ACC Server Admin Handbook

Version 1.0.7

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 Configuration files

The server is exclusively configured via JSON files in UTF16-LE format. 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".

```json
{
  "udpPort": 9201,
  "tcpPort": 9201,
  "maxClients": 30,
  "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".
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.

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.

```json
{
  "serverName": "Kunos Test Server #03",
  "adminPassword": " adminPw123",
  "trackMedalsRequirement": 3,
  "safetyRatingRequirement": 49,
  "racecraftRatingRequirement": -1,
  "password": "accessPw123",
  "spectatorSlots": 1,
```

```
    "spectatorPassword": "spectPw432",
    "maxClientsOverride": 30,
    "dumpLeaderboards": 0,
    "isRaceLocked": 1,
    "randomizeTrackWhenEmpty": 0,
    "centralEntryListPath": ""
    "allowAutoDQ": 1,
    "shortFormationLap": 0,
    "dumpEntryList": 0,
}
```

Property	Remarks
serverName	The server name displayed in the ACC UI pages
adminPassword	Password to elevate via "Server admin commands"
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".
spectatorSlots	Defines the amount of spectators that may join this server without owning a car; see "About Spectators"
spectatorPassword	Password to enter the server as spectator. Must be different to "password" if both is set.
maxClientsOverride	Possibility to override the "maxClients" value in the "configuration.json". Can only reduce that value. Use it if you e.g. rented a 24 slot server, but want to restrict it for some reason. Defaults to 30.
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. |

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.

```json
{
  "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.
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". Values (0.0, 0.1, …. 1.0)
rain	If weather randomness is off, defines the static rain level. With dynamic weather, it increases the rain chance. 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 (but less chaotic compared to versions before 1.0.7)
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

Sessions are expressed as an array of:

Property	Remarks
hourOfDay	Session starting hour of the day (values 0 - 23)

dayOfWeekend	Race weekend day: 1 = Friday, 2 = Saturday, 3 = Sunday; see "Race weekend 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

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 night 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. Low "weatherRandomness" enables a realistic evolvement while having a plausible and stable day. A simulation of 10 different days with a value of 3 or 4 could produce such rain flows:
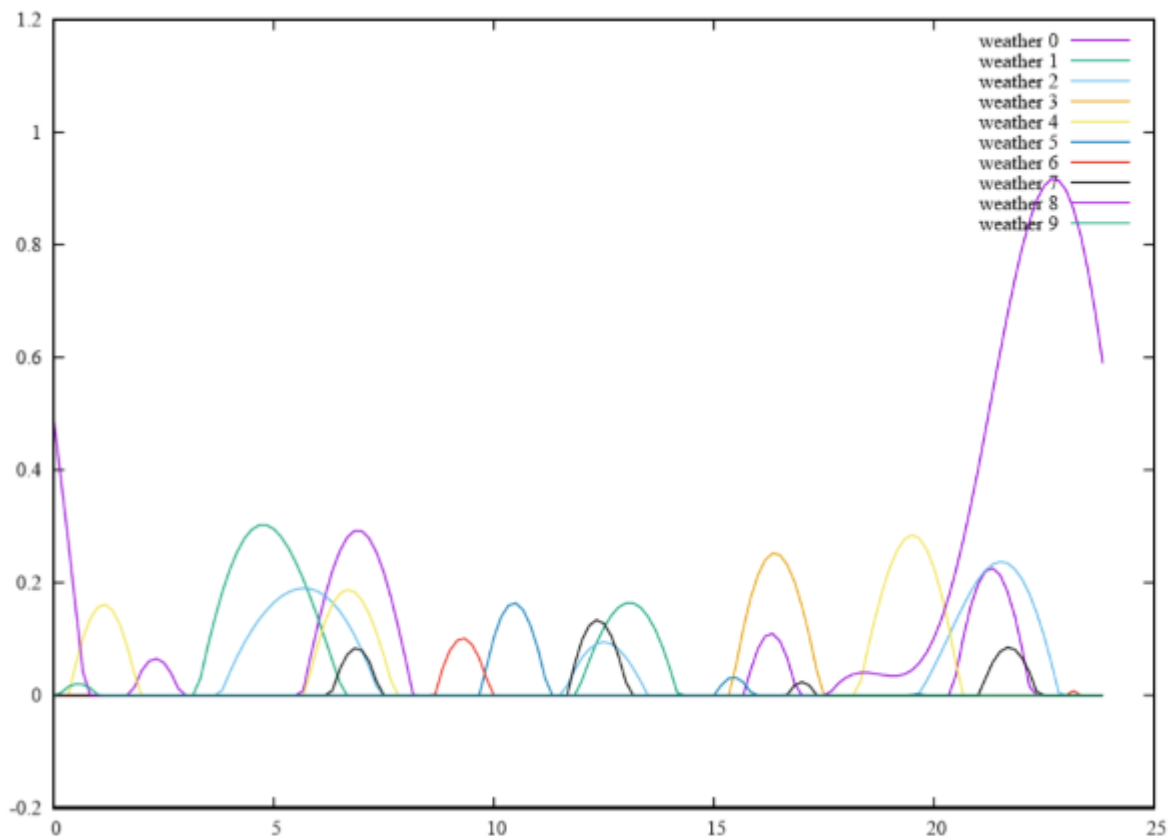

