ACC Server Admin Handbook

Version 1.9.3

I Discussion & FAQ

To discuss this document, find a FAQ and ask more questions please visit our Forum thread: link.

II About ACC Multiplayer

The ACC Multiplayer can be split into three sections, where Server Admins can configure and run custom servers for the first two parts.

II.1 Public Multiplayer

Public MP Servers are accessible via the Quickjoin button and the Server List. They run a limited set of options and additional restrictions, so the users know what to expect when they blindly enter a server.

II.2 Private Multiplayer

Once a server is password protected, or doesn’t register to the Lobby at all, we can allow more

sophisticated settings and remove those restrictions - private groups and leagues are capable of letting their users know what specific settings a server may run. We also consider them expert admins, which means this document will focus on the Public MP options first and add a section for private MP once we have features implemented. Up to ACC Version 1.0.5, there are very few extra settings implemented.

II.3 Competition (CP) Servers

Until the final concept for the CP servers is chosen, they will be only run by Kunos. So this document will ignore the CP servers.

III Basic configuration

III.1 File location

You will find the current server files in your ACC steam installation folder:

Steam\steamapps\common\Assetto Corsa Competizione\server

For a first setup, copy the folder to your dedicated server. For updates, just copy and overwrite the accServer.exe. Be sure to have a look at the "log" folder in case the server does not start – in case it detects implausible configurations, it will give a reasonable precise error message and close the process immediately.

III.2 Server configuration files

The server is exclusively configured via JSON files in UTF16-LE format. Using UTF8 file encodings may seem to work but will lead to wrong readings. In general, it is a good idea to get used to the JSON syntax, and/or test your configurations with an online syntax check.

To get a clean start, you can just remove the .json files and start the server once, it will auto-generate them with current defaults. Additionally, you can reduce the "configVersion" contained in each file to get new properties generated (this is true for all the configs). Certain "advanced" settings will be hidden when the value is set to it's default during this process.

The configuration is split into different files, which represent different levels of what you would possibly like to keep or change frequently:

III.2.1 configuration.json

Here we define the very technical settings that possibly never change and define the server "identity".

```
{
  "udpPort": 9201,
  "tcpPort": 9201,
  "maxConnections": 85,
  "lanDiscovery": 1,
  "registerToLobby": 1,
  "configVersion": 1
}
```

The most important thing to know is that both ports must be unique on the system, the firewall allows connections and the ports are accessible from the internet.

Attention: Running a server on a private PC is not recommended. It requires opening and forwarding Ports onto your private system, which makes it vulnerable to random and/or malicious internet traffic. Additionally, private ISP bandwidth is often asymmetrically limited in the upload, which can easily lead to a bad server performance and in the result in a bad multiplayer experience for everyone around.

Property	Remarks
tcpPort	ACC clients will use this port to establish a connection to the server
udpPort	Connected clients will use this Port to stream the car positions and is used for the ping test. In case you never see your server getting a ping value, this indicates that the udpPort is not accessible
registerToLobby	When 0, this server won't register to the backend. Is useful for LAN sessions. If 0, the server is declared "Private Multiplayer". See serverList.json to learn how to bypass the backend's server list and discover servers not listed.
maxConnections	Replaces "maxClients". The maximum amount of connections a server will accept at a time. If you own the hardware server, you can just set any high number you want. If you rented a 16 or 24 slot server, your Hosting Provider probably has set this here and doesn't give you write-access to this configuration file.
lanDiscovery	Defines if the server will listen to LAN discovery requests. Can be turned off for dedicated servers.
publicIP	Explicitly defines the public IP address this server is listening to. Useful if the backend is connected via a different gateway (for example the AWS Accelerator IP). If the publicIP is used, the server has to respond to an additional handshake, or it will immediately shutdown on backend connect.

III.2.2 settings.json

The setting defines your personal server settings, which may be changed from time to time, but also define the server.


```
{
  "serverName": "Kunos Test Server #03",
  "adminPassword": " adminPw123",
  "carGroup": "GT4",
  "trackMedalsRequirement": 3,
  "safetyRatingRequirement": 49,
  "racecraftRatingRequirement": -1,
  "password": "accessPw123",
  "spectatorPassword": "spectPw432",
  "maxCarSlots": 30,
  "dumpLeaderboards": 0,
```

```
  "isRaceLocked": 1,
  "randomizeTrackWhenEmpty": 0,
  "centralEntryListPath": "",
  "allowAutoDQ": 1,
  "shortFormationLap": 0,
  "dumpEntryList": 0,
  "formationLapType": 3
}
```

Property	Remarks
serverName	The server name displayed in the ACC UI pages
adminPassword	Password to elevate via “Server admin commands”
carGroup	Defines the car group for this server. Possible values are `"FreeForAll", "GT3", "GT4", "GTC", "TCX"` where “FreeForAll” will allow any driver to join with any car (that he defined as Primary Car). GT3, GT4, GTC, TCX will restrict this server to GT3, GT4, Cup-like (Porsche Cup, Lamborghini ST, Ferrari CHL) and TCX (BMW M2) entries
trackMedalsRequirement	Defines the amount of track medals that a user has to have for the given track (values 0, 1, 2, 3)
safetyRatingRequirement	Defines the Safety Rating (SA) that a user must have to join this server (values -1, 0, …. 99)
racecraftRatingRequirement	Defines the Safety Rating (RC) that a user must have to join this server (values -1, 0, …. 99)
password	Password required to enter this server. If a password is set, the server is declared “Private Multiplayer”.
spectatorPassword	Password to enter the server as spectator. Must be different to “password” if both is set.
maxCarSlots	Replaces “maxClientsOverride” and “spectatorSlots”. Defines the amount of car slots the server can occupy; this value is overridden if the pit count of the track is lower, or with 30 for public MP. The gap between maxCarSlots and maxConnections defines how many spectators or other irregular connections (ie entry list entries) can be on the server.
dumpLeaderboards	If set to 1, any session will write down the result leaderboard in a “results” folder (must be manually created). See ”Session results”
isRaceLocked	If set to 0, the server will allow joining during a race session. Is not useful in “Public Multiplayer”, as the user-server matching

	will ignore ongoing race sessions.
randomizeTrackWhenEmpty	If set to 1, the server will change to a random track when the last drivers leaves (which causes a reset to FP1). The “track” property will only define the default state for the first session.
centralEntryListPath	Can override the default entryList path “cfg/entrylist.json”, so multiple ACC servers on the machine can use the same entrylist (and custom car files). Set a full path like “C:/customEntryListSeriesA/”, where the entrylist is stored. Attention: The path seperators have to be slashes (/), backslashes (\) will not work.
allowAutoDQ	If set to 0, the server won’t automatically disqualify drivers, and instead hand out Stop&Go (30s) penalties. This way a server admin / race director has 3 laps time to review the incident, and either use /dq or /clear based on his judgement.
shortFormationLap	Toggles the short and long formation lap. Long formation is only usable on private servers.
dumpEntryList	Will save an entry list at the end of any Qualifying session. This can be a quick way to collect a starting point to build an entry list, and is a way to save the defaultGridPositions which can be used to run a race without Qualifying session and predefined grid. Also see the corresponding admin command.
formationLapType	Toggles the formation lap type that is permanently used on this server: 3 – default formation lap with position control and UI 0 – old limiter lap 1 – free (replaces /manual start), only usable for private servers
ignorePrematureDisconnects	Removes a (very good) fix where users can randomly lose the connection. There is no sane reason to turn this off. 1 – default: less arbitrary connections lost 0 – more timeouts, but strict disconnection of anyone who appears inactive for 5 seconds. Can be useful on unsupported platforms where TCP sockets act differently

III.2.3 event.json

Defines the race weekend the server runs. This configuration file is meant to be swappable, so you can easily switch between different event templates by renaming/overwriting them.

```
{
  "track": "spa",
```

```
  "preRaceWaitingTimeSeconds": 60,
  "sessionOverTimeSeconds": 120,
  "ambientTemp": 26,
  "cloudLevel": 0.3,
  "rain": 0.0,
  "weatherRandomness": 3,
  "configVersion": 1,
  "sessions": [
    {
      "hourOfDay": 10,
      "dayOfWeekend": 1,
      "timeMultiplier": 1,
      "sessionType": "P",
      "sessionDurationMinutes": 20
    },
    {
      "hourOfDay": 17,
      "dayOfWeekend": 2,
      "timeMultiplier": 8,
      "sessionType": "Q",
      "sessionDurationMinutes": 10
    },
    {
      "hourOfDay": 16,
      "dayOfWeekend": 3,
      "timeMultiplier": 3,
      "sessionType": "Q",
      "sessionDurationMinutes": 20
    }
  ]
}
```

Property	Remarks
track	The track we run, see “Track name list”. Setting a wrong value will also print out the available track keys in the log. With the 1.8 update the “open” season was introduced, replacing 2018-2020 variants of the track
preRaceWaitingTimeSeconds	Preparation time before a race. Cannot be less than 30s.
sessionOverTimeSeconds	Time after that a session is forcibly closing after the timer reached 0:00. Something like 107% of the expected laptime is recommended (careful: default 2 minutes does not properly cover tracks like Spa or Silverstone).
ambientTemp	Sets the baseline ambient temperature in °C, see “Race weekend simulation”
trackTemp	Obsolete: Track temperatures are always simulated based on ambient temperature, sun angle, clouds and other aspects.
cloudLevel	Sets the baseline cloud level, see “Race weekend simulation”. Has large impact on the cloud levels and rain chances. Values (0.0, 0.1, …. 1.0)
rain	If weather randomness is off, defines the static rain level. With

	dynamic weather, it defines the expected rain level, dependent on weatherRandomness. Values (0.0, 0.1, …. 1.0)
weatherRandomness	Sets the dynamic weather level, see "Race weekend simulation". 0 = static weather; 1-4 fairly realistic weather; 5-7 more sensational
postQualySeconds	The time after the last driver is finished (or the sessionOverTimeSeconds passed) in Q sessions and the race start. Should not be set to 0, otherwise grid spawning is not secure.
postRaceSeconds	Additional time after the race ended for everyone, before the next race weekend starts.
sessions	A list of session objects, see the next table
metaData	A user defined string that will be transferred to the result outputs.
simracerWeatherConditions	Experimental/not supported: if set to 1, this will limit the maximum rain/wetness to roughly 2/3 of the maximum values, translating to something between medium and heavy rain. It may be useful if you feel forced to run very low cloudLevel and weatherRandomness values just to avoid thunderstorm; however high levels (0.4+ clouds combined with 5+ randomness) will still result in quite serious conditions.
isFixedConditionQualification	Experimental/not supported: if set to 1, the server will take the rain, cloud, temperature, rain levels literally and make sure whatever is set up never changes. Daytime transitions still happen visually, but do not affect the temperatures or road wetness. Also rubber/grip is always the same. This is intended to be used for private league qualification servers only. weatherRandomness has to be set to 0, otherwise isFixedConditionQualification will be completely disabled.

