desiring it.

51 Indicator adjustment here refers to the action of a) enabling (off→left/right), b) disabling (left/right→off), or c) inverting (left↔right) the indicator mode.

secondary cases of d) deactivating indicators at prolonged stops, when the game requests that the engine be shut down, and e) passively having the indicators deactivated due to having exceeded the “critical” steering angle towards the corresponding direction.

In all cases it is possible that the simulated AI driver “forget” / “neglect” to adjust their indicator mode. It is also possible that the adjustment is carried out after a shorter or longer delay.

On a least- to most-likely-to-forget scale, the various cases can be ordered as follows:

- (Least likely) (a) At scheduled stop arrival.
- (c) En route, with activation or adjustment being less likely to be neglected than (timely) deactivation.
- (b) At scheduled stop departure[52], most vehicles will most of the time opt to *invert* their indicators[53]; some may lazily just turn them off instead.
- (e) In the event of automatic indicator deactivation due to steering, vehicles will be somewhat prone to lazily leaving them off, even if not yet having completed that turn, or having exited that junction.
- (Most likely) (a) Deactivation[54] at prolonged / engine-off stops.

When indicator adjustment is forgotten, the vehicle will most likely remember[55] to correct them the next time a change of indicator setting gets communicated to it by the game.

Indicator “forgetfulness” and adjustment delays are dynamic. While certain vehicles will on average be more prone to forgetting to (timely) adjust their indicators, none are guaranteed to *never* or *always* act in the same way.

52 Note that OMSI’s (the path segment’s, to be more precise) opinion on the matter is effectively ignored in this case; as long as the path segment *before* the stop mandated the proper behavior, the behavior at departure will be automatically and correctly deduced. Exception: When the path segment *just after* a stop mandates anything other than indicator deactivation, its instruction will still be obeyed by vehicles (not taking forgetfulness into account).

53 In other words, if they indicated right at arrival, they will indicate left at departure.

54 Naturally, in vehicles where indicators are only functional when electrics are fully activated (ignition key at position 2 or greater), forgetful behavior will not manifest, unless the engine is left running at that stop.

55 Exception: When there are several adjacent path segments that all mandate the same indicator setting (e.g., two subsequent right turns), and the vehicle forgets to adjust its indicators the first time around, it will also keep forgetting their adjustment every subsequent time, until finally told to deactivate them. This is fine as long as all path segments represent *the same* junction / curve / turn – when this is not the case, i.e., when such segments stand for *consecutive, yet independent* junctions / curves / turns, repetitive neglectful behavior exhibited by vehicles will appear highly unrealistic. Unfortunately this is a known limitation of the current algorithm.

2.15. Non-functional requirements

Resilience to use interference

With the exception of a) opening the folding passenger windows, if present and unlocked, and b) pushing the stop request buttons, no other[56] cockpit triggers / mouse events are made available to the observing user when the vehicle is AI-controlled. Interfering with the AI work-flow and causing the vehicle to remain stuck indefinitely should therefore be impossible[57].

Resilience to OMSI interference

OMSI is obnoxious at de-/re-spawning/teleporting AI buses at will, without ever explicitly notifying them about the change in context. Significant measures have been taken to ensure that vehicles can detect such anomalous transitions and continue operation as gracefully[58, 59] as possible.

Deactivation of unsupported functions

When the AI takes over a previously user-controlled vehicle, it will deactivate or otherwise invalidate any functions it does not "natively understand"[60], thereby effectively preventing non-aesthetically-pleasing or erroneous side effects.

---

56 With the exception of still being able to affect the steering wheel, as well as the throttle and brake pedals via mouse or gaming controller, which is unintentional – OMSI appears to use its own internal "triggers" for such primitive input.

57 Exceptional edge cases: a) When crashing a user- into an AI-controlled bus; b) when allowing the AI to take over a severely damaged vehicle.

58 For example, suppose an AI vehicle was last active at a prolonged stop, with a shut-down engine and open doors. Normally that vehicle would require a significant amount of time – let's say half a minute on average – to depart in a "human-like" fashion, as explained in previous sections. If suddenly teleported to the middle of a busy junction though, the vehicle would have to resume operation *instantly*, such that the user can't notice the discrepancy – not even implicitly due to other AI traffic being blocked.

59 As a second example, suppose an AI vehicle was last active at a point in time where the weather changed from "rainy" to "perfectly sunny". As explained in previous sections, after a bit of time the observer would expect that this very vehicle would have undergone certain changes in state, e.g. having switched its lights off or having opened some windows. Sadly, the notion of time isn't continuous for AI vehicles – they only "exist" for as long as they remain on the same loaded tiles as their observing user. Therefore, adjustments of minutes would likely take hours to occur, or fail to ever manifest at all. To compensate, AI vehicles have been made aware of such "temporal anomalies", so as to instantly adjust their state if need be, and such that their observer can't tell the difference, i.e., that the particular AI vehicle was only loaded for a total of a minute and a half over the span of the 3 hours the user spent driving their own bus across the map.

60 For example the AI is capable of deactivating the retarder switch, detect and cancel-out out a transmission in "R" (by switching back to "N"), drag the driver cubicle door / ramp / doors opened via emergency valves back closed, etc..

3. Other significant changes

3.1. Lighting

The lighting subsystem has undergone significant refinement as part of this work.

Almost all[61] light sources, both on the exterior and interior, up to and including marker lights and the lights of the rear license plate, have become independently operable in one way or another. This enables individual light sources to have and/or produce more realistic illumination effects, to individually affect circuit voltage, and to individually become defunct under different conditions.

Light sources can already be defunct when a vehicle first gets spawned, including vehicles initially spawned in AI mode. Light sources can also become defunct subsequently, either due to naturally reaching their end-of-life, or due to receiving collision damage. The overall probability and frequency of either event occurring depends on a) the type[62] of source, b) its significance[63] and where[64] exactly it is installed on the vehicle's body, c) the vehicle's age[65], d) the user's maintenance setting[66], and of course e) if applicable, collision severity. It should also be noted that even when a collision does not produce visibly damaged lights, lights in the proximity of the collision's coordinates will still be *strained* as an implicit result, negatively affecting their lifetime expectancy.

Last but not least, the circuit-level constraints governing activation of light source (groupings) have been revisited as well. Particularly when the engine is not running and the battery is weak, certain non-essential lights will be *throttled* before the battery depletes, as a precautionary measure to conserve energy.

Some further changes are enumerated in the changelog section.

---

61 With the exception of matrix-integrated light sources, as well as dashboard lighting and function indicators, which have not yet been integrated with the revamped lighting subsystem.

62 For example, the lifetime of a LED source is generally greater than that of a fluorescent, in turn being generally greater than that of a halogen one.

63 For example, during maintenance, a headlight is more likely to be replaced on time (prior to burning out) than a marker light.

64 Both in terms of significance and collision susceptibility. For example, an orange marker light, *A*, located towards the middle of the vehicle, is statistically less prone to being damaged as a result of "micro-collisions" (pavement "scratches") than another orange marker, *B*, located closer to the vehicle's head or tail. Furthermore, *A* might be considered of greater contributing value to safety, and maintenance will be more likely to identify and replace it on time. For both reasons, the user will generally be more likely to witness light *B* in a defunct state, than light *A*.

65 Older vehicles will have batteries and other components of their electrical circuitry in a sub-optimal state compared to younger vehicles, in turn (slightly) negatively impacting performance (lifetime) of connected light sources.

66 Only for user-spawned vehicles, and only while user-controlled; for AI-spawned/-controlled vehicles OMSI does not communicate

3.2. Heating and cooling

The implementation of the heating and cooling subsystem has been replaced with a different one, yielding a number of improvements. This section enumerates some key differences between the current and original implementation.

All heating and cooling functions now undergo non-instant adjustment, i.e., feature “inertia”. The length of an adjustment period depends on the kind of function, the difference between the current and target temperature and humidity, the target state (function activation vs. deactivation), and random variation. Simply put, the user should generally anticipate that flipping a switch will *not* produce a visible change in sound, temperature and humidity until several seconds or even minutes have elapsed.

A new passive function, coined “greenhouse-like effect”, has been introduced. Depending on insolation and particularly during the summer, the inside temperature can rise significantly above the ambient (outside) temperature. This effect can to some extent be mitigated by keeping windows, hatches, and doors open as much as possible, or using the A/C functions.

The passenger A/C function has become more effective in cooling[67] mode, while its effectiveness has been somewhat decreased in heating mode. For heating purposes, the cabin heaters remain a much more effective alternative. As usual, the passenger A/C’s output and allowed (minimum, maximum) target temperatures cannot be altered by the user[68]. Its fan speed will automatically adjust depending on the difference between the current and target cabin temperature.