Figure 1:example calculation of 10 days with low/medium randomness; y = rain %, x = hour of day

You see most of the days have some rain mixed in but are dry in most of the time. That means the chance to witness rain during a 20-minute session is low but can happen. Extending the time by either longer sessions or higher time multipliers will raise the chance.

Also note that most rain showers are in the "light/medium rain" region, only one day delivered a full thunderstorm. This is where the baseline of both "cloudLevel" and "rain" play an important role; raising either will make rain more probable.

Raising the "weatherRandomness" towards the upper end causes more extreme scenarios, where the weather is changing a lot often per day, and is allowed to leave the baseline values more easily. A

more extreme case would be around values 5 or 6:
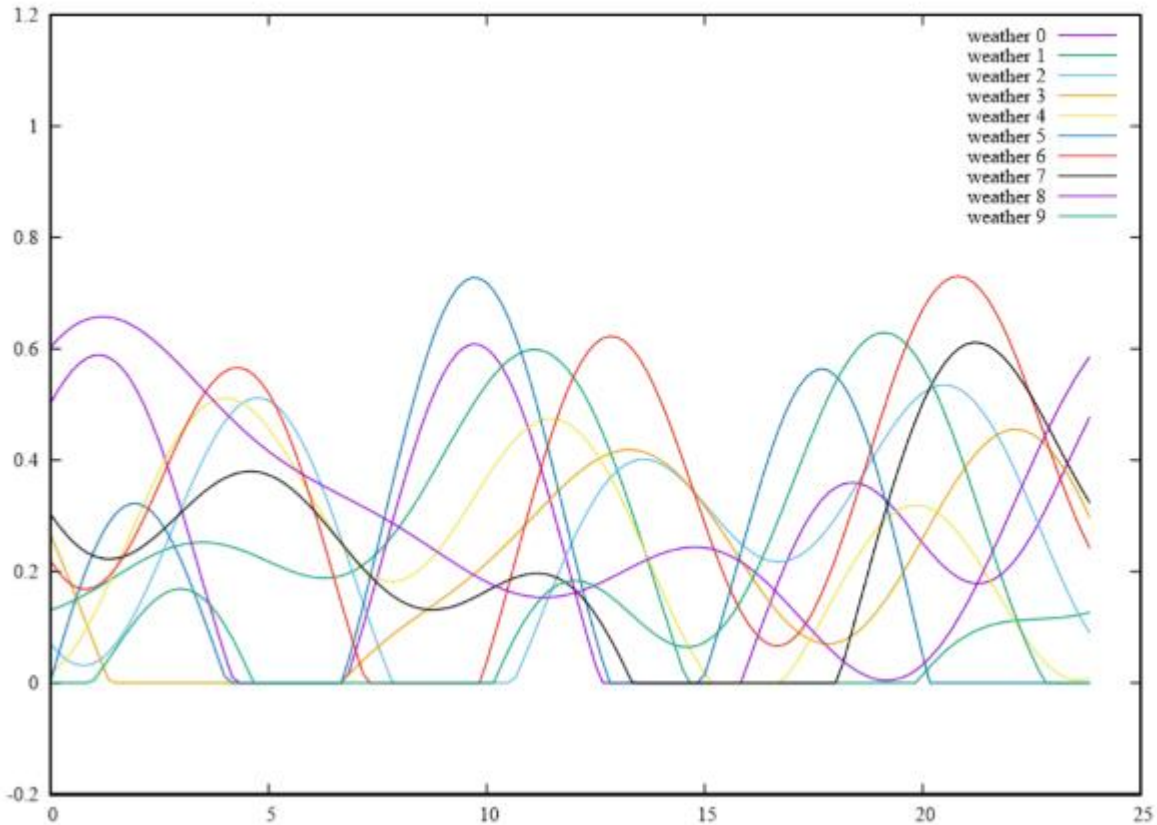
It's clear how both frequencies and amplitudes change. While this looks nice and sensational, we do not recommend to use extreme values. Even if there is no rain, a value of 2 will already notably shift around clouds, temperatures and wind – but in a way you can still understand what is going on while sitting in a race car. If the temperatures for example raise and drop by 5°C within minutes, most users will constantly have the feeling that something is wrong and they don't understand what it is.