Sessions are expressed as an array of:

Property	Remarks
hourOfDay	Session starting hour of the day (values 0 - 23)
dayOfWeekend	Race day (1 = Friday, 2 = Saturday, 3 = Sunday) – relevant to the

	grip and weather simulation.
timeMultiplier	Rate at which the session time advances in realtime. Values 0, 1, … 24
sessionType	Race session type: P, Q, R for (P)ractice, (Q)ualy, (R)ace
sessionDurationMinutes	Session duration in minutes

Remarks:

1) At least one non-race session must be set up
2) Setting up unreasonable day and hours (also consider time multipliers!) can lead to wrong track and weather behaviour, e.g. avoid jumping from Saturday to Friday

III.2.4 eventRules.json

Defines the pitstop rules. Public MP servers will ignore this json file and use default values.

```
{
  "qualifyStandingType": 1,
  "pitWindowLengthSec": -1,
  "driverStintTimeSec": -1,
  "mandatoryPitstopCount": 0,
  "maxTotalDrivingTime": -1,
  "maxDriversCount": 1,
  "isRefuellingAllowedInRace": true,
  "isRefuellingTimeFixed": false,
  "isMandatoryPitstopRefuellingRequired": false,
  "isMandatoryPitstopTyreChangeRequired": false,
  "isMandatoryPitstopSwapDriverRequired": false,
  "tyreSetCount": 50
}
```

Property	Remarks
qualifyStandingType	1 = fastest lap, 2 = average lap (running Endurance mode for multiple Q sessions) . Use 1, averaging Qualy is not yet officially supported.
pitWindowLengthSec	Defines a pit window at the middle of the race. Obviously covers the Sprint series format. -1 will disable the pit window. Use this combined with a mandatoryPitstopCount = 1.
driverStintTimeSec	Defines the maximum time a driver can stay out without getting a penalty. Can be used to balance fuel efficient cars in endurance races. The stint time resets in the pitlane, no real stop is required. -1 will disable the stint times. driverStintTimeSec and maxTotalDrivingTime are interdependent features, make sure both are set or off.

mandatoryPitstopCount	Defines the basic mandatory pit stops. If the value is greater zero, any car that did not execute the mandatory pitstops will be disqualified at the end of the race. The necessary actions can be further configured using the “isMandatoryPitstopXYRequired” properties. A value of zero disables the feature.
maxTotalDrivingTime	Restricts the maximum driving time for a single driver. Is only useful for driver swap situations and allows to enforce a minimum driving time for each driver (IRL this is used to make sure mixed teams like Pro/Am have a fair distributions of the slower drivers). -1 disables the feature. driverStintTimeSec and maxTotalDrivingTime are interdependent features, make sure both are set or off. Will set the maximum driving time for the team size defined by “maxDriversCount”, always make sure both are set.
maxDriversCount	In driver swap situations, set this to the maximum number of drivers on a car. When an entry has fewer drivers than maxDriversCount, maxTotalDrivingTime is automatically compensated so that those "smaller" entries are also able to complete the race Example: 3H race length, 65 minutes driverStintTimeSec and 65 minutes maxTotalDrivingTime will result in 65 minutes of maxTotalDrivingTime for entries of 3 and 105 (!) minutes for entries of 2.
isRefuellingAllowedInRace	Defines if refuelling is allowed during the race pitstops.
isRefuellingTimeFixed	If set to true, any refuelling will take the same amount of time. If turned off, refuelling will consume time linear to the amount refuelled. Very useful setting to balance fuel efficient cars, especially if combined with other features.
isMandatoryPitstopRefuellingRequired	Defines if a mandatory pitstop requires refuelling.
isMandatoryPitstopTyreChangeRequired	Defines if a mandatory pitstop requires changing

	tyres.
isMandatoryPitstopSwapDriverRequired	Defines if a mandatory pitstop requires a driver swap. Will only be effective for cars in driver swap situations; even in a mixed field this will be skipped for cars with a team size of 1 driver.
tyreSetCount	Experimental/not supported: Can be used to reduce the amount of tyre sets any car entry has for the entire weekend. Please note that it is necessary to force cars to remain in the server, or drastically reduced tyre sets will be ineffective, as rejoining will reset the tyre sets.

The basic pitstop features offer a huge array of combinations and different aspects you can set up your non-public races. Not every combination does make sense though, so it's your responsibility to setup the rules so drivers have a good experience.

It should be entirely possible to create a race in the style of a single Sprint series race as well as a 3-24h endurance race with or without driver swaps. Of course this shouldn't limit you to think about your series and especially race event durations, and possibly add more fairness to balance fuel efficient cars – or allow a certain depth of tactics.

Additional notes regarding stints:

- Stint timer (top left corner in the timing HUD) is reset when the car crosses the pit entry and starts counting down again when crossing the pit exit.

- When serving a penalty, the stint timer will freeze and continue counting down after pit exit without resetting.

- When a player's total remaining driving time is less than his current stint time, the total driving time will override the stint timer (!).

When this happens, the stint timer's background turns red, indicating the final stint of the active driver.

For events with mandatory pitstop rules, the features of driverStintTimeSec and maxTotalDrivingTime require each other to be set, if one of them is not set, an excessive safe value is set by the server automatically. In order to make sure your event runs as intended, ensure that driverStintTimeSec, maxTotalDrivingTime, and maxDriversCount values are all set respecting the race length and any additional overtime.

III.2.5 assistRules.json

Can be used to turn off certain assists for any car connected to this server. Beware: disabling assists will effectively remove the effect, but there is no special handling how the assists look like in the menu. Without instructions, users will be surprised and confused – up to a point where they become a risk for other drivers. Whenever you think about disabling something, please be sure this is really

necessary and a risk in terms of fairness. It is out of question that the (quite strong) driving aids “Stability Control” and “Autosteer” may be candidates for league racing, but just turning off the ideal line will not improve anything for anyone (except that the one driver using it may become less safe and ruins the race of others). Even innocent elements like auto-engine start and pit limiter may just force users to re-map their wheels, and for example lose the ability to use their indicators in lapping traffic – again nobody is winning in this scenario.

For (very) obvious reasons, public MP servers will ignore this json file and allow everything.

```
{
  "stabilityControlLevelMax": 25,
  "disableAutosteer": 1,
  "disableAutoLights": 0,
  "disableAutoWiper": 0,
  "disableAutoEngineStart": 0,
  "disableAutoPitLimiter": 0,
  "disableAutoGear": 0,
  "disableAutoClutch": 0,
  "disableIdealLine": 0
}
```

Property	Remarks
stabilityControlLevelMax	Set’s the maximum % of SC that can be used. In case a client has a higher SC set than allowed by the server, he will only run what is allowed (25% in this example). Obviously setting this property to 0 removes all SC, including mouse and keyboard users. The Stability Control is an artificial driving aid that allows the car to act out of the physics boundaries, and highly recommended to overcome input methods like Keyboards, Gamepads and Mouse steering. However, there is a built-in effect that makes the SC performance inferior, so in theory using (and relying) on SC is already more than enough penalty, and the way to improve performance is to practice driving without. Default: 100
disableAutosteer	Disables the steering aid that is only available for gamepad controllers. Unlike SC, this works inside the physics and does not allow unrealistic driving behaviour – except that this is a very strong aid with superhuman feeling for grip and high reaction speed. There is a built-in penalty that should balance the driving performance in most cases, and give an incentive to learn not to use the driving

	aid. Default: 0
disableAutoLights	Forces the equivalent assist option to “off”
disableAutoWiper	Forces the equivalent assist option to “off”
disableAutoEngineStart	Forces the equivalent assist option to “off”
disableAutoPitLimiter	Forces the equivalent assist option to “off”
disableAutoGear	Forces the equivalent assist option to “off”
disableAutoClutch	Forces the equivalent assist option to “off”
disableIdealLine	Forces the equivalent assist option to “off”

III.3 Client configuration files

III.3.1 serverList.json

If this json file exists in a user’s documents folder, that ACC client will perform LAN server lookups on that remote IP address instead of the local network. This can be used to allow dedicated users to find and enter a server that is not listed in the server list, or simply bypassing the server list as an emergency response to backend issues.

If executed well, the first use case can be used as a very strong protection versus malicious denial of service attacks. In this case

a) Make sure your server IP isn’t already known to potential attackers

b) Set `"registerToLobby": 0` in the configuration.json

c) Make sure Port UDP 8999 is accessible and forwarded by the firewall (all ACC servers listen on this UDP port for LAN discovery pakets)

d) Prepare the serverList.json with your server’s public IP

e) Distribute the serverList.json and make sure all your users (Drivers, Teammates, Spectator roles like Broadcaster, Commentator, Stewards) save it in Users/Documents/Assetto Corsa Competizione/Config

The json file consists of a simple `leagueServerIP` property:

```
{
	"leagueServerIP": "158.2.32.117"
}
```

Property	Remarks
leagueServerIP	Declares the server IP which will return ALL existing ACC servers on during a “LAN Server” search.

IV Race weekend simulation

Each server cycle will simulate a race weekend. The configuration starts at Friday night, and simulates weather, track conditions including support program race traffic until the first session configured. With active dynamic weather this means each weekend is unique and unpredictable, while the configuration determines how plausible, subtle or crazy the weekend will feel like.

IV.1 Weather simulation

The comprehensive weather system will start Friday just after midnight and evolve around the baseline values of clouds, rain and temperatures. It is linked to the time of the day, which means running a higher time multiplier in a session will also accelerate the rate of weather changes.