The driver A/C function[69] has been greatly reduced in effectiveness – after all it is supposed to heat or cool the driver cubicle and not the entirety of the vehicle. Thus, even at maximum temperature and fan settings, the driver A/C will not appear to significantly influence the overall[70] vehicle temperature. The driver A/C is totally ineffective[71] for heating purposes – leaving it in “passive”, engine-assisted mode is a much better idea when intending to heat the driver’s place.

Both A/C functions are implicitly coupled to a humidity management ((de-)humidification) function. That function will automatically activate whenever a) any A/C function is active, and b) the cabin humidity drops below or exceeds comfortable[72] values. The user can also explicitly activate it via the

67 The A/C functions will by default perform heating when the temperature inside is below 19 degrees, and cooling when the temperature inside exceeds 22 degrees.

68 They can however be adjusted via the many constants in `uchill_consts.txt`.

69 Including the sub-function where it operates in engine-assisted, as opposed to A/C-based mode.

70 In a future version we might further emphasize the fact by simulating different “thermometers” for a) the driver cubicle and b) the reset of the cabin. This is a non-trivial exercise, however, and OMSI is anything but helpful in that regard.

71 Exception: For humidity management purposes (humidifying or dehumidifying the vehicle), leaving the driver A/C active actually *does* make sense.

72 Approximately 40~70%.

“heat-or-frost” button[73]. When humidity management is active, the “air origin” (fresh vs. recycled) mode of both A/C functions will transition to “recycled”.

Both A/C functions also feature an alternative, implicit, “passive” / “eco” mode, which is automatically chosen when the temperature on the outside is already (much) more comfortable than the temperature on the inside. The A/C functions will then effectively operate as plain fans, just allowing air from the outside to flow inside, until equilibrium is achieved.

Whenever a function is explicitly activated, yet constraints[74] for its activation do not hold, or has been implicitly / automatically enabled by the system, the dashboard indicator of the corresponding button or switch, where available, will blink.

For further details on heating and cooling functions, the user is encouraged to refer to the external documentation of the UCHill project.

73 Usually to be found on the classic 4-button side panel.

74 For example, A/C active but engine off; or humidity management active but humidity already within comfortable interval.

A. Trigger and .cti-variable reference

This section captures triggers and .cti-file-compatible variables that have either been newly-introduced or semantically altered as part of this work.

A.1. Triggers

`cp_fog_light_mode_increase_or_disable`
Increases or disables the vehicle's fog lights' mode, transitioning as shown below:
off [→ head fog lights] → head* + tail fog lights → off*
* If installed (depending on `lights_cti_has_head_fog_lights`)

`cp_fog_light_mode_decrease`
Decreases the vehicle's fog lights' mode until completely disabled, transitioning as shown below:
head + tail fog lights [→ head* fog lights] → off*
* If installed (depending on `lights_cti_has_head_fog_lights`)

`cp_passenger_window_[1|2]_[l|r]_lock`
Locks or unlocks the corresponding window (labeled beginning from the vehicle's front), *iff*:

- `cp_cti_has_openable_passenger_windows` is 1.
- The vehicle *is not* AI-controlled.
- The window is *closed*.

Note: When a *user vehicle* is initially spawned, its passenger windows are all always *locked*.

`cp_passenger_window_[1|2]_[l|r]_open`
Opens or closes the corresponding window (labeled beginning from the vehicle's front), *iff*:

- `cp_cti_has_openable_passenger_windows` is 1.
- The window is *unlocked*.

Note that, *unless locked*, AI passengers and/or, if applicable, the AI driver, may still adjust (open, close, or, additionally, in the case of the driver, lock) the window, regardless of your opinion.

`cp_not_aus_sw_toggle`
Activates or deactivates the emergency stop brake overriding ("NOT-AUS") switch. When activated:

- The stop brake will no longer automatically engage upon opening doors.
- Kneeling will no longer be possible; if the vehicle was previously knelt down, it will automatically rise back up, independently of door state.
- Manual stop brake engagement is still possible as usual (via the "20-h-switch").

`lights_ai_test_toggle`
Triggers the diagnostic mode of the AI lighting script, unless already running. Refer to Appendix C for details.

`kw_wipermode_[up|down]`
These triggers have been modified to behave equivalently to `cp_fog_light_mode_increase_or_disable` and `cp_fog_light_mode_decrease`.

A.2. .cti-variables

New .cti-variables are introduced on an as-needed basis in order to promote script reuse / portability across different CSB vehicles, in those cases where variation is trivial, i.e., when there are at most a handful of discrete cases that are pertinent to a particular functional aspect. It is not our intent to provide such variables for every conceivable case of configuration variation, particularly if such variation does not manifest in the scope of the CSB pack.

`lights_cti_has_head_fog_lights`

0. (Default) No head fog lights installed.
1. Head fog lights installed.

`lights_cti_has_drl`

0. (Default) No DRL installed.
1. DRL installed.

Note: DRL are currently only supported at the script level. To get an actual, "DRL-capable" vehicle, one must additionally, at the model.cfg level:

- Introduce the actual (2x) DRL light unit `[mesh]`es.
- Define accompanying visual effects via `[matl_nightmap]` and/or `[spotlight]` and/or `[light_enh[_2]]`, as appropriate, given the state of the `lights_drl_[l|r]_unit` variables.

`lights_cti_has_second_reverse_and_tail_fog_light`

0. (Default) No extra (left) reverse light and (right) tail fog light pair installed.
1. Extra (left) reverse light and (right) tail fog light pair installed.

Note: Supported only at script level – these extra light sources must still be integrated at the model.cfg level, given the state of the variables:

- `lights_rueckfahr_l_bulb` and/or `lights_rueckfahr_l_matl_mode` (for extra reverse light).
- `lights_nebelschluss_r_bulb` and/or `lights_nebelschluss_r_matl_mode` (for extra tail fog light).

`lights_cti_indicating_left_or_right_requires_full_elec`

0. (Default) Simple "left-or-right" indicator mode is operable when ignition key is at position 1 ("electrics on" / "ACC") or greater.
1. Simple "left-or-right" indicator mode is operable when ignition key is at position 2 ("pre-ignition" / "ON") or greater.

`lights_cti_int_lights_level_2_requires_running_engine`

0. (Default) Interior lighting level 2 (all fluorescent light tubes active) is only available when engine is running.
1. Interior lighting level 2 is available when engine is running, but also when ignition key is at position 2 ("pre-ignition" / "ON") or greater.

`cp_cti_has_door_wing_lock_switch`

0. (Default) No front door wing lock switch installed.
1. Front door wing lock switch installed.

`cp_cti_has_blinking_hazard_indicator_switch`

0. (Default) Hazard indicator dashboard switch *does not* blink when active.
1. Hazard indicator dashboard switch blinks when active.

`cp_cti_has_openable_passenger_windows`

0. (Default) Passenger windows *cannot* be interacted with. Neither the user nor the AI can use them.
1. Passenger windows can be interacted with ((un-)locked/opened/closed), by user and AI.

`cp_cti_hatches_require_full_elec`

0. (Default) Hatches are operable when ignition key is at position 1 (“electrics on” / “ACC”) or greater.
1. Hatches are operable when ignition key is at position 2 (“pre-ignition” / “ON”) or greater.

`cp_cti_driver_window_requires_full_elec`

0. (Default) Driver window is operable when ignition key is at position 1 (“electrics on” / “ACC”) or greater.
1. Driver window is operable when ignition key is at position 2 (“pre-ignition” / “ON”) or greater.

`door_cti_front_door_decoupled_from_stop_brake`

0. (Default) Stop brake is automatically engaged upon front door opening.
1. Stop brake *is not* automatically engaged upon front door opening.

Notes:

- This is a replacement for the traditional `Door_HST_Bremse_Aktiv` constant. One should however pay attention to the inverted semantics of its values.
- Setting this to 1 may (rarely) allow the AI to drive with open front doors, as explained in 2.3.

`wiper_cti_wipers_require_full_elec`

0. (Default) Wipers are operable when ignition key is at position 1 (“electrics on” / “ACC”) or greater.
1. Wipers are operable when ignition key is at position 2 (“pre-ignition” / “ON”) or greater.

`uchill_cti_driver_ac_maintenance_mode_requires_full_elec`

0. (Default) The driver A/C function becomes partially operable (fan-only / “maintenance mode”) when ignition key is at position 1 (“electrics on” / “ACC”) or greater.
1. The driver A/C function becomes partially operable when ignition key is at position 2 (“pre-ignition” / “ON”) or greater.