So please feel free to experiment with those values and create a cool server setup, but keep in mind that it doesn't have to reliably rain and stop twice within a 20 minute race session, and that users have to have a chance to follow the weather development.

IV.2 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)

/manual start	When applied during the pre-session phase of a race, it will remove the automatic throttle for the next formation lap. No rules apply, so they have to be sorted by the drivers and race director
/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.

VI Entry lists

Using entry lists a server admin can setup 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".

```json
{
  "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,
      "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).

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 setup 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:
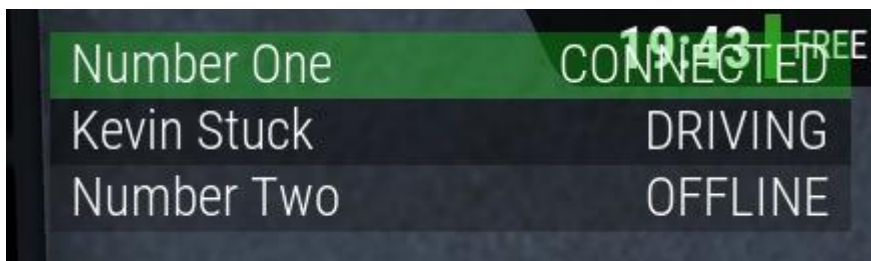
Using the Pit strategy page in the (M)ulti(F)unctional(D)isplay, the driver can assign a driver swap which will be automatically executed during the next pitstop.

VI.2 Custom cars

Please note that custom cars are not yet officially supported, and thus the features of custom appearance are not convenient to use.

Still, you can define custom cars for entry list entries, which will overwrite anything the user chose. That way you can either give your admin entry a cool and unique appearance, or make sure your league members run the correct car for this season, and have consistent appearance and race numbers.

Given the "exampleCar.json", the file needs to be located in cfg/cars/example.json (or customEntryListPath/cars/example.json) and may look like this:

```
{
  "raceNumber": 123,
  "auxLightKey": 1,
  "auxLightColor": 67,
  "skinTemplateKey": 102,
  "skinColor1Id": 56,
  "skinColor2Id": 298,
  "skinColor3Id": 221,
  "sponsorId": 15,
  "skinMaterialType1": 0,
  "skinMaterialType2": 0,
  "skinMaterialType3": 0,
  "rimColor1Id": 284,
  "rimColor2Id": 321,
  "rimMaterialType1": 2,
  "rimMaterialType2": 2,
  "teamName": "My custom Team",
  "displayName": "My custom Team",
  "competitorName": "My custom Team",
  "carModelType": 2,
  "cupCategory": 0,
  "useEnduranceKit": 1
}
```

The skin templates and colors are not yet a supported feature, but clever users already figured out quite a lot about the json struct, so this may be useful:

https://www.assettocorsa.net/forum/index.php?threads/car-customization-sponsor-ids.57877/#post-1059085

When joining this server (as the user with the entry list item for this exampleCar.json) while picking a Bentley, it will load the carModelType 2 = Ferrari and apply the custom appearance:

Setting overrideCarModelForCustomCar to 1 will allow to use the chosen car model, while still applying the appearance values:

VII About Spectators

To join a server as 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
```

```json
            },
            "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

Value
monza
zolder
brands_hatch
silverstone
paul_ricard
misano
spa
nurburgring
barcelona
hungaroring

IX.2 Car model list

Value	Car model
0	Porsche 991 GT3
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 991.2 GT3 Cup
10	Nissan GT-R Nismo GT3 2017
11	Bentley Continental GT3 2016
12	Aston Martin Vantage V12 GT3
13	Lamborghini Gallardo R-EX

14	Jaguar G3
15	Lexus RC F GT3
16	Tba.
17	Honda NSX GT3
18	Lamborghini Huracan SuperTrofeo

IX.3 Driver Category list

Value	Category
3	Platinum
2	Gold
1	Silver
0	Bronze

IX.4 Cup Category list

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

IX.5 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).