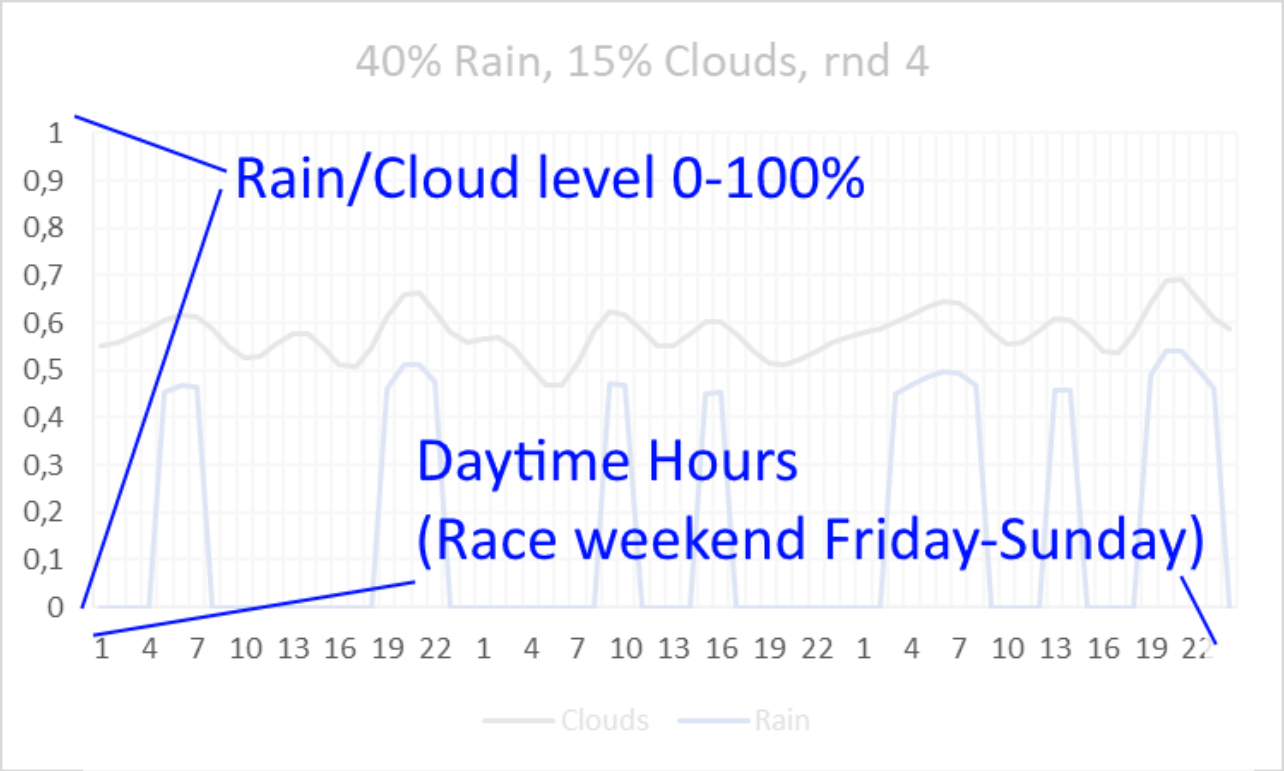
Weather will drive temperatures, sun impact, wind and cloud levels which can lead to rain. Unlike previous versions of the weather model, it is recommended to fully use the range of randomness, cloud and rain parameters to see really dynamic weather with sensational changes – the gravity towards Thunderstorm conditions is removed. Before, there was a set of combinations of cloudlevel and randomness that would create good weather scenarios, but even tiny changes would result in almost dry conditions – or an almost binary cycle between thunderstorm and blue sky.

“rain” now defines the baseline gravity, so set this to whatever intensity you want to see – if rain falls. Please note that even in the most static scenario, rain can just stop for a while. There is still an interconnection with cloud levels, and you can expect some accumulation – high cloud levels will increase the maximum rain intensity above the configured rain level. Still, this is a very useful parameter to declare what kind of rain levels the drivers may face.

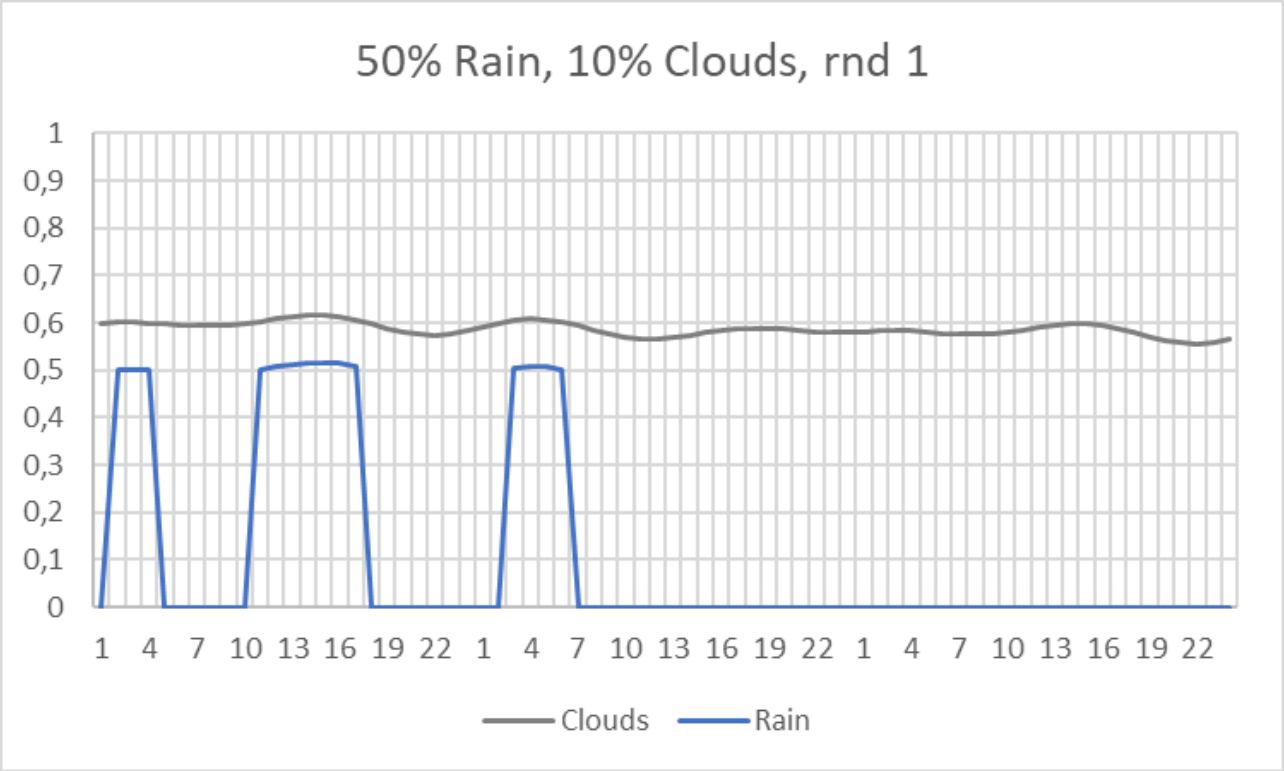
“cloudLevel” sets both the amount of clouds throughout the weekend, but also affects the chance to rain. To recreate typical April-Eiffel weather (where we see rapid blue-sky-into-medium-rain-into-blue-sky transitions), keep the “cloudLevel” low. If you seek a British overcast weekend with 50 shades of rain, keep the cloudLevel at medium levels.

“weatherRandomness” is now much more responsible for how dynamic the weather feels. Higher values allow quicker changes, and more deviation from the declared values. Low randomness allows more control and predictability, especially looking at the maximum rain intensity.

IV.2 Example weekend weather flows

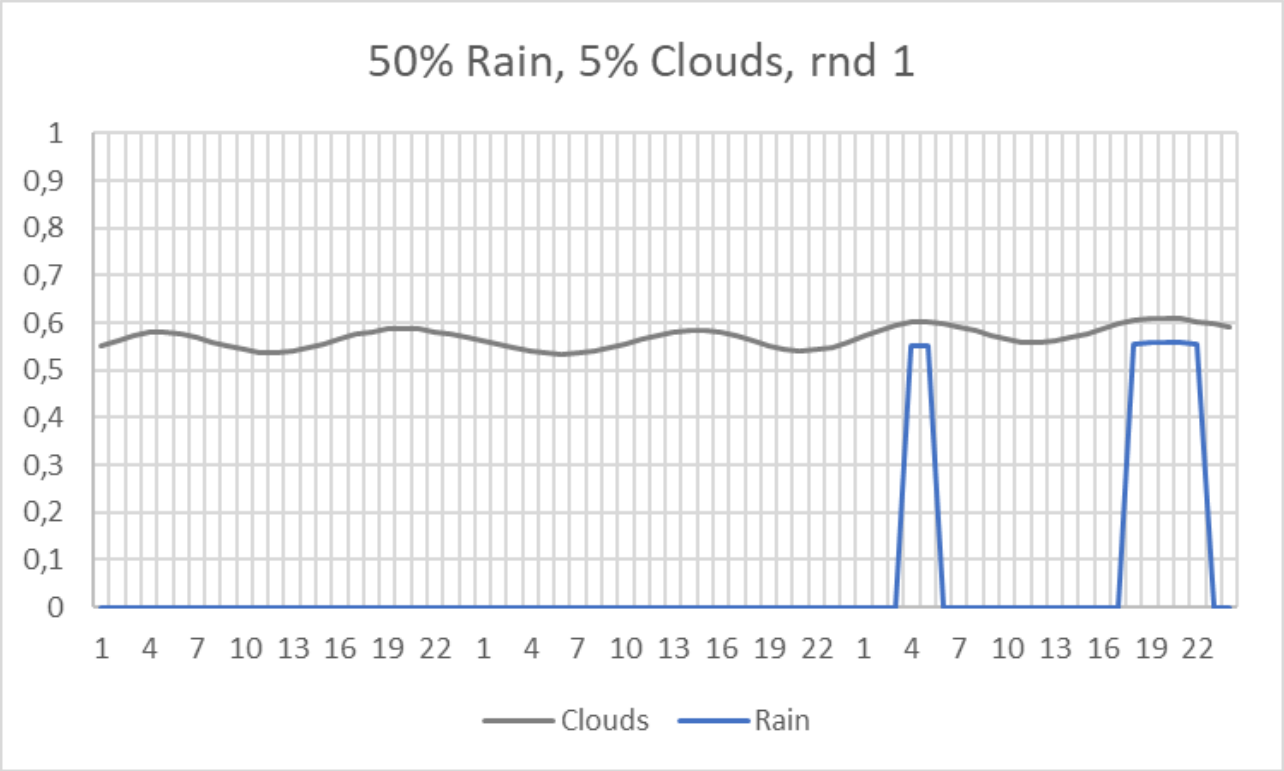


We will go through a series of settings and describe one(!) possible weather flow throughout this weekend. Please note that most of the scenarios are tipping-point situations, and they can also turn out very differently (ie. many low-rain/cloud variants can also end up entirely dry, if the clouds decide to drift downwards). Those scenarios aren’t guaranteed outcomes, but should explain the dependency between the three parameters that define how the weather behaves.