`uchill_cti_passenger_ac_maintenance_mode_requires_full_elec`

0. (Default) The passenger A/C function becomes partially operable (fan-only / “maintenance mode”) when ignition key is at position 1 (“electrics on” / “ACC”) or greater.
1. The passenger A/C function becomes partially operable when ignition key is at position 2 (“pre-ignition” / “ON”) or greater.

B. Known issues

This section is about aspects of the current implementation that are *blatantly wrong*, and certainly in need of fixing. Aspects that are *halfway correct*, yet could benefit from further refinement, won't be enumerated here – the list of ideas on future work is virtually endless.

<table>
<tr><th>ID</th><th colspan="4">Summary</th></tr>
<tr><td>B-1</td><td colspan="4">OMSI crashes upon quitting and situation change.</td></tr>
<tr><th>Resolution status</th><th>Severity</th><th>Occurrence frequency</th><th>Estimated effort</th><th>Priority</th></tr>
<tr><td>Unresolved</td><td>High</td><td>Low</td><td>Unknown</td><td>Low</td></tr>
<tr><th colspan="5">Symptoms</th></tr>
<tr><td colspan="5">After having played on a map where instances of the enhanced vehicles of this modification were spawned either due to the AI-list or directly by the user, the OMSI process fails to terminate normally (“app-hang” / “app-crash”). Trying to switch map / situation also causes the process to become unresponsive.</td></tr>
<tr><th colspan="5">Identified causes and resolution progress</th></tr>
<tr><td colspan="5">We have not been able to identify the cause yet (OMSI emits nothing relevant in its logfile, even under <code>-logall</code> / <code>-debug</code>). For all we know it might be due to something as trivial as a Unicode character in some configuration file where Unicode is not “tolerated” by OMSI, or something as hard-to-overcome as OMSI being unable to handle reading from too many script files.</td></tr>
<tr><th colspan="5">User-level workarounds</th></tr>
<tr><td colspan="5">Quit OMSI via Windows’ task manager, should it fail to gracefully terminate on its own, or hang during situation reload.</td></tr>
</table>

<table>
<tr><th>ID</th><th colspan="4">Summary</th></tr>
<tr><td>B-2</td><td colspan="4">OMSI raises error when enhanced vehicle assigned to unscheduled AI.</td></tr>
<tr><th>Resolution status</th><th>Severity</th><th>Occurrence frequency</th><th>Estimated effort</th><th>Priority</th></tr>
<tr><td>Cause identified</td><td>Moderate</td><td>Low</td><td>Moderate</td><td>Low</td></tr>
<tr><th colspan="5">Symptoms</th></tr>
<tr><td colspan="5">OMSI raises a “Zugriffsverletzung” when an enhanced vehicle not currently on a schedule is assigned to the AI.</td></tr>
<tr><th colspan="5">Identified causes and resolution progress</th></tr>
<tr><td colspan="5">This is typically the response of the game to a script attempting to query the timetable without first checking whether the vehicle indeed has one. The modification’s scripts have been double-checked to ensure they are not the cause. Preexisting / unmodified scripts making use of such functionality, like predominantly the scripts driving destination display, ticket printer, and IBIS devices, are thus likely candidates. The bug is also present in the original vehicles.

Due to the bug hardly ever manifesting – that is, unless the user provokes it, by e.g. registering enhanced vehicles as unscheduled AI traffic with the corresponding AI-list, or manually allowing the AI to take over a vehicle without a schedule in-game – we are disinclined to invest time into its resolution, as things currently stand. Also, if certain non-freely-distributable (copyrighted) matrix scripts, used by some of the CSB vehicles, turned out being responsible, we</td></tr>
</table>

would not be allowed to directly provide patches as part of this modification.
User-level workarounds
• Don’t assign your active vehicle to the AI, unless you’re currently on a schedule. • Don’t register any vehicles of this modification as *unscheduled* (e.g., as part of the “NormalCars” AI-group) with any AI-lists.

ID	Summary			
B-3	Humans crossing the vehicle’s path at a stop may prevent it from ever departing.			
Resolution status	**Severity**	**Occurrence frequency**	**Estimated effort**	**Priority**
Unresolved	Moderate	Low	Unknown	Low
Symptoms				
When, just before a vehicle’s departure from a scheduled stop, its path gets crossed by humans (crossing the street right in front of the stopped vehicle, for example), the humans and the vehicle may enter a “deadlock” kind of situation, where neither party is willing to move any further, both blocking each other.				
Identified causes and resolution progress				
There might be a link between this bug and our enhanced departure work-flow – vehicles telling OMSI they departed before actually having done so, primarily to circumvent the stop overlapping effect. Much experimentation and testing is needed for a definitive “diagnosis”.				
User-level workarounds				
In F4 camera mode, distance yourself from the affected vehicle by one or more tiles, depending on your configured number of loaded tiles. OMSI should then either de-/re-spawn the blocked vehicle, and/or de-spawn the troublesome humans.				

ID	Summary			
B-4	AI perception under rainy (+ windy) conditions is unreliable.			
Resolution status	**Severity**	**Occurrence frequency**	**Estimated effort**	**Priority**
Partially resolved	Low	Moderate	Unknown	Moderate
Symptoms				
Under environmental conditions that are rainy, and more so when additionally windy, AI vehicles may perpetually “cycle” (loop between two or more states / settings) the state of some or all of those functions / components / subsystems that depend on precipitation perception. For example vehicles may activate their fog lights when above a certain velocity, switch them off when happening to be driving slowly for some amount of time, and so on and so forth, in a never-ending cycle.				
Identified causes and resolution progress				
The primitive precipitation rate exposed by OMSI is, besides the chosen rain/snow intensity setting, dependent on velocity and wind direction (and consequently vehicle orientation) and speed. A proper resolution of this issue would entail somehow obtaining a normalized, stable precipitation rate representation. Sadly OMSI does not at present expose those variables to scripts, nor documents their relationship, effectively precluding resolution. Our present workaround is to simply have the AI acknowledge precipitation *increase* (much) *faster* than *decrease*, to				

<table>
<tr><td>somewhat limit cycling occurrence potential / frequency.</td></tr>
<tr><th>User-level workarounds</th></tr>
<tr><td>No real workarounds available – besides, of course, the Ostrich algorithm (“sticking your head in the sand and pretending there’s no problem”).</td></tr>
</table>

<table>
<tr><th>ID</th><th colspan="5">Summary</th></tr>
<tr><td>B-5</td><td colspan="5">AI cannot recognize cloudiness early in the morning / late in the evening.</td></tr>
<tr><th colspan="2">Resolution status</th><th>Severity</th><th>Occurrence frequency</th><th>Estimated effort</th><th>Priority</th></tr>
<tr><td colspan="2">Partially resolved</td><td>Low</td><td>Low</td><td>Unknown</td><td>Moderate</td></tr>
<tr><th colspan="6">Symptoms</th></tr>
<tr><td colspan="6">The subset of AI vehicles switching off exterior lighting early in the morning, particularly close to the time of sunrise, appear “indifferent” to the weather in terms of cloud cover. Real-world drivers, on the other hand, even if switching off lights early in the morning on clear days, would likely prolong their use of lights on cloudy days.</td></tr>
<tr><th colspan="6">Identified causes and resolution progress</th></tr>
<tr><td colspan="6">OMSI exposes neither the “cloud type” nor “brightness” weather settings per se to scripts, but rather a generic value of environmental brightness, that, while dependent on those settings, also reflects the level of illumination by street lights within the vehicle’s proximity (among other factors). Precisely due to street light skew, early in the morning, vehicles cannot reliably infer whether the day is going to be clear or cloudy (fortunately they can to some extent heuristically tell the difference later during the day).