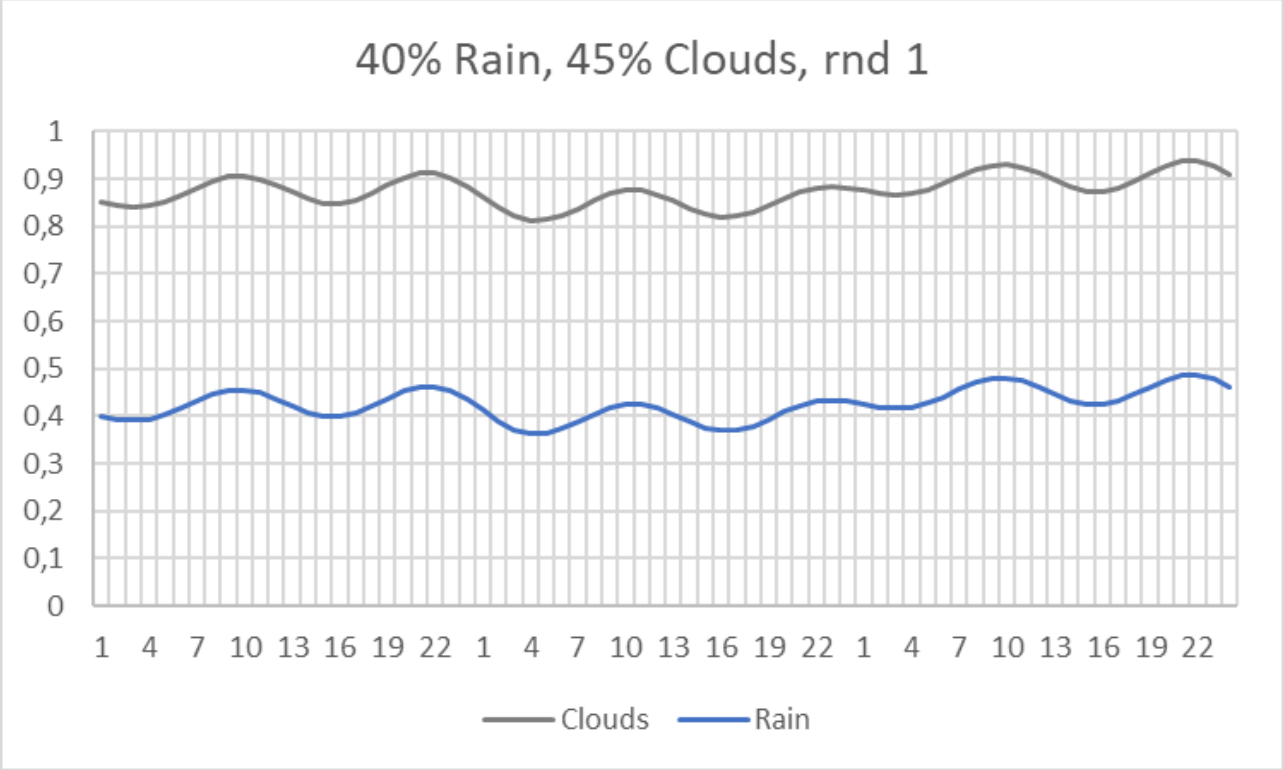


In this scenario, the weatherRandomness 1 (which translates to ~15% variability in SP) applies minimal dynamics on the given 50% rain parameter as baseline. The cloud will vary closely around 60% (in a way this is 50% rain +10% cloud), and therefore playing around the required threshold of 60% cloud level where rain can start to fall.

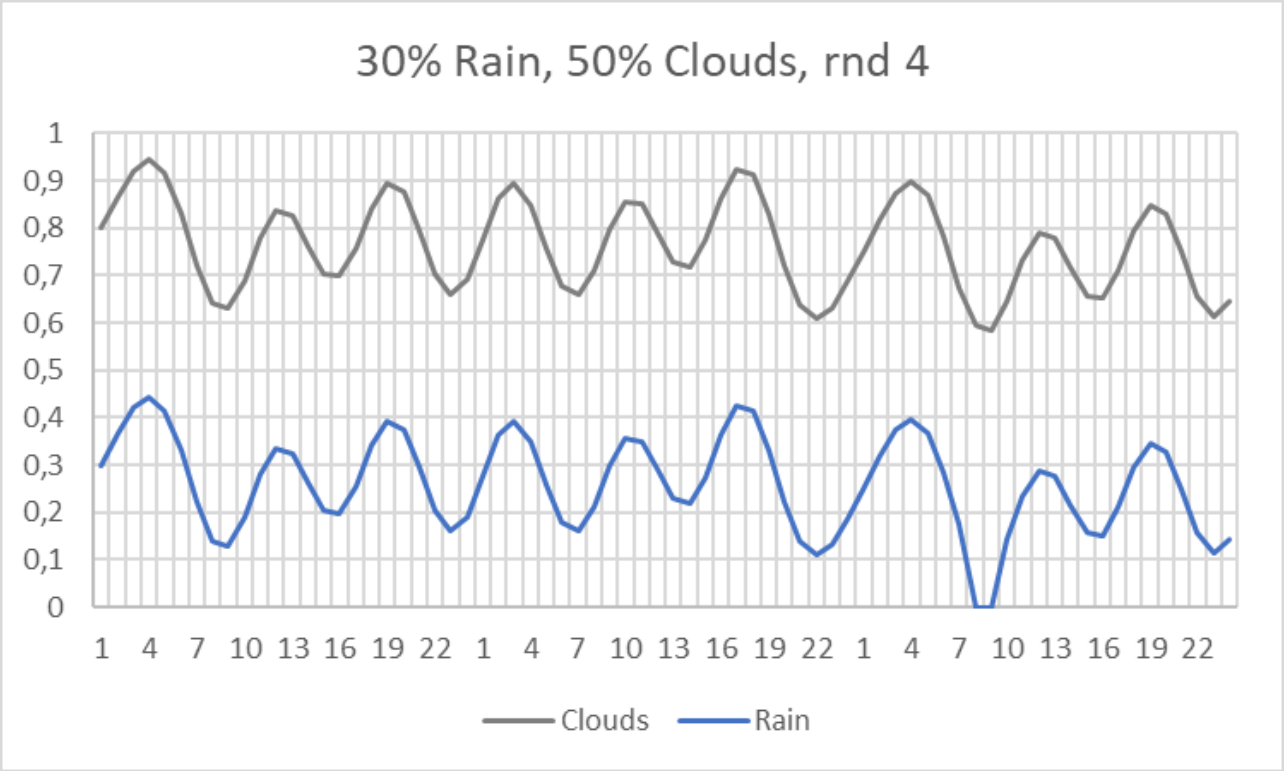
However, if the rain starts, the combination of “rain 0.5” and “weatherRandomness 1” will create predictable medium rain with very few variations. This is the power of low variation, you get what you define.



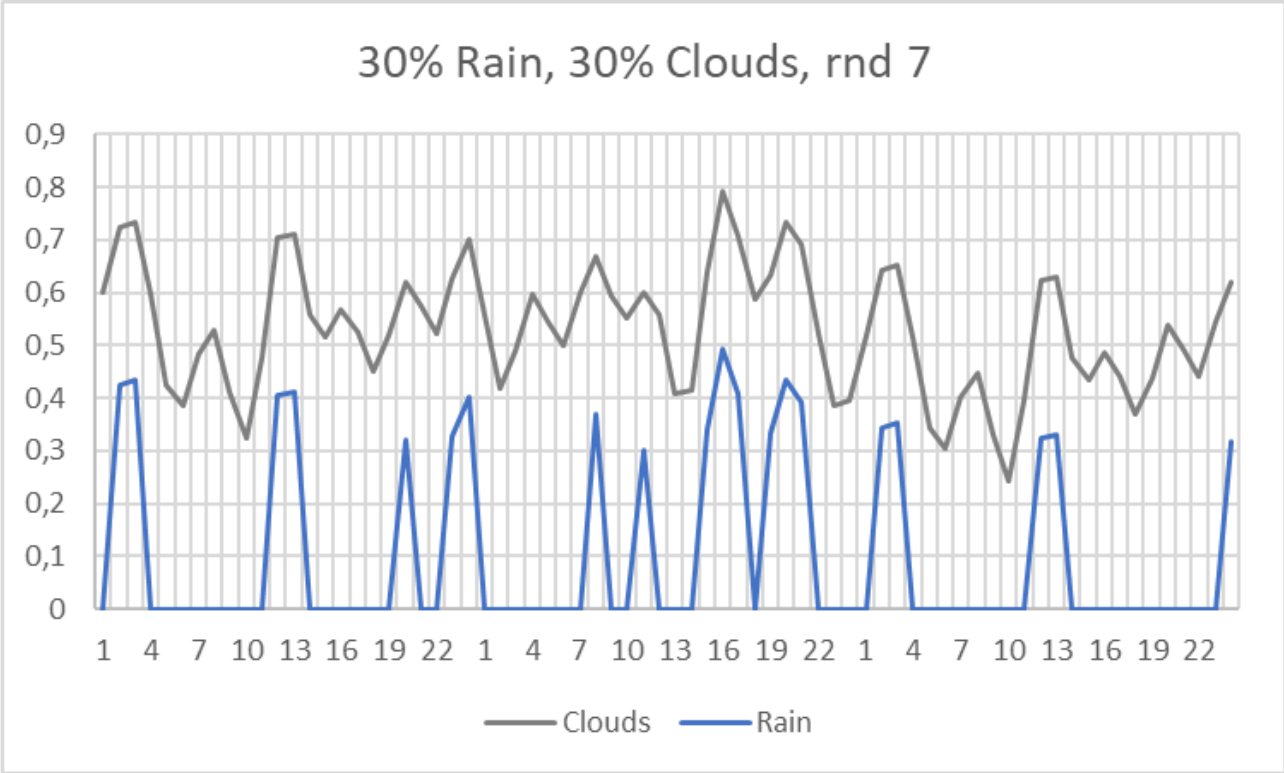
The 60% cloud level threshold is largely important for settings like this; if you put even 5% less clouds, the chance for medium rain will rapidly drop (but still give the desired ~50% rain).



If we instead seek for predictable rainy weekends, we need to raise the cloud level so it reliably stays in the higher values, allowing constant rain with the desired intensity.