Our current sub-optimal workaround is to a) dramatically limit the percentage of vehicles that switch off exterior lights close to sunrise, and b) have the remainder of vehicles (a rough 2%) behave independent of OMSI’s stance on brightness, as a contributing factor.</td></tr>
<tr><th colspan="6">User-level workarounds</th></tr>
<tr><td colspan="6">None available.</td></tr>
</table>

<table>
<tr><th>ID</th><th colspan="5">Summary</th></tr>
<tr><td>B-6</td><td colspan="5">AI cannot recover on its own from (severe) crash with user.</td></tr>
<tr><th colspan="2">Resolution status</th><th>Severity</th><th>Occurrence frequency</th><th>Estimated effort</th><th>Priority</th></tr>
<tr><td colspan="2">Cause identified</td><td>Moderate</td><td>Low</td><td>Moderate</td><td>Moderate</td></tr>
<tr><th colspan="6">Symptoms</th></tr>
<tr><td colspan="6">When a user - AI crash causes significant (electrics and/or engine and/or transmission) damage do the AI vehicle involved, the latter will stay damaged and “refuse” to behave normally, even after the user has “called the police”. The damaged vehicle has thus the potential to block traffic anywhere it gets subsequently re-spawned at by the game.</td></tr>
<tr><th colspan="6">Identified causes and resolution progress</th></tr>
<tr><td colspan="6">This is partly bug, partly feature. It is a feature because the original intent was for the AI to “deny” taking over a severely-damaged user vehicle, until the user fixes it (under standard scripts the AI vehicle would magically appear to drive on even without an operable engine, which we deemed plain ugly). It is a bug because we did not anticipate this particular variation (the user’s vehicle causing damage to another vehicle that is already AI-controlled) until beta-testing revealed that OMSI does not automatically repair crashed-into AI vehicles. A fix for this issue will likely be made available in a future release.</td></tr>
</table>

User-level workarounds
• Disable collisions. • Take over control of the damaged AI vehicle and “delete” / de-spawn it. • Take over control of the damaged AI vehicle and repair it.

<table>
<tr><th>ID</th><th colspan="4">Summary</th></tr>
<tr><td>B-7</td><td colspan="4">OMSI occasionally raises an error when user attempts to repair their vehicle.</td></tr>
<tr><th>Resolution status</th><th>Severity</th><th>Occurrence frequency</th><th>Estimated effort</th><th>Priority</th></tr>
<tr><td>Unresolved</td><td>High</td><td>Low</td><td>Unknown</td><td>Low</td></tr>
<tr><th colspan="5">Symptoms</th></tr>
<tr><td colspan="5">When the user triggers the native repair function of the game menu, OMSI may respond with a “Stack Überlauf” error. Repairs are still carried out. The same error might be issued anew when subsequently attempting to terminate the game or reloading / changing situation, likely leading to an OMSI process “hang”.</td></tr>
<tr><th colspan="5">Identified causes and resolution progress</th></tr>
<tr><td colspan="5">The bug manifests when repair times require (according to OMSI’s idiosyncratic nature) “too much” time – roughly above 200 minutes. Since there appears to be an internal OMSI issue involved here, little can be done – save for limiting repair duration, which isn’t a true solution either...</td></tr>
<tr><th colspan="5">User-level workarounds</th></tr>
<tr><td colspan="5">None available.</td></tr>
</table>

C. AI lighting subsystem diagnostic mode

Purpose

Because manual testing takes a painful amount of time, and has the potential of yielding seemingly erroneous results, the AI lighting script exposes a testing facility, through which multiple AI lighting "profiles" can be obtained via a single vehicle at short intervals, and analyzed in a semi-automated fashion.

Concretely, aided by the testing framework one may:

- Deduce how vehicles behave under the current conditions.
- Verify that the specified constraints hold.
- Verify that less frequent cases specified, that are hard to witness via manual testing, actually occur.
- Infer, in terms of logical consistency, the correctness of the specification itself.

Usage

1. Start OMSI.
2. Map the lights_ai_test_toggle trigger to some key (combination) of your input device.
3. Select a map and set the new situation's time and/or entry point to values that are *unlikely* to cause the game to load any buses, initially; e.g. a time past midnight and/or an entry point where scheduled AI traffic is infrequent.
4. Load the map.
5. Adjust the weather and time to your liking.
6. Change the time, if necessary, such that an AI bus gets spawned in your proximity. It is important that an (initially) AI-controlled bus be used.
7. Take over control of the vehicle while it is still moving, i.e., before it reaches any scheduled stop, otherwise observations may be inaccurate.
8. If the weather is wet, fully stop the vehicle, otherwise observations may be inaccurate.
9. Activate the testing mode via lights_ai_test_toggle. A sound should be emitted to confirm activation.
10. Wait until you hear that same sound again. A full testing round of generating and processing 10,000 lighting profiles typically takes 5~10 seconds.
11. Quick-save (Ctrl+S) the game.
12. Open <OMSI2>\maps\<map_you_were_on>\lastsn.osn. Copy everything between lights_ai_test_status and the line following lights_ai_test_ctx_ext_off_and_int_on_percent to a new file. Ensure that a) lights_ai_test_status has the value 0, and b) lights_ai_test_ctx_total_configs has a value greater than 0; otherwise testing likely never commenced in the first place.
13. Study the values of the variables recorded.

Core metrics reference

lights_ai_test_ctx_ext_off_percent[[_without]_drl]
Percentage (x/10000) of vehicles (equipped, or not equipped with DRL, or both combined) not using any exterior light source, under the current conditions.

lights_ai_test_ctx_standstill_percent[[_without]_drl]
Percentage of vehicles using standstill/position lights, under the current conditions.

lights_ai_test_ctx_standstill_and_front_fog_lights_percent[[_without]_drl]
Percentage of vehicles using standstill/position lights *and* front fog lights, under the current conditions.

lights_ai_test_ctx_headlights_percent[[_without]_drl]
Percentage of vehicles using headlights, under the current conditions.

lights_ai_test_ctx_headlights_and_front_fog_lights_percent[[_without]_drl]
Percentage of vehicles using headlights *and* front fog lights, under the current conditions.

lights_ai_test_ctx_headlights_and_full_fog_lights_percent_without_drl
Percentage of vehicles using headlights *and* full (front and tail) fog lights, under the current conditions.

lights_ai_test_ctx_ext_on_for_seeing_percent[[_without]_drl]
Percentage of vehicles using any exterior light source for the purpose of seeing, under the current conditions.

lights_ai_test_ctx_ext_on_for_seeing_reduced_percent[[_without]_drl]
Percentage of vehicles using any exterior light source for the purpose of reduced seeing, under the current conditions.

lights_ai_test_ctx_ext_on_for_seeing_adverse_percent[[_without]_drl]
Percentage of vehicles using any exterior light source for the purpose of adverse seeing, under the current conditions.

lights_ai_test_ctx_ext_on_for_being_seen_percent[[_without]_drl]
Percentage of vehicles using any exterior light source for the purpose of being seen, under the current conditions.

lights_ai_test_ctx_int_off_percent
Percentage of vehicles not using any interior light source, under the current conditions.

lights_ai_test_ctx_int_dimmed_percent
Percentage of vehicles using reduced/dimmed interior lighting, under the current conditions.

lights_ai_test_ctx_int_full_percent
Percentage of vehicles using full interior lighting, under the current conditions.

Tips

- Percentages are expressed in the interval [0, 1], 0 = 0%, 1 = 100%.
- Both the time and weather conditions are set in stone throughout test execution. Ensure they are correct *before* initiating the test mode.
- Using a "freshly-spawned" AI bus for each set of to-be-tested conditions is recommended.
- Variables with a name prefixed "_ctx_" refer to metrics that depend on the testing context (time, weather, etc.). Those should be the primary focus. The remainder of the `lights_ai_test_xxx` variables are mainly destined for debugging.
- The game *will* significantly lag while testing is in progress, this is normal – more or less the whole AI lighting script is run 100x/frame, to minimize the overall time each full testing round takes to complete.
- There is no guarantee of 100% accuracy in recorded metrics. Rounding errors after the first few decimal places are commonplace in OMSI.
- Avoid testing under conditions that are *both* windy *and* rainy/snowy; the error margin, due to precipitation dependence on vehicle orientation versus wind direction, will greatly increase (the script cannot currently reliably normalize precipitation under such conditions).
- Avoid testing under conditions of optically dense but *scattered* clouds (such as the standard “Cumulus 2” and “Cumulus 3” ones; the error margin, due to brightness dependence on vehicle location, will greatly increase. Instead use the provided “Haze” and “Dust” types of clouds, that induce a more uniform brightness reduction.

D. Changelog

Legend	
Change type ID	**Description**
F	Feature: Introduced completely new functionality that is expected to be considered value-adding by end-users.
I1	User-tangible improvement: Made existing functionality more realistic or complete.
B1	User-tangible bug fix: Corrected existing functionality to conform to its specification, and end-users can perceive the fact.
I2	User-intangible improvement: Modified non-functional aspects of existing functionality, e.g., improved code performance or readability.
B2	User-intangible bug fix: Corrected something that was previously erroneous, but didn't affect end-users.

D.1. Changes in 1.0.0 (06/2018)

ID	Affected vehicles	Affected components	Change type	Description
1-1	CVAG 72	AI, lights, cockpit	F	Introduced preference-based AI usage of interior and exterior lighting. Authored "specification-style" document, formalizing the implementation's functional requirements.
1-2	CVAG 72	Lights, model config	I1	Implemented a simplified `[interiorlight]` analog for pseudo-illumination of key objects of out-of-focus vehicles.
1-3	CVAG 72	Lights, model config	I1	Under interior lighting level 1, when the engine is on, the third row of passenger tubes now activates as well.
1-4	CVAG 72	Cockpit, lights, model config	I1	Fog light switch decoupled from standstill / headlight switch. Added new fog-light-specific ("push" / "pull") triggers.
1-5	CVAG 72	Lights	I1	Passengers no longer complain about the vehicle's interior being too dark under lighting level 1.
1-6	CVAG 72	Model config	B1	Money is now properly illuminated.
1-7	CVAG 72	Lights	B1	Standstill and position lights remain on when electrics off, provided the battery can cope.
1-8	CVAG 72	Lights	B1	High beam flashing always possible, regardless of engine's and electrics' state, provided the battery can cope.
1-9	CVAG 72	Lights	B1	Fog lights are usable if a) at least standstill on, and b) engine on or fog lights switch set to maximum (head (if present) + tail).
1-10	CVAG 72	Lights	B1	Hack for AI recognition of manual high beam flashing (on + manual off) when headlights on (there appears to be an underlying OMSI bug making this impossible when playing by the SDK's rules).
1-11	CVAG 72	Lights, cockpit	I2	To facilitate proper functioning of the AI lighting subsystem, the user can

ID	Affected vehicles	Affected components	Change type	Description
				no longer affect lighting state when the vehicle is AI-controlled.
1-12	CVAG 72	Lights, cockpit	I2	Relocated lighting-related triggers and cockpit-specific logic to cockpit script, for reasons of organization and coupling reduction (light script should not modify cockpit state, and vice versa, with some exceptions still remaining for resolution in a future version). Also got rid of the long obsolete “ober-”/”unterdeck” logic; retained the traditional trigger names for now though, for users’ convenience.

D.2. Changes in 1.0.1 (06-07/2018)

ID	Affected vehicles	Affected components	Change type	Description
2-1	CVAG 72	Lights	I2	Implemented diagnostic utility for conveniently verifying the AI lighting subsystem’s conformance to its specification.

D.3. Changes in 1.1.0 (07-08/2018)

ID	Affected vehicles	Affected components	Change type	Description
3-1	CVAG 72	Lights, model config, passenger cabin config	F	Enhanced interior light sources and states thereof to be more realistic. There are now 6 individually operable passenger cabin fluorescent light tubes, as opposed to the original number of 8 that were controllable in 2 groups of 4. As a consequence, most light-source-to-mesh and -to-passenger-seat mappings had to be revised as well, as had the `[matl_nightmap]` textures of the perceived light emitters themselves. Given the opportunity, the speakers’ (texture) positions were adjusted as well, for more realism.
3-2	CVAG 72	Model config	B1	Temp-fixed a small but ugly wall “gap” behind the driver cubicle, at the point where it meets the wall/ceiling.
3-3	CVAG 72	Lights	B1	Door spotlights now actualize when electrics are off (they previously remained in whatever state they were in previously, until electrics were re-enabled). The same applies to the case where air pressure is insufficient for door operation, that is, when cockpit door buttons blink.
3-4	CVAG 72	Lights	B1	Door spotlights’ timers now expire instantly whenever activation conditions other than doors are no longer satisfied. Previously that was not the case. The bug would manifest in the following edge case: Assume that exterior lighting is active and a door is open, hence its corresponding spotlights are on. Close the door. Just after it has completely closed, quickly deactivate and reactivate the exterior lighting. The expected behavior is that the spotlights go off and stay off. The effective behavior, prior to this fix that is, was that spotlights would go off and on again, due to the timer not having been reset in the meantime.

ID	Affected vehicles	Affected components	Change type	Description
3-5	CVAG 72	Lights	B1	Driver's light is now properly synchronized with front door's spotlights (it would previously update with a short yet noticeable delay of 1 frame).
3-6	CVAG 72	Lights, collision, VDV, model config	I1	Every exterior and interior (excluding dashboard control / indicator lights) light source now individually consumes power (integrated with busbar-logic) and can individually become defunct (integrated with game-native repair function), either due to collision-induced damage (exterior lights only) or "natural" end-of-life. The likelihood of either occurring depends on a) the source type (e.g. LED last longer than fluorescent light tubes, in turn lasting longer than halogen spots), b) the source's significance (e.g. burnt out door spots tend to be replaced less frequently than the driver's spot), c) the source's location (e.g. orange marker lights directly at the front are more susceptible to damage (from, say, accidental contact with sidewalks) than those in the middle), d) the user's maintenance setting (when vehicle spawned by user, otherwise random for AI), and e) the vehicle's age (present year - year of manufacture). Furthermore, when battery critically low and engine off, circuit-level light source (groups) may get throttled (automatically deactivated), depending on their significance, in order to conserve energy. Lastly, fluorescent tubes may enter a partially defunct / "dimmed" state, sometimes long before reaching end-of-life. When in that state, they do not actually *emit* any visible light, but are purely cosmetic.
3-7	CVAG 72	Lights, collision, VDV, model config	I2	Added configurable (.cti-based) simple script-level DRL support. DRL are automatically activated when a) engine running and b) exterior lights (except high beam flashing) off. Also integrated DRL with `Spot_Select`.
3-8	CVAG 72	Lights, collision, VDV, model config	I2	Added configurable (.cti-based) script-level support for an additional tail fog light on the right, and an additional reverse light on the left.
3-9	CVAG 72	Lights, collision, VDV	B1	Light source lifetime once again decreased continuously post-spawn, as happens in the M+R MANs. In the Citaro the pertinent logic was for some reason removed, thus lights would never burn out post-spawn.
3-10	CVAG 72	Lights	B1	Light source lifetime no longer re-initialized on saved situation reload, i.e., on saved vehicle re-spawn.
3-11	CVAG 72	Lights, model config	I1	Introduced beam effects for vast majority of exterior light sources. Also new `[matl_nightmap]` textures for some of them.
3-12	CVAG 72	Lights, VDV	I1	VDV now reports damage of any exterior light source, not merely of broken indicators, headlights, and high beam light bulbs, as it used to.
3-13	CVAG 72	Lights, collision	I1	Collision / breaking glass sound only triggered when an exterior light source at the front becomes defunct due to collision impact; sound volume dependent on which and how many sources have broken.
3-14	CVAG 72	Model config	B1	Driver cubicle's door's material properties changed such that the object looks "less dark". Same for various other parts of the interior which differed significantly, in terms of absorbed light, from their surroundings.
3-15	CVAG 72	Model config	I1	More objects visible in out-of-focus vehicles, including steering wheel, driver's seat, door buttons, and a simplified texture for the VDV dash's display.