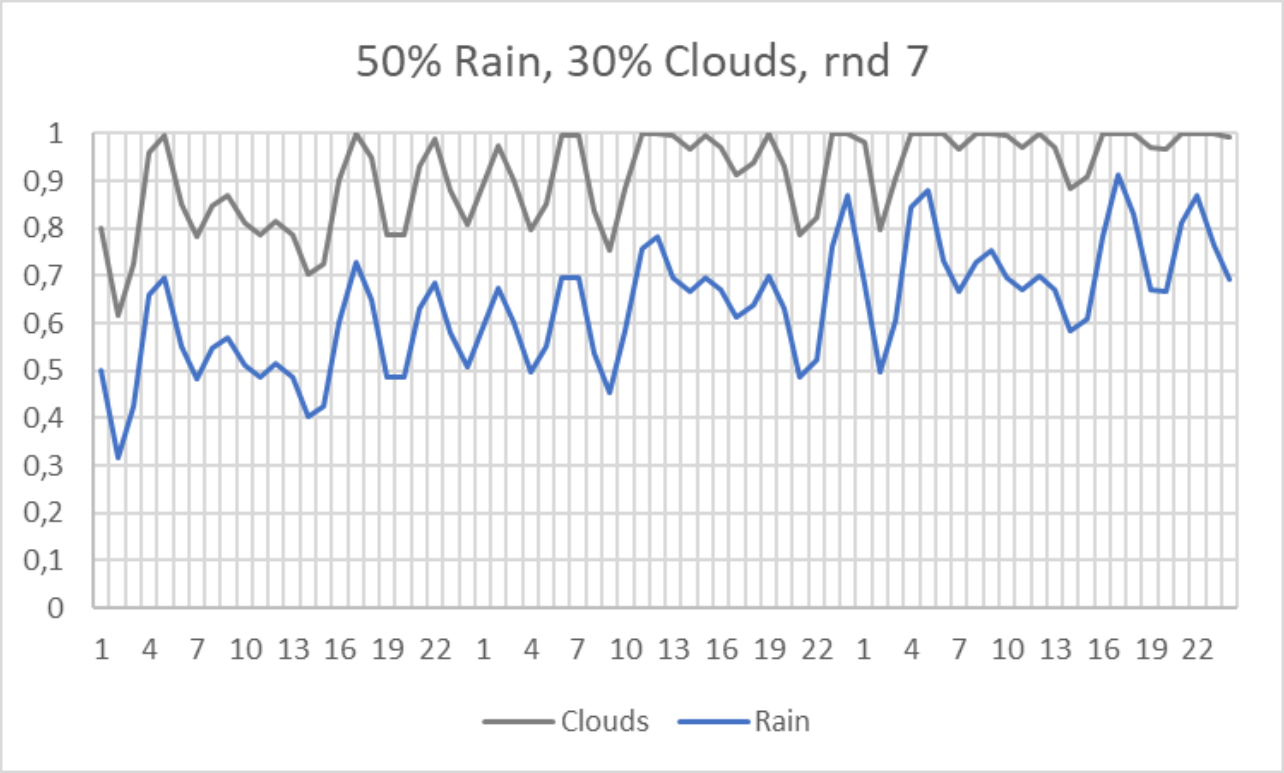


Increasing “weatherRandomness” allows more variation, and speeds up the frequency of changes. Given a similar scenario with slightly reduced rain (to compensate for the higher rain intensity variation) will allow a rainy weekend that has much more change in rain intensity, even stopping rain for two hours.

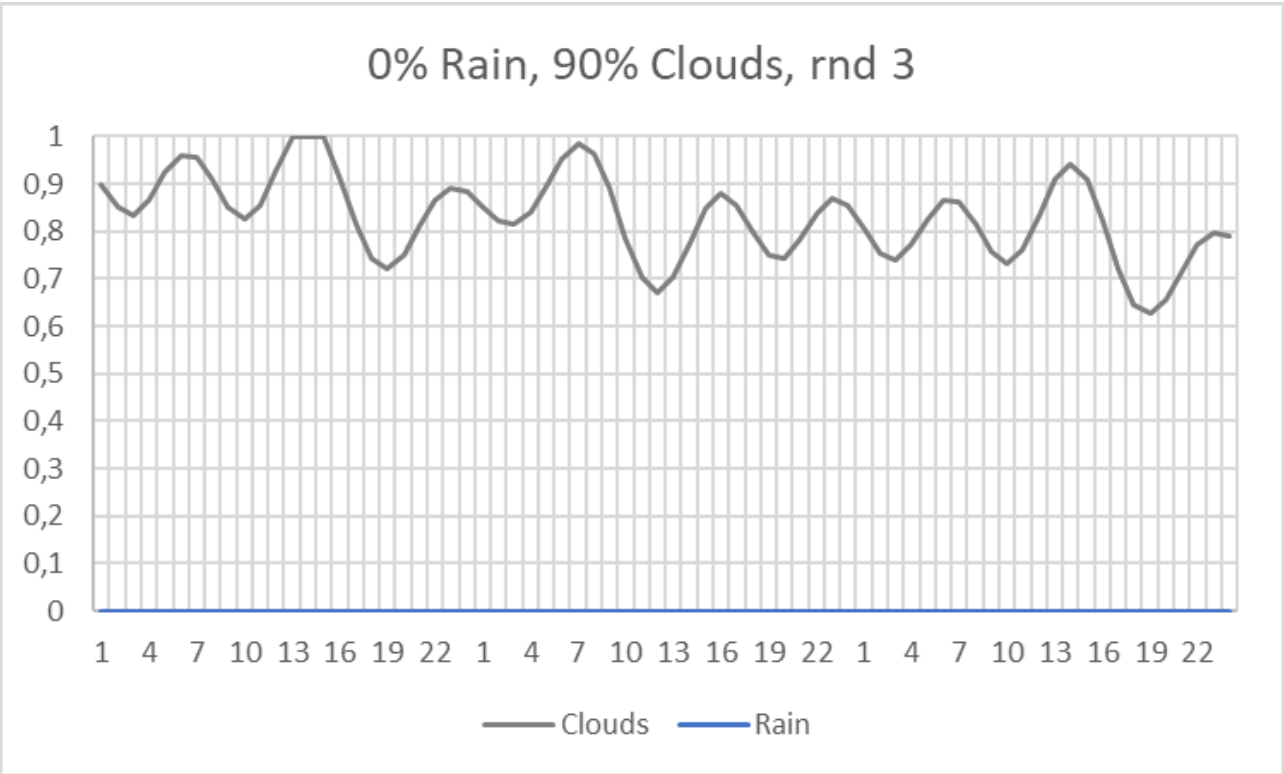


Increasing randomness will again speed up the changes, and increase the difference of cloud and rain intensity. Be careful – heavy rain and thunderstorm conditions can return for any combination of high values, though this model version improves the control we have drastically. But if you ask

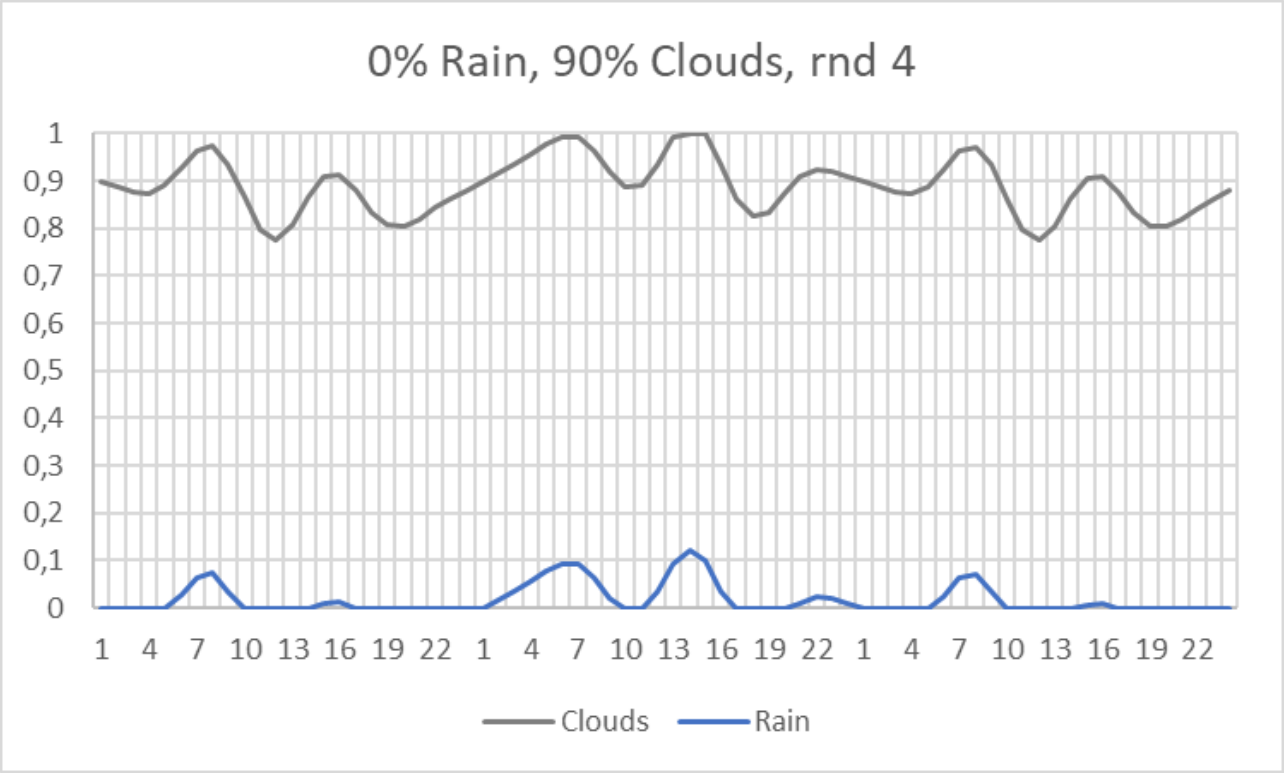
for 70 or 90% rain, you will get it. Using highest weatherRandomness also will give what it promises. Stick with rain values below 50% and medium randomness values to keep simracers happy.