ID	Affected vehicles	Affected components	Change type	Description
3-16	CVAG 72	Lights, model config	I1	More objects pseudo-illuminated in out-of-focus vehicles, including cockpit ceiling, driver cubicle's (door), dashboard, steering wheel, ticket validator, interior door surfaces, and seats. Furthermore, objects residing in cockpit area are only illuminated when driver's light on.
3-17	CVAG 72	VDV	I2	Incorporated Morphi's v. 4.4 changes to VDV script.
3-18	CVAG 72	VDV, model config	B1	VDV "percentage bar" icons, e.g. those reflecting fuel availability, no longer get "stuck" on display, when electrics have been abruptly shut down while VDV still undergoing initialization.
3-19	CVAG 72	IBIS, ticket printer, model config	B1	Certain IBIS / ticket printer screens, such as the "departure clearance / signal" overlay, are no longer erroneously rendered when electrics off / unavailable.
3-20	CVAG 72	Lights, VDV	I1	VDV only reports "consumption warning" when light / indicator *sources* are actually active, i.e., actually consume power, not when merely their switch(es) are on; if e.g. interior lighting is active, yet all passenger cabin tubes are defunct, no warning will be emitted. Also, "defunct exterior lights warning" now driven solely by exterior light sources (it previously also accounted for the driver's light, which seemed unrealistic, given the message).
3-21	CVAG 72	Lights, model config	I1	Added a level of indirection between light source state (voltage and operational status), and illumination effects (`[matl_nightmap]`s, `[light_enh_2]` cone/star effects, and `[spotlight]` effects). This allows for more granular control in general, and in particular prevents night-maps from being disabled prematurely, due to a voltage below 0.5 being rounded-down to 0 at the model.cfg level. Also, `Spot_Select` / `[spotlight]` now influenced by light source state, rather than circuit state.
3-22	CVAG 72	Model config	I1	Passenger cabin "STOP" request indicator's `[matl_lightmap]` now always applied, independent of exterior light status (as long as there is sufficient power availability).
3-23	CVAG 72	Lights, model config	I1	Door spotlights now based on `[interiorlight]`s, as opposed to `[matl_lightmap]`s. Doors' interiors, as well as their handles, are now illuminated by the newly-introduced dynamic lights.
3-24	CVAG 72	Door, lights, VDV	I1	Door errors (e.g., insufficient pressure) no longer affect interior lights otherwise affected by door status (driver's light, front door's spotlights, and front-most passenger cabin fluorescent tubes); it only affects the dashboard's door buttons, which then blink.
3-25	CVAG 72	Lights	I1	`AI_Light` and `AI_Interiorlight` signaling improvements when vehicle is user-controlled: • For standstill lights to be perceived as active by other AI vehicles, a) at least one source (standstill, marker, head- or fog light) at the vehicle's head needs to be active, and b) at least one marker light on both sides needs to be active, and c) at at least one source (dimmed brake, marker, or fog light) at the vehicle's tail need to be active. • For indicators to be perceived as active by other AI vehicles, accordingly per side, at least a) the head or side, and b) one tail indicator light need to be active. • For cabin lights to be perceived as active by AI passengers, at least one interior fluorescent tube must be fully functional (partially defunct / "worn-out" state doesn't count).

ID	Affected vehicles	Affected components	Change type	Description
3-26	CVAG 72	VDV, IBIS, ticket printer, model config	I1	VDV (dash lights and display), IBIS / ticket printer (display), and passenger information display are assumed to be connected to an independent battery with infinite capacity, and hence no longer “flicker” when main battery is critically low (typically manifesting during “cold-start”). This oversimplification was introduced to account for the fact that those devices would otherwise need to properly re-initialize / “reboot” under power loss, rather than instantly become operational again, which was in turn deemed too costly to properly implement at the time (may be addressed in a future version).
3-27	CVAG 72	Model config	I2	Removed seemingly duplicate “wagenkasten(_orlf)” entries from model.cfg.
3-28	CVAG 72	Model config	B1	`[interiorlight]`s moved from “shadow” to “wagenkasten” mesh, to prevent lights’ height to vary depending on the shadow’s height, which in turn appears to depend on the underlying terrain the vehicle is on. Previously, that fact would induce curious light “flickering” or complete effect invalidation when e.g. driving down a slope or over an otherwise uneven surface.
3-29	CVAG 72	Lights, cockpit, AI	I2	Added some inline commentary to `lights.osc` and `lights_ai.osc`.

D.4. Changes in 1.2.0 (08/2018)

ID	Affected vehicles	Affected components	Change type	Description
4-1	CVAG 72	Cockpit, model config	I1	Hatches are now only operable when electrics on; they also no longer open instantly.
4-2	CVAG 72	Cockpit, model config	B1	Various dashboard switch control lights no longer remain on when electrics off, including those of the “school driver” indicator mode’s switch, and the side panel’s heating / cooling switches.
4-3	CVAG 72	Cockpit	B1	Auxiliary heating switch now emits a sound when triggered / released.
4-4	CVAG 72	Sound config	B1	Integrated Morphi’s v. 4.4 E3-ZF sounds (previously a Voidth sound-set was in use).
4-5	CVAG 72	Heating-cooling, cockpit, VDV, model config, sound config	I1	Integrated “UCHill” (alternative heating / cooling subsystem), featuring non-instant activation / deactivation of heating / cooling functions, better humidity management, a greenhouse-like effect, and a reduced / “maintenance” fan mode for the driver and passenger A/C functions.
4-6	CVAG 72	Exterior sound volume, sound config	I1	Fine-tuned min/max volume of exterior sounds inside, as well as how much individual open-able surfaces (doors, windows, hatches) contribute.
4-7	CVAG 72	Door, cockpit	I2	Moved door wing locking and “school driver” indicator mode triggers to cockpit script.
4-8	CVAG 72	All scripts having triggers	I2	To facilitate subsequent AI enhancements, the user is no longer allowed to trigger any cockpit, VDV, IBIS / ticket printer, and cash-desk mouse-event when the vehicle is AI-controlled, including emergency door opening

ID	Affected vehicles	Affected components	Change type	Description
				valves, the ramp, and the driver cubicle's door. Sole exceptions: Passenger windows and stop request buttons.
4-9	CVAG 72	Cockpit	I2	Removed a significant amount of leftover cockpit script triggers and macros needed for long-obsolete MAN functions and/or functions exclusive to other Citaro versions (e.g. sliding passenger windows encountered in eastern EU and Russian variants; stand-alone fog light switches; duplicate heating and cooling controls). Likewise removed unnecessary "clutter" from door script (support for 3-/4-D variants and automatic doors). Some functions may be re-introduced on an as-needed basis in later versions, to accommodate for other variants' needs.
4-10	CVAG 72	Lights, cockpit, model config	B1	Indicator improvements: • Activation in "school driver" mode is no longer delayed. • When indicators already active in simple "left-or-right" mode, triggering the "hazard" or "school driver" mode causes the interval timer to reset and hence all indicator lights to go on at that exact instant. • When indicators already active in "hazard" or "school driver" mode, but indicator lever also set, disabling the overlapping mode does not cause the interval timer to reset. • All relevant dashboard indicators are now properly synchronized with the interval timer. • Inverting (left↔right) the indicator lever triggers two sounds (off + on). Activating (off→left/right) or deactivating (left/right→off) the indicator lever triggers a single (on or off, accordingly) sound. There were previously multiple overlapping triggers in the cockpit script which in certain cases erroneously emitted either too many or too few sounds (or none at all). These triggers were cleaned up and merged into just three – "set-left", "set-right", "set-off". • When electrics off, indicators only blink when in "hazard" or "school driver" mode. Indicator light source group output is cut off in simple "left-or-right" mode.
4-11	CVAG 72	Cockpit, sound config	B1	"Hand-brake-released-while-engine-off" warn-sound no longer incorrectly triggered when electrics on but circuit damaged due to collision; likewise for the "schnaufen" and "ambience" sounds.
4-12	CVAG 72	Cockpit, model config	I1	Main sun blind is now fully retractable.
4-13	CVAG 72	Door, cockpit, VDV	I1	Less flexible but more fault-tolerant door wing locking logic: • Locking mode ("rear wing" / "front wing" / "none") can now only be adjusted when both wings are closed. • When switch flipped *while* doors are *in the process of* opening or closing, individual wings can no longer become stuck open or closed; nor can one wing open and the other close, at a single "keystroke" (button push). • VDV display correctly depicts the *effective* state of doors; e.g., if only front wing open, only front wing will be displayed as such, *regardless* of the wing locking switch's setting.
4-14	CVAG 72	Door, cockpit, lights, VDV	I2	Consolidated door opening status into a single block of logic, consistently reused for: • Evaluation of opening / closing preconditions. • Dashboard door button illumination. • Door spotlight activation.

ID	Affected vehicles	Affected components	Change type	Description
				• VDV door status icon rendering. Also synced door button illumination with VDV door status icons.
4-15	CVAG 72	Door	I1	Driver cubicle door’s position affects front door’s opening status perceived by AI passengers, i.e., when the cubicle’s door is even slightly open, passengers won’t attempt to board via the front door and “walk through” the cubicle’s door intersecting with their path.
4-16	CVAG 72	Door	B1	Stop brake no longer automatically *re*-engaged when triggering door *closure*.
4-17	CVAG 72	Door	B1	One can no longer “trick” the stop brake to be engaged and instantly disengaged anew, when triggering door opening *prior to* having stopped, and immediately afterwards proceeding to accelerate.
4-18	CVAG 72	Door, brake, engine, AI	I2	Precautionary measures for AI-controlled vehicles: • Opening and closing of doors does not induce loss of air pressure. • Fuel consumption is zero (in the sense that the vehicle retains exactly the amount of fuel it was spawned with, for as long as it remains under the AI’s control). • Engine ignition always possible and in a timely fashion, even if battery old and/or depleted.
4-19	CVAG 72	Cockpit, door, AI, model config, sound config	F	Enhanced AI usage of doors, wing locking, and the stop brake. Brake pedal released by AI when stop brake or handbrake engaged. Throttle first mildly “tapped”, then, after a random delay, firmly pushed by AI, when stop brake engaged and is to be released.
4-20	CVAG 72	Cockpit, model config	I1	Animated wiper lever wash mode.

D.5. Changes in 1.3.0 (09/2018)

ID	Affected vehicles	Affected components	Change type	Description
5-1	CVAG 72	Lights, model config	I1	Windows are now illuminated by primary passenger cabin lights.
5-2	CVAG 72	AI, cockpit	F	Per-vehicle preference-based AI usage of sun blinds.
5-3	CVAG 72	AI, cockpit	F	Per-vehicle preference-based AI usage of driver’s window.
5-4	CVAG 72	AI, cockpit	F	Per-vehicle preference-based AI usage of heating cooling functions, including: • Driver A/C (temperature, fan speed, air circulation). • Passenger A/C • Cabin heaters
5-5	CVAG 72	AI, cockpit	F	Per-vehicle preference-based AI usage of hatches.
5-6	CVAG 72	AI, cockpit, engine, transmission, brake	F	Asynchronous, randomly-delayed sequences of engine ignition / switching off, and engagement / disengagement of transmission and handbrake, upon arrival at / departure from prolonged stops.

ID	Affected vehicles	Affected components	Change type	Description
5-7	CVAG 72	AI, cockpit	F	Per-vehicle preference-based AI usage of wipers.

D.6. Changes in 1.3.1 (10/2018)

ID	Affected vehicles	Affected components	Change type	Description
6-1	CVAG 72	AI	I1	AI stop work-flow can now detect when next stop is located directly adjacent to current stop (two bus stop cubes being close enough such that their docking distances overlap) and refrain from *visibly* departing (closing doors, etc.). This is a typical occurrence at terminal / initial stops, such as the terminus at "U-Bhf. Ruhleben" on the M+R Spandau map, where buses with standard AI logic will *seem as if* they were to *depart*, only to *instantly arrive* (re-open doors, etc.) again, without having moved a single centimeter in between stop cubes.
6-2	CVAG 72	AI	I1	AI stop work-flow can now detect contextual anomalies, such as the user having changed the time, or the vehicle itself having been "hibernated" (typically due to the user leaving its tile) by the game and "revived" at some arbitrary later point in time, possibly "teleporting" in the process to a different map location, according to its schedule. Awareness of the fact empowers the vehicle to instantly adapt under such conditions and minimize downtime (e.g. remaining stuck in the middle of a busy junction for several seconds, awaiting for the engine to be turned back on, doors to close, etc.), while otherwise still behaving naturally.
6-3	CVAG 72	AI	I1	AI stop work-flow can now detect whether a departure is actually a *parking maneuver*, and behave appropriately (by, if needed, turning the engine on sooner, and occasionally skipping closure of the front door, if applicable). This occurs upon arrival at stops with a long docking distance, where another vehicle (typically another AI bus) blocks the path, but still departs prior to the blocked vehicle's departure time having arrived. The game then instructs the blocked vehicle to depart and stop again, at the same stop, but at the proper location, bus-stop-cube-wise.
6-4	CVAG 72	AI	I1	Long-running AI timers, such as those governing lighting adjustment, can now detect and account for hibernation-induced temporal anomalies. Previously, e.g. a lighting adjustment scheduled to occur with a 10-minute delay might have taken hours to actually manifest, due to the vehicle spending most of its lifetime in hibernation (think e.g. buses serving on short, pure-AI lines, which the user only encounters for a few seconds at a time, while on few shared tiles).
6-5	CVAG 72	Door, AI	I1	Front door's first wing no longer blocks in AI mode. This is only a temp-fix to account for the fact that the AI is not currently capable of detecting blocked wings and resolving the issue by repeatedly re-opening and re-closing them, or employing the emergency valves.
6-6	CVAG 72	AI, cockpit, IBIS, ticket printer, VDV, transmission	I1	During AI takeover of a previously user-controlled vehicle, unsupported AI functions (e.g., retarder, auxheat) are simply disabled, to avoid influencing AI logic. Likewise, AI is now capable of (somewhat) gracefully handling the following less trivial situations: • Resetting emergency door opening valves, if left open, and

ID	Affected vehicles	Affected components	Change type	Description
				properly close the doors afterwards. • Realistically dragging the rear door wheelchair ramp to the folded/closed position, if left open – but only after having opened the rear door. • Realistically dragging the driver cubicle's door closed, if left open. • Switching transmission mode from "R" to "N" or "D", if "R" engaged by user. • Acknowledging the "departure clearance" IBIS / ticket printer overlay. • Resetting the IBIS / ticket printer mode to default, if changed by user. • Resetting the VDV display mode to default, if changed by user.
6-7	CVAG 72	AI, cockpit	I1	During AI takeover of a previously user-controlled vehicle, which is severely damaged (broken engine, transmission, or electrics), the AI will refuse to magically get it to work, but instead opt to do nothing until the user repairs it (via the native menu function). Specifically the AI will switch off the engine if running, switch transmission to "N", engage the handbrake, and activate the hazard indicator mode (if electrics operable).

D.7. Changes in 1.3.2 (10/2018)

ID	Affected vehicles	Affected components	Change type	Description
7-1	CVAG 72	AI	B1	AI stop work-flow can now detect and disregard a stale stop request issued prior to hibernation. Previously, revived vehicles would open the rear door upon arrival, due to a saved stop request, without actually having passengers on board at present.
7-2	CVAG 72	AI, cockpit	I1	AI stop work-flow now closes doors as soon as possible under (subjectively defined) cold or hot (when A/C active) conditions. Furthermore, for added realism, the AI will try to predict, based on its schedule (albeit not always successfully) whether a stop is only going to last for a few seconds, and if so, it may close doors asynchronously when no longer needed for boarding or disembarking, in preparation for a timely departure.

D.8. Changes in 1.4.0 (10/2018)

ID	Affected vehicles	Affected components	Change type	Description
8-1	CVAG 72	AI, cockpit	F	Enhanced AI indicator usage.
8-2	CVAG 72	Sound config	I2	Introduced separate sound configuration file for out-of-focus vehicles.

D.9. Changes in 1.5.0 (11-12/2018)

ID	Affected vehicles	Affected components	Change type	Description
9-1	CVAG 72	Cockpit, model config, sound config	I1	Folding passenger windows can now be locked and/or completely disabled per .cti.
9-2	CVAG 72	AI, cockpit	F	AI (both driver, when vehicle AI-controlled, and passengers) can adjust folding passenger windows' state.
9-3	CVAG 72	Model config	B1	Several key objects were re-ordered at the model.cfg level to attain better "layering" of transparent surfaces. Previously e.g. the black surface in the rear of the driver cubicle used to render *after* ("on-top-of") exterior window materials, producing not-so-pretty results. Doing so also allows for portions of the exterior to once again get dirty.
9-4	CVAG 72	Engine, transmission, brake	I2	Incorporated Morphi's v. 4.4 changes to engine, brake, and transmission scripts.
9-5	CVAG 72	Cockpit, model config	I1	The indicator hazard mode switch can now blink, if configured to do so per .cti.
9-6	CVAG 72	Main, bus config	I2	Restructured overall script / macro execution order, both at the .bus and main script level, in an attempt to somewhat reduce coupling and achieve better organization and predictability of the aggregate input / output pipeline.
9-7	CVAG 72	Dirt, AI	F	AI buses can now spawn and/or get dirty in the process. Washing the vehicle also takes a bit longer.
9-8	CVAG 72	Cockpit, lights	I1	The following functions now require that the ignition key be set to "ON", as opposed to merely *not* being "OFF": • Wipers • Indicators in regular "left-or-right" mode. • Headlights, head fog lights, primary cabin lights. These previously required that the engine be running. Now they require either that, or that the ignition key be at position 2 or 3.
9-9	CVAG 72	Cockpit, door, VDV	I1	Implemented emergency stop brake deactivation switch logic, mapped to dummy cockpit "NOT-AUS" switch. Added VDV warning message to be displayed in the event of the switch being triggered. Removed ability of kneeling down when stop brake overridden.
9-10	CVAG 72	AI, cockpit, lights	I1	During parking maneuvers, the AI no longer: • Resets / closes driver window, if lowered at arrival and hot outside. • Reactivates cabin lights, if temporarily switched off at arrival. • Resets / activates door wing lock, if disabled at arrival and hot outside.
9-11	CVAG 72	Distribution archive, documentation	I2	Various distribution archive changes: • Added independent .bus and model, passenger-cabin, and sound configs. • Added independent directory structure for scripts and modified / new objects, and textures. • Authored trigger / .cti-var reference. • Compiled this very changelog.