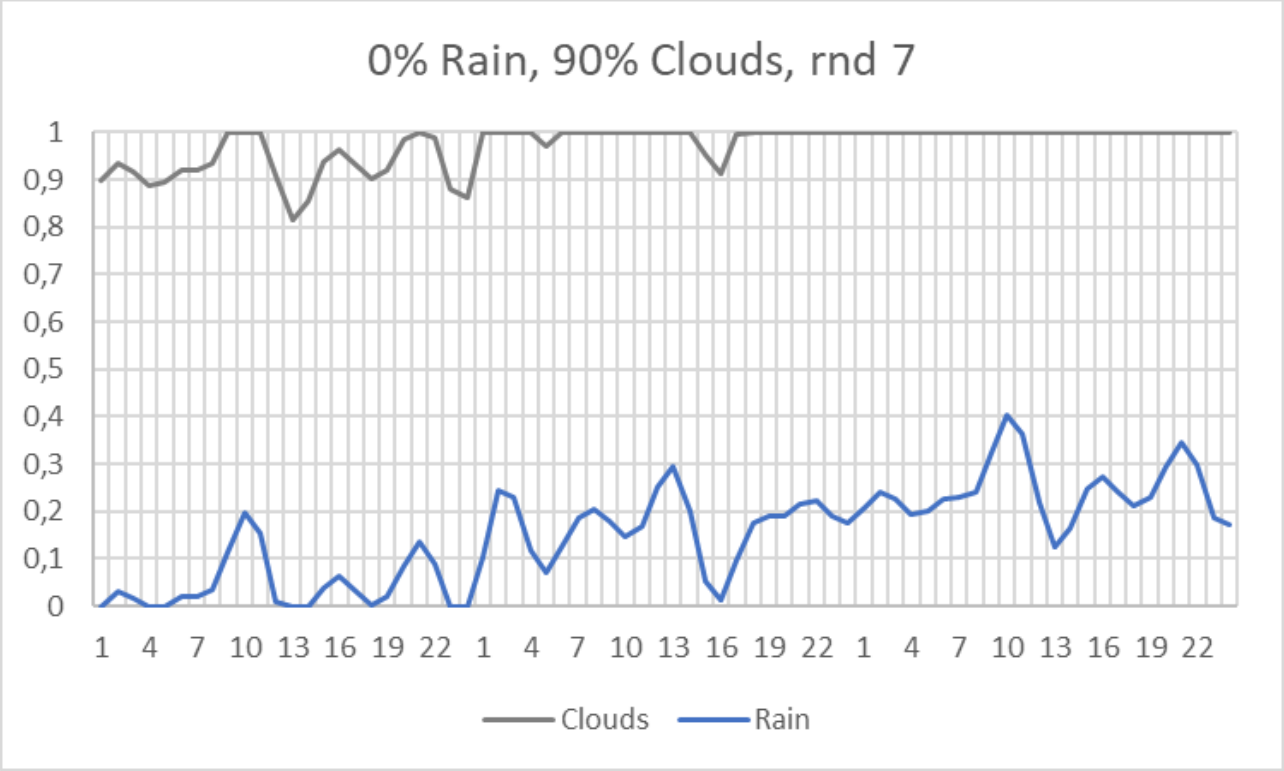
Here we turned 50% baseline rain into thunderstorm territory through the weekend; while it only has a few tolerable peaks into heavy rain during Friday, the Sunday (usually the race day) goes above 90% rain. This will happen with significant rain+cloud values combined with maximum randomness.



To demonstrate the power of the new “rain” parameter: This is nasty overcast weekend that is guaranteed to be dry. This happens for any combination where rain is set to 0 and weatherRandomness is below 4.

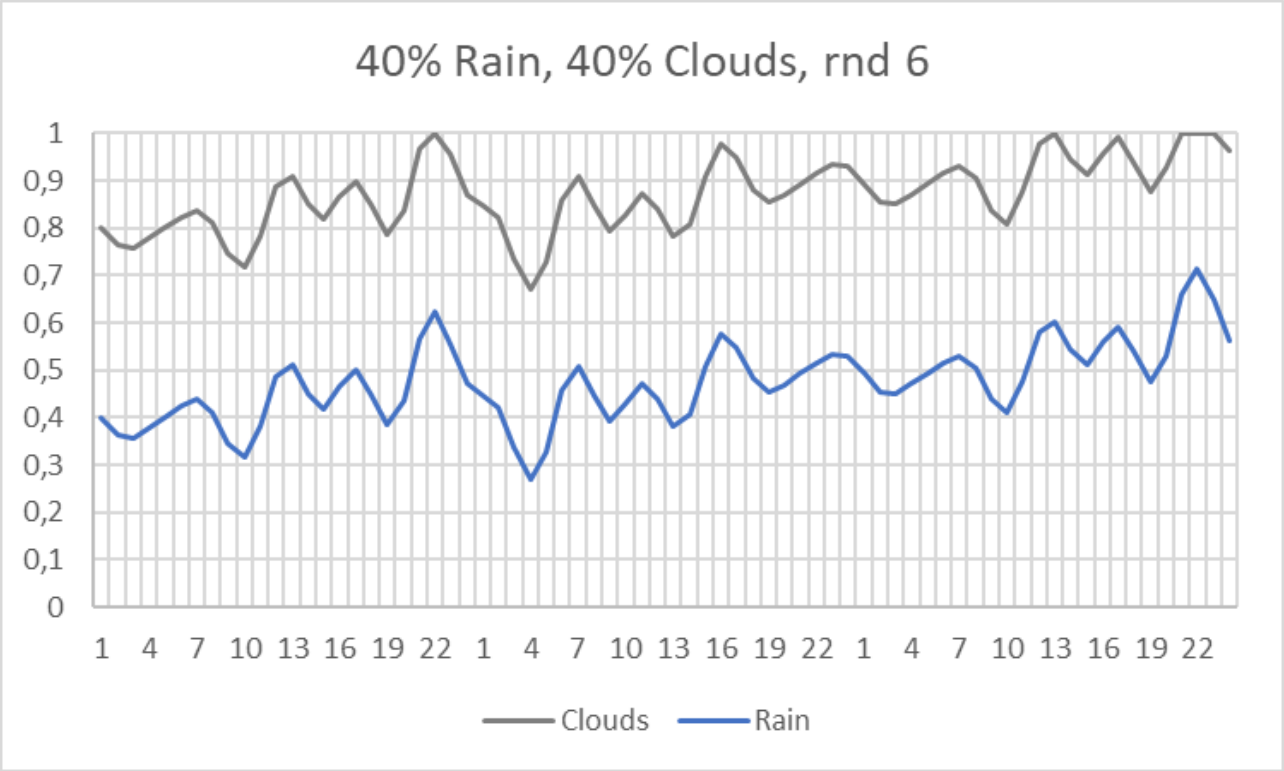


Zero “rain” combined with “weatherRandomness” 4+ allows rain within the natural variation. 4 will keep the clouds move quite a lot, but even with 90% cloud level the rain intensity will not exceed very light rain.

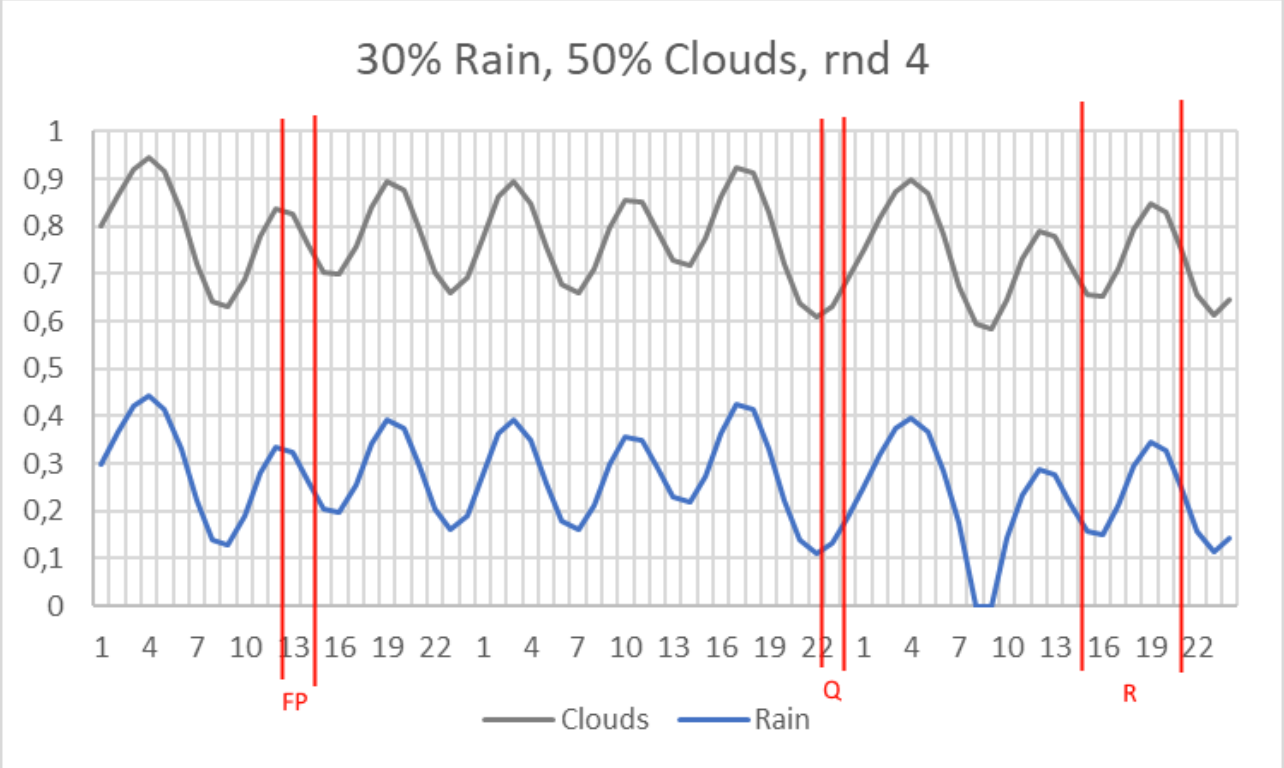


This demonstrates the natural deviation with maximum randomness: We will see many changes,

and the rain intensity can creep up even into the medium rain range.



The shape remains similar if we add baseline rain (and reduce cloud level and weatherRandomness to compensate).



Please note that those charts visualize the weather flow during the whole (simulated) weekend. Your individual session will cover fairly small slices, based on the session length and time multiplier.

IV.3 Scenario examples

1: signature "anything can happen" with potentially high variability:

"cloudLevel": 0.25-0.35,

"rain": 0.2,

"weatherRandomness": 5-7,

2: alternative with potentially more gradual variation

"cloudLevel": 0.45-0.6,

"rain": 0.0,

"weatherRandomness": 5-7,

3: guaranteed overcast (no rain, only changing clouds)

"cloudLevel": 0.6-1.0,

"rain": 0.0,

"weatherRandomness": 1-3,

4: guaranteed sunny (only minor cloud changes)

"cloudLevel": 0.0-0.4,

"rain": 0.0,

"weatherRandomness": 1-3,

5: predominantly overcast with potential rain

"cloudLevel": 0.6-0.9,

"rain": 0.0,

"weatherRandomness": 4-7,

6: guaranteed light-medium rain all weekend

"cloudLevel": 0.6-0.8,

"rain": 0.1-0.3,

"weatherRandomness": 1-3,

7: guaranteed medium-heavy rain

"cloudLevel": 0.6-1.0,

"rain": 0.45-0.8,

"weatherRandomness": 1-3,

IV.4 Track simulation

During a Blancpain race weekend, many different things are going to influence the track conditions. Friday night, we will start with fairly low or no rubber line, and a dusty track in general. Once the support programs start to run, the track will get a bit cleaner and build up rubber. How your first session looks like strongly depends on the race day you set up – Friday morning before 10am will give you the virgin track, while Saturday evening is already quite good grip. If it didn't rain in between of course; high weather variation will of course often wash out the track. But as we also simulate the traffic of the other series during the weekend, you may find yourself in wet track with low/no rain, and the dry line started to form. Or the opposite, your session may find a really rubbered ideal line while it just starts to rain (beware, highly slippery).

Unlike the weather, the track is not affected by the time multiplier, and always "runs" in real time. That means even if you have a quick thunderstorm for one minute, your track won't immediately rush to full wetness levels, and it also will dry out in real time. The water dissipation rate depends on sun angle, cloud level, temperatures and wind. That means a hot, sunny mid-day scenario may clean up the track within minutes, while you can still find the track wet and full of puddles after a cold night (no sun) or cloudy hours.

Again, having realistic and plausible weather settings will help users to understand what is going on based on their every-day experience, which in the end makes their experience better – be careful with extreme weather settings and time multipliers. It is also recommended to think about the real track observations; Locations like Spa or Nürburgring tend to have unlimited amounts of surprises, while Barcelona is known to be quite stable and hot (I'll deliberately leave out the Italian tracks here, after having visited them in 2018 and 2019).

In a recap, we exposed very few parameters to a quite complex system, and playing with those will vastly influence the experience on your server. It is worth to learn and experiment with those settings!

V Server admin commands

While connected to a server (both as driver and spectator), users can elevate to "server admins" if they are aware of the password. That allows them to use a few special commands. Version 1.0 start with a limited set, which is expected to be extended in future versions.

To elevate to admin, hit "enter" to use the chat and type

/admin adminPw123

A notification will tell you if successful. Additionally, you can setup an "Entry lists" entry for the

admin(s) steamids.

Once elevated, you can use several commands:

Command	Parameters	Remarks
/next		Skips the current session
/restart		Restarts the current session. Do not use this during the preparation phase
/kick	car race number	Kicks a user from the server, preventing him to join again until the race weekend restarts
/ban	car race number	Bans a user from the server, preventing him to join again until the server restarts
/dq	car race number	Instantly disqualifies the car, teleporting it to the pits with locked controls
/clear	car race number	Removes pending penalties (e.g. Drivethrough or Stop&Go) and DSQs.
/clear_all		Removes all penalties and DSQs from all cars.
/tp5	car race number	Adds a 5s penalty to the given car. Can accumulate and will be added in the final result and json outputs. Also comes with a /tp5c variant for special message “for causing a collision”.
/tp15	car race number	Adds a 15s penalty to the given car. Can accumulate and will be added in the final result and json outputs. Also comes with a /tp15c variant for special message “for causing a collision”.
/dt	car race number	Assigns a drivethrough penalty to the given car. As other DTs, it has to be served within 3 laps or the car will become disqualified. In case of a race finish within those 3 laps, the DT will be transformed to a 80s time penalty. Also comes with a /dtc variant for special message “for causing a collision”.
/sg10	car race number	Assigns a Stop&Go 10s penalty to the given car.
/sg20	car race number	Assigns a Stop&Go 20s penalty to the given car.
/sg30	car race number	Assigns a Stop&Go 30s penalty to the given car.

/manual start		Removed + replaced by settings.json formationLapType
/manual entrylist		Manually creates an entry list json file containing the drivers connected in this moment (see also Settings -> dumpEntryList)
/debug formation		Prints information about the current cars and their states. Useful information to report bugs during the formation lap
/debug bandwidth		Let's the server console print the current bandwidth relevant data for both TCP and UDP traffic. Type again to turn it off.
/debug qos		Let's the server console print the current network connection quality data. Type again to turn it off.
/ballast	car race number kg	Sets the car's ballast (in kg) to what you defined. This overrides any other ballast this car has been assigned. Range is 0 to 100. Example: /ballast 113 15 will give car #113 15kg of ballast
/restrictor	car race number % restrictor	Sets the car's restrictor (in %) to what you defined. This overrides any other restrictor this car has been assigned. Range is 0 to 20. Example: /restrictor 113 7 will give car #113 7% restrictor, resulting in roughly 7% less hp at max rpm.

Notes:

- The race control commands /tp5, /tp15, /dt have another variant each (/tp5c, /tp15c, /dtc), which will change the message to a more immersive "for causing a collision"

VI Entry lists

Using entry lists a server admin can set up special roles that link drivers (by Steam Id) to those configuration entries.

It allows the server to identify persons, and allows to force or allow various aspects. Entry list entries will always bypass Rating Requirements, and will be able to join servers even if they are full (as long as we have pit slots left, and the driver number is smaller than configuration.json/ maxClients).

To start, just add a new file called entrylist.json in the “cfg” folder. Using an entry list does not interfere with the classification of “Public” or “Private” MP, and you can selectively use an entry to e.g. reserve a slot for you in a Public MP server. Serious groups might use slots for their members and run a race with a 90 SA restriction in Public MP to fill up their rows with highly capable “randoms”.


```
{
  "entries": [
    {
      "drivers": [
        {
          "playerID": "S765611xxxxxxxxxx1"
        }
      ],
      "raceNumber": 88,
      "forcedCarModel": -1,
      "overrideDriverInfo": 0,
      "isServerAdmin": 1
    },
    {
      "drivers": [
        {
          "firstName": "First",
          "lastName": "Driver",
          "shortName": "NO1",
          "driverCategory": 2,
          "playerID": "S765611xxxxxxxxxx3"
        },
        {
          "firstName": "Another",
          "lastName": "Person",
          "shortName": "NO2",
          "driverCategory": 1,
          "playerID": "S765611xxxxxxxxxx4"
        }
      ],
      "raceNumber": 114,
      "forcedCarModel": -1,
      "overrideDriverInfo": 0,
      "defaultGridPosition": -1,
      "ballastKg": 0,
      "restrictor": 0,
      "customCar": "exampleCar.json",
      "overrideCarModelForCustomCar": 1,
      "isServerAdmin": 0
    }
  ],
  "forceEntryList": 0
}
```

Property	Remarks
entries	List of entries, see next table
forceEntryList	Will reject drivers that are not in the entry list. Default is 0, which allows the partial definition of entries in a “normal” server configuration. Cannot be used on public servers.

Entry definitions:

Property	Remarks
drivers	List of drivers, see next table. Must at least contain one driver with the SteamId
raceNumber	The preferred race number if set, -1 if the driver may decide by picking his car. Values 1 - 998
forcedCarModel	If not set to -1: user cannot join with a different car, see “Car model list” for the values
overrideDriverInfo	If set to 1, the driver’s name and category will be overridden by what is setup in the entry list. If set to 0, it’s up to the client joining.
customCar	If set to a filename, the car, team and appearance will be used no matter what the user chose (Exception: overrideCarModelForCustomCar). This is useful for leagues and events, where we want consistent car appearance and the chosen car model for the corresponding driver/team. The custom car file has to be located in a “cars” folder next to the entrylist.json (also works for centralEntryListPath). Leave blank (“”, =default) to let the user chose the car via car selection UI.
overrideCarModelForCustomCar	If customCar is used, this setting will apply the car model configured if the value is set to 1 (which is the default). If set to 0, all values except the carModel are applied, so the user is free to pick a car but while team name and appearance will be applied.
isServerAdmin	If set to 1, that user will be automatically elevated to server admin when he joins.
defaultGridPosition	If set to a value of 1 or greater, this car will obtain this grid position if a race starts without any qualifying session

	before (e.g. P -> R).
ballastKg	Assigns ballast in kg for this car. Will be additive to ballast for the car model (via bop.json), and can be overridden by the admin command /ballast. Range is 0 to 100.
restrictor	Assigns restrictor in % for this car. Will be additive to restrictor for the car model (via bop.json), and can be overridden by the admin command /restrictor. Range is 0 to 20.

For each entry in “drivers”, we will need at least the SteamId defining the entry. Other possible values:

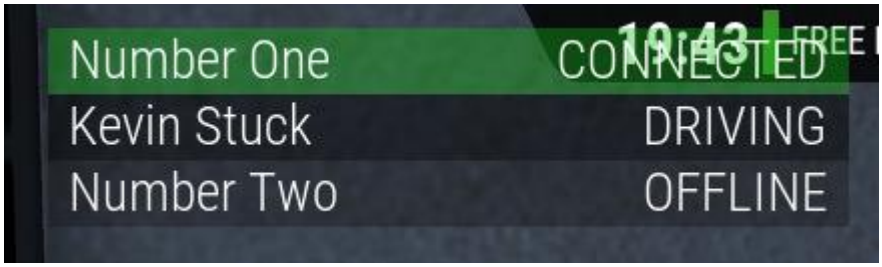
Property	Remarks
firstName	First name of the driver, if “overrideDriverInfo” is set to 1
lastName	Last name of the driver, if “overrideDriverInfo” is set to 1
shortName	Short name of the driver, if “overrideDriverInfo” is set to 1
driverCategory	Bronze/Silver/Gold/Platinum category, if “overrideDriverInfo” is set to 1. See “Driver Category list” for values
playerID	Steam64 Id for this client; Add a “S” in the front

More options are to come and will be added to the document. Please do not ask for the undocumented fields that may be auto generated, they aren’t supported or implemented yet.

VI.1 Teams

You may have noticed that entry lists can have more than one driver on a car entry – which is how we set up Teams for Driver Swaps. Once you declare two drivers, the first driver to join will enter the car, while the other drivers will join in a special spectator role (they must not use the spectator password).

While in a team, drivers will see a list in the driving HUD, displaying the connection state of their mates:



Using the Pit strategy page in the (M)ulti(F)unctional(D)isplay, the driver can assign a driver swap

which will be automatically executed during the next pitstop.

Additionally, drivers can swap during Practice and Qualifying sessions while the car is inside the pitlane. Just use the chat command &swap <driver number>, so in the example above “&swap 1” would hand over the car from “Kevin Stuck” to “Number one” (as Stuck is driving, and One is connected).

VI.2 Custom cars

With the version 1.2, the livery editor now makes custom skin templates an official feature. You can simply create an “exampleCar.json” using the livery editor and find the corresponding json file in Documents\Assetto Corsa Competizione\Customs\Cars.

Given the “exampleCar.json”, the file needs to be located in cfg/cars/example.json (or customEntryListPath/cars/example.json) in order to have a full server defined car entry.