ID	Affected vehicles	Affected components	Change type	Description
				• Authored AI lighting diagnostic mode usage guide; added to distribution archive new cloud textures which might be useful for testing purposes.

D.10. Changes in 1.5.1 (01-02/2019)

ID	Affected vehicles	Affected components	Change type	Description
10-1	CVAG 72	Cockpit, model config	B1	The 4 green indicators adjacent to the VDV display, that are not “wired” to any function in particular, now briefly light up during dashboard initialization (electrics “power-up”).
10-2	CVAG 72	Wiper	B1	Wipers no longer temporarily get “stuck” and/or de-synchronized in interval mode, if previously interrupted due to power loss. Likewise no de-synchronization occurs when wipers deactivated while inactive (e.g. when electrics off).

10-3	CVAG 72	Door, lights, wiper, collision, AI, model config, repaint config	I1	New .cti-vars for:
				• Front door wing locking switch availability (+ AI integration).
				• Whether, under a non-running engine, the following functions are allowed / “unlocked” if electrics are switched on (ignition key ≥ position 1), or only at “pre-ignition” (ignition key ≥ position 2):
				  ○ Simple “left-or-right” indicator mode.
				  ○ Hatch adjustment.
				  ○ Driver window adjustment.
				  ○ Driver and passenger A/C fan operation (in reduced / “maintenance” mode).
				  ○ Wiper operation.
				• Whether, under a non-running engine, cabin lighting level 2 (all tubes on) is allowed / “unlocked” at “pre-ignition” (ignition key ≥ position 2), or not at all.
				
				Additionally refactored the two .cti-vars for additional tail fog light and reverse light pair into one.

10-4	CVAG 72	Distribution archive, documentation	I2	Documentation changes: • Decided to drop “specification-style” approach – it would take too long to finish and would be too tedious to maintain by a single editor in the long run. Instead “compressed” existing specification on AI lighting into a simpler, non-technical format and added summary descriptions on the rest of the AI subsystems. Merged that new document with other supplementary documentation that used to reside in individual documents up to this point. • Added installation section to new readme. • Added known issues to new readme. • Added credits and legal sections to new readme. • Added independent .cti subdirectory.
10-5	CVAG 72	AI, lights, cockpit	B1	Vehicles spawned “in-parallel” (e.g. at situation load time, when a bunch of buses are initialized at the same instant) frequently seemed to exhibit *too many* similarities, in terms of AI behavior and other vehicle traits. The

ID	Affected vehicles	Affected components	Change type	Description
				initialization flow of several scripts has been adjusted in an attempt to circumvent this, although we fear there might be an underlying OMSI issue there (worst case: random number generation being the same across vehicles, when requested at the same instant / frame).
10-6	CVAG 72	AI	I1	“Master” AI fail-safe (see item (6-7)) now also triggered when tank without fuel and/or battery depleted (but otherwise operable).
10-7	CVAG 72	Lights	I1	Vehicle age contributes less to light source malfunction probability; maintenance frequency is what matters most.
10-8	CVAG 72	AI	I1	Passengers’ temperature perception is now driven by a combination of environmental and cabin temperature, as opposed to solely the latter. This is of significance when the vehicle is user-controlled: The cabin temperature may then rise several degrees above the environmental temperature due to insolation (greenhouse effect), to which passengers can then respond by being more willing to open windows, even when it is not really that warm outside. Note that this does not really apply to AI-controlled vehicles. First and foremost, window adjustment by passengers only occurs in focused vehicles (which is not typically the case for AI-controlled vehicles, since the user has their own vehicle in focus), due to OMSI not communicating seat occupation status to scripts in unfocused vehicles. Secondarily, in AI mode OMSI alters the cabin temperature such that it doesn’t depart from the environmental temperature too much, which then translates to a much more constrained insolation-induced temperature increase, and hence less of a motivating factor for passengers to open windows unless the environment too is uncomfortably warm. Additionally it should be noted that the driver’s temperature perception remains entirely driven by the environmental temperature for the time being, primarily out of reasons of complexity. This may be revisited in a future version.
10-9	CVAG 72	AI	I1	Unconditional door opening at prolonged stops now varies per stop based on dynamically-evaluated preference and environmental perception (it used to be static depending solely on the latter).
10-10	CVAG 72	AI	B1	Accounted for an AI stop work-flow edge case where door closure would not successfully occur when vehicle re-spawned just about the time its doors were beginning to open (due to stop arrival the instant before), causing the vehicle to either remain perpetually stuck or erroneously drive on with an open front door. The bug was caused by the fact that a) door closure is only possible when the door is already sufficiently open (to avoid premature repetition / state inversion when door button held pushed down longer than absolutely necessary), and b) the vehicle only having (out of design) a single shot (frame) at gracefully adapting in the event of a re-spawn.
10-11	CVAG 72	All scripts depending on electrics state	I2	Minor changes across scripts for more robust battery depletion detection ((`elec_busbar_main` ≠ 0) → (`elec_busbar_main` > `elec_busbar_minV` > 0)).
10-12	CVAG 72	AI, cockpit	B1	Fixed a bug where the vehicle would, during a quick / parking maneuver departure shortly after prolonged stop arrival, magically manage to ignite the engine in spite of transmission still being in “D”.

ID	Affected vehicles	Affected components	Change type	Description
10-13	CVAG 72	AI	I2	Added some inline documentation to `ai_pre_cockpit.osc`.
10-14	CVAG 72	Cockpit, model config	I1	The 4 buttons on the heating / cooling side panel are now toggle-able: They no longer remain visually pushed-down when mouse or keyboard key(s) released. Also adjusted corresponding (push/release) behavior sound-wise.
10-15	CVAG 72	Heating-cooling	I1	The roof-mounted A/C unit no longer emits a sound on the outside when solely the driver A/C function is active; either the passenger A/C and/or the humidity management function must be active for the unit to be audible.
10-16	CVAG 72	Heating-cooling	I1	The fan speed of the passenger A/C function (both functionally as well as sound-wise) now automatically adjusts proportionally to the difference between the temperature of the cabin and the target (ideally-attainable, minimum for cooling or maximum for heating) temperature.
10-17	CVAG 72	Cockpit	B1	Driver window sliding up or down sound no longer emitted when electrics turned off while window switch still held down. Previously the sound would be emitted perpetually (without the window actually moving) until electrics switched back on.
10-18	CVAG 72	AI	I1	AI indicator inversion (e.g. “left” if indicated “right” on arrival) during stop departure now occurs closer to the actual time of departure than it used to (previously vehicles had the tendency to start indicating way too early).
10-19	CVAG 72	AI	B1	The AI can now gracefully handle the case where it needs to adjust the driver window and/or hatches, yet its electrics do not permit doing so (because the AI reduced their operational mode as a result of turning off the engine upon prolonged stop arrival). The AI will then temporarily increase the mode of electrics for as much time apposite adjustments take to complete, then reduce it once again.
10-20	CVAG 72	Dirt, AI	I1	Minor changes to “dirtiness” algorithm, such that overall fewer AI vehicles get significantly dirty under good weather conditions, yet some still do.
10-21	CVAG 72	AI, cockpit	I1	The AI can now manually and momentarily toggle wipers on and off again, under dry conditions, when wiper wings “stuck” at a non-zero angle (due to having previously been forcefully deactivated, in turn due to electrics mode reduction upon prolonged stop arrival).
10-22	CVAG 72	Wiper	I1	When a) wipers at non-zero angle, then b) electrics turned off or reduced, depending on configuration, then c) wipers switched off, then, upon d) reactivating or resetting the electrics, wipers won’t move. Furthermore, if wipers subsequently reactivated, their angle will increase again, even if during previous usage it was on the decline (i.e., new wiper phase = min(previous phase, 2 * π - previous phase)).
10-23	CVAG 72	AI	B1	Addressed a bug where vehicles would always fail, i.e., “neglect” to adjust (invert or disable) their indicators upon scheduled stop departure, when previously a parking-maneuver-style departure had occurred. This edge case would only manifest when vehicles would not be instructed to turn their engine off (since then they would usually disable indicators as well, “masking” this otherwise faulty behavior).
10-24	CVAG 72	Lights, AI	I1	Reduced overall probability of vehicles initially spawned in AI mode having defunct lights.
10-25	CVAG 72	Passenger	I1	Improved passenger illumination (`[interiorlight]` → `[passpos]` /

ID	Affected vehicles	Affected components	Change type	Description
		cabin config		`[illumination_interior]` mappings).