VI.3 Additional BoP

In addition to the admin commands and entry list based ballast and restrictor, you globally define those values for car and track combinations.

To accomplish this, just add a bop.json like this:

```
{
  "entries": [
    {
      "track": "zolder",
      "carModel": 7,
      "ballastKg": 1,
      "restrictor": 2
    },
    {
      "track": "silverstone",
      "carModel": 2,
      "restrictor": 5
    }
  ]
}
```

Each “entry” can be understood as a line with a composite key of track and car model, and the

ballast/restrictor are applied if both match. This way you can apply a small balance to one precise combination, or create your full blown custom BoP for all kinds of seasons, tracks and cars.

Important notes:

1) The additional BoP is at the moment adjustable on public servers, but we will look at this closely. If we see bad and overdone efforts that affect the experience of average users, we will limit the additional BoP to private servers

2) It is very hard to improve the current BoP, much harder than it may look (especially for 2019, where SRO did a great job). Do not fall for the mistake to use hotlap performance as the reference, and also make sure to start with tiny adjustments.

The values of the bop.json act additive to the entry list, so you can combine a championship penalty with your track/car specific changes. Any admin command (/ballast or /restrictor) will set the corresponding value to whatever you define, and overwrite potential configurations until the car rejoins.

VII About Spectators

To join a server as a spectator, just enter the “spectatorPassword” in the password box in the Multiplayer Server List. You won’t occupy a car or pitslot, but still can chat and elevate to admin.

Spectators still use server resources, so they will be limited by the total “maxClients” count – however spectator slots are not limited by the global limit of 30 cars, nor do they occupy pit slots.

VIIISession results

VIII.1 Result files

Using the “dumpLeaderboards”: 1 option, any session that is finished will write the final standing into a .json file in the “results” folder. Those files are generated with a filename in the pattern of “190806_193009_R.json”, including date, time and session type (P, Q, R).


```
{
    "sessionType": "R",
    "trackName": "silverstone",
    "sessionIndex": 1,
    "sessionResult": {
        "bestlap": 117915,
        "bestSplits": [
            34770,
            49359,
            33258
        ],
        "isWetSession": 0,
        "type": 1,
        "leaderBoardLines": [
            {
                "car": {
                    "carId": 1073,
                    "raceNumber": 912,
                    "carModel": 0,
                    "cupCategory": 0,
                    "teamName": "",
                    "drivers": [
                        {
```

```
                        "firstName": "Somebody",
                        "lastName": "Else",
                        "shortName": "SOE",
                        "playerId": "S76561191111111111"
                    }
                ]
            },
            "currentDriver": {
                "firstName": "Somebody",
                "lastName": "Else",
                "shortName": "SOE",
                "playerId": "S76561191111111111"
            },
            "currentDriverIndex": 0,
            "timing": {
                "lastLap": 119223,
                "lastSplits": [
                    35286,
                    50178,
                    33759
                ],
                "bestLap": 118404,
                "bestSplits": [
                    35265,
                    49659,
                    33438
                ],
                "totalTime": 719894,
                "lapCount": 6,
                "lastSplitId": 0
            },
            "missingMandatoryPitstop": 0,
            "driverTotalTimes": [
                0.0
            ]
        }, ...
    ]
 },
 "laps": [
    {
      "carId": 1073,
      "driverIndex": 0,
      "laptime": 125511,
      "isValidForBest": true,
      "splits": [
        40197,
        51537,
        33777
      ]
    }, ...
],
 "penalties": [
    {
        "carId": 1079,
        "driverIndex": 0,
        "reason": "Cutting",
        "penalty": "DriveThrough",
        "penaltyValue": 3,
        "violationInLap": 0,
        "clearedInLap": 1
    },
    {
        "carId": 1081,
        "driverIndex": 0,
        "reason": "PitSpeeding",
```

```
            "penalty": "StopAndGo_20",
            "penaltyValue": 20,
            "violationInLap": 4,
            "clearedInLap": 5
        }
    ]
}
```

In a nutshell, the root object contains information about the session and track, next to a leaderboard representation, a complete list of laps and a complete list of penalties for this session. The properties should be self-explanatory, in case of doubts and questions please do not hesitate to join the official support forum.

IX Appendix

IX.1 Track name list with slots

Note: Public MP is limited to unique pit box OR 30 at max.

Value	Unique pit boxes	Private server slots
monza	29	60
zolder	34	50
brands_hatch	32	50
silverstone	36	60
paul_ricard	33	80
misano	30	50
spa	82	82
nurburgring	30	50
barcelona	29	50
hungaroring	27	50
zandvoort	25	50
kyalami	40	50
mount_panorama	36	50
suzuka	51	105
laguna_seca	30	50
imola	30	50
oulton_park	28	50
donington	37	50
snetterton	26	50
cota	30	70
indianapolis	30	60
watkins_glen	30	60
valencia	29	50

IX.3 Car model list

Value	Car model
0	Porsche 991 GT3 R
1	Mercedes-AMG GT3
2	Ferrari 488 GT3
3	Audi R8 LMS
4	Lamborghini Huracan GT3
5	McLaren 650S GT3
6	Nissan GT-R Nismo GT3 2018
7	BMW M6 GT3
8	Bentley Continental GT3 2018
9	Porsche 991II GT3 Cup
10	Nissan GT-R Nismo GT3 2017
11	Bentley Continental GT3 2016
12	Aston Martin V12 Vantage GT3
13	Lamborghini Gallardo R-EX
14	Jaguar G3
15	Lexus RC F GT3
16	Lamborghini Huracan Evo (2019)
17	Honda NSX GT3
18	Lamborghini Huracan SuperTrofeo
19	Audi R8 LMS Evo (2019)
20	AMR V8 Vantage (2019)
21	Honda NSX Evo (2019)
22	McLaren 720S GT3 (2019)
23	Porsche 911II GT3 R (2019)
24	Ferrari 488 GT3 Evo 2020
25	Mercedes-AMG GT3 2020
26	Ferrari 488 Challenge Evo
27	BMW M2 CS Racing

28	Porsche 911 GT3 Cup (Type 992)
29	Lamborghini Huracán Super Trofeo EVO2
30	BMW M4 GT3
31	Audi R8 LMS GT3 evo II
32	Ferrari 296 GT3
33	Lamborghini Huracan Evo2
34	Porsche 992 GT3 R
35	McLaren 720S GT3 Evo 2023
50	Alpine A110 GT4
51	AMR V8 Vantage GT4
52	Audi R8 LMS GT4
53	BMW M4 GT4
55	Chevrolet Camaro GT4
56	Ginetta G55 GT4
57	KTM X-Bow GT4
58	Maserati MC GT4
59	McLaren 570S GT4
60	Mercedes-AMG GT4
61	Porsche 718 Cayman GT4

IX.4 Driver Category list

Value	Category
3	Platinum
2	Gold
1	Silver
0	Bronze

IX.5 Cup Category list

Value	Category
0	Overall
1	ProAm
2	Am
3	Silver
4	National

IX.6 Session type list

Value	Session type
0	Practice
4	Qualifying
10	Race

X Changelog

X.1 1.0

Initial creation of the document

X.2 1.0.5

- Fixed glitched document reference
- Added section “Custom cars”
- Added EntryList property `forceEntryList`
- Added EntryList item values `overrideCarModelForCustomCar` and `customCar`

- Added Settings `randomizeTrackWhenEmpty`
- Added “VIII.1” Result files; Leaderboard files are now deprecated

X.3 1.0.7

- Added config.json property `lanDiscovery`
- Added Setting `allowAutoDQ`
- Added Setting `shortFormationLap`
- Added Setting `dumpEntryList`
- Marked Event property `trackTemp` as obsolete
- Added Event property `postRaceSeconds`
- Updated section “Result files” to reflect the 1.0.6 changes
- Added admin commands /manual start, /manual entrylist, /debug formation, /debug bandwidth, /debug qos
- Revisited the Weather section to small corrections. 1.0.7 has a more reasonable range of weather randomness values, so 1-5 should be good to use, and 2-4 should feel quite plausible (on a low time multiplier).

X.4 1.1

- Added pitstop rules
- Added race control commands (/tp5 - /dtc)
- Added 2019 tracks
- Added 2019 cars

X.5 1.1.1

- Replaced `maxClients` by `maxConnections`
- Replaced `maxClientsOverride` by `maxCarSlots`
- Removed `spectatorSlots`

X.6 1.2

- Added additional BoP (admin cmd, entrylist, bop.json)
- Removed parts of the custom car configuration, as this is now easier using the livery editor

X.7 1.3

- Added IGT DLC tracks `kyalami_2019, mount_panorama_2019, suzuka_2019, laguna_seca_2019`
- Added `formationLapType` parameter
- Removed admin cmd /manual start

- Added freaking comma in example settings.json

X.8 1.3.4

- Fixed another freaking comma
- Added unique pit count and max player starts (for private MP with multi-pit-assignment) to track list
- Added experimental/not supported properties of the event.json
- Added `metaData` property in the event.json
- Added additional explanations to the event rules (for swap situations)
- Added and updated penalty admin cmds (/clear, /clear_all, /tpX, /dt, /sgX)

X.9 1.3.6

- Flipped values for 0/1 formation lap types

X.10 1.3.7

- Added assistRules.json
- Added driver swap chat command

X.111.5.0

- Added `carGroup` entry in setting.json
- Added GT4 car model numbers
- Added further explanations to the eventRules.json
- Added experimental tyresetCount to the eventRules.json

X.121.5.8

- Reworked the weather chapters to reflect the improved model and behaviour

X.131.6.0

- Added 2020 tracks
- Added 2020 cars

X.141.7.0

- Added British GT tracks
- Updated track grid limits

X.151.7.4

- Added `ignorePrematureDisconnects` in setting.json

X.161.8.0

- Updated car groups (Cup, ST is merged to GTC)

- Removed all “_2019”, “_2020”, “_2021” track suffixes, please just use the track name in any configuration now
- Added the BMW M4 GT3 entry

X.171.8.2

- Inserted important hint for the `isFixedConditionQualification` setting

X.181.8.5

- Added the `publicIP` setting in configuration.json

X.191.8.12

- Added the Challengers Pack Cars (Ids 26, 27, 28, 29, 31)
- Added the TCX car group description

X.201.8.15

- Added the US track details

X.211.8.17

- Split “configuration files” into “server configuration files” and “client configuration files”
- Added the serverList.json description

X.221.8.21

- Added Valencia
- Added the Ferrari 296, Huracan Evo2 and 992 GT3

X.231.9.3

- Added the McLaren 720S GT3 Evo 2023