ISIMOTOR2 AI GUIDE

By: Exefune

First to all, I want to thank to the community of Rfactor, GTR2, Race 07, F1 99-02 Challenge, and Automobilista 1, thanks to all the investigation and work, it's where I found all of the information to make this guide.

Also I want to thanks now to people that helped me directly or indirectly, such as Shovas, The iron Wolf, TShirt, Speednut357, MJQT, jgf, Matthew Sibanyoni, Niels Heusinkveld, Gabrielxdxd4, barryg, VonDuth for its tutorial. And the people that tried my settings and informed such like gt_gamrx22, Iker Vera and Mitzu Community by people like sigmaengine2099, xplotao and Mitzu himself

Thanks to Crimson-764 for fix all my English error (im not native sorry)

Really, thanks to all of you, this it's possible.

Now we start with this tutorial and documentation:

ISIMOTOR AI its divided into a lot of files, there values spreaded everywhere so maybe if you want to do something, maybe only one value can`t help you, so in this documentation, its gone divided into different section with each file

Also, am gone try to cover all variables that are general in isimotor2 games, but some variables are exclusive to X games, so for that, I divided into some names

SB: all Simbin Games
RF1: Rfactor 1
AMS: Automobilista

F1C: F1 Challenge 99-02

M/S: meter per second

Km/h Kilometer per hours

This it’s all the files that interact with the AI

PLR>AIW>HDV>TBC>RCD

And some other indirectly like GDB, Veh, etc.

PLAYER FILE (.PLR)

First, we start with the “most” simplest one, here its defined all user options but have some hidden AI settings that generally you can access to the in game menu, so we gone go here and change to our liking

generally its under UserData/YourProfile/YourProfile.PLR

Open it with any text editor like notepad

Now we can start (i recommend use ctrl+b to search each line we gona adjust)

QUICK AI Driver Strength: it’s just the AI driver strenght that you can found in the game slider

Ai power calibration: this is not knowed what exactly do. There it’s a theory that say this influence the factors depending on the % of the strength, i recommend leaving in 7 or in 4 or in 0

Ai to Ai Collisions: This one calculates the collision of AI. At higher values, it makes the AI have less contact. I recommend a value between 18 and 40 (although Automobilista use 80 value, its unknown if possible to use this 80 in other isimotor game). In some cases, a high value makes the cars too swervy because they want to avoid collisions, so it's better to have a value between 18 and 30 for those cases but you can use the number you want.

From VonDutch:

Player Car Equal= 0 means the AI cars will see more variance from 1st through last. If you set it at 1 the AI will see less variance between the fastest and slowest cars.

AI Realism: this its the aggresion slider basicaly, higher values, more aggresive AI, this its recomended to touch in GTR2, VonDutch recomends 1.30000. in other isimotor games you can just touch the aggresion slider and gona work fine depending of your RCD files

CrashRecovery: this one make your flip your car when upright, also make AI stay more glued to dry line (accordin to F1C community). In rfactor and AMS its recommended to leave it in 0 (for more realism) or 2, in other games leave it to 0 (if you want realism). Also atleast in F1C and GTR2 if you have set in 1, when a AI car become upright, after being flipped the driver gona DNF the race.

Speed Compensation: seems make artificialy more close racings, its not very know how work, only some information can be graben from rfactor that say have a range of 0-10

(Exclusive to RF and AMS): Speed Comp Dist="500.00000" // <= 2.0: fraction of track length, > 2.0: actual distance in meters for max speed comp.

Autocalibrate AI Mode (only in F1C and SIMBIN games, Rfactor and AMS have totaly diferent purpose so im gona explain in an exclusive section): basically, make the AI go more close to player performance, its seems use your qualifying times to calibrait and seems only work when Strength<100 (further test recommended).

Following variables can be overwriten in rfactor and automobilista in .hdv files, sadly other games requires you to adjust here in .plr so do with caution

AIFuelUsage: this affect how much AI use fuel, its better leave in 0.90-1.00 (I generaly stick around 0.97 or 0.99) but if you use cars that take a lot of fuel like Turbo cars its better use 0.3 or 0.4

AI Brake_Power Usage: high values make more lately the AI brakes.

AI Brake_Grip Usage: when more high value more hard the AI brakes

As ISImotor AI web say: Increase (decrease) when AI brake too early (too late) before corners

VonDutch Recommendation (if you want to know the opinion of him): AI Brake Power Usage="1.20900" // Fraction of theoretical brake power that AI attempt to use // This one is important. By raising this number above 1 it will help to get rid of the special brakes the AI seem to have. It fools the AI into thinking that they have more brake power than they do. I have gone as high as 1.5000 but back off when I see the AI braking too deep and crashing. This will vary from track to track.
AI Brake Grip Usage="0.95550" // Fraction of theoretical brake grip that AI attempt to use // this is the amount of braking grip that the AI will attempt to use, the lower this number is the longer it will take for ther AI to slowdown. Conversely the higher the number the shorter the distance for the AI to slowdown. Used in combination with the AI Brake Power Usage figure this will really allow you to tune the AI to brake more humanlike. (Should be always lower or Little lower than Brake Power Usage)

AI Corner_Grip Usage: how fast they gona try to take a corner. as ISImotor AI web say: Increase (decrease) when AI are slower (faster) around corners than they should be

A good practice by Niels, its to set AI brake and AI corner using Motec or a telemetry tool, put the AI to drive your car in a track with good AIW and then see how they brake, take the turns and accelerate to adjust each value to be more close to the humans.

AI Max Load: Quote originated from ISI "The default value above define the upper bound of the range of the force in Newtons that the car will exert on the road. This value is used only to create an internal table for the physics system.".

Now in practical effects, this affect how AI calculate they braking and also how they try to use the track when overtake other cars. I recomend only leave vanilla number but you can try to decreasse a litle like to 37000. I made the following formula with help of RF2 spreadshet and Chatgpt so this mean it’s not perfect but i hope can help to modders that want a more exact value for them

((Mass+Fuel)*9.80665)*2.5= AI MAX LOAD
Rembenber that fuel, each liter weight 0.75

(In Case of games that not allow put those lines in individual HDV, use the more weighty car of the game)

In GTR2 the saleen it’s the more weighty car, and its tank capacity its 100 liters

So we do this
((1250+ (100*0.75)*9.80665)*2.5=32484.5281

And now I round the number up
32500.0000
In case you notice the AI start to go out of track or brake too late or youd don`t like the behaviur after adjust this, sum +5000 to the result until they stop to go out of track

AI Min Radius: The Tightess that AI can turn**,** should be leave as default but if you are wiling to put more precise value, you can use this formula from RF2, help to improve AI Lap Times.

Wheelbase*5.5= AI Min Radius

In gtr2 i use viper as an example
2.880 × 5.5 = 15.84

So GTR2 best value its 15.84
(in case you can`t use hdv values (like SIMBIN games), use the car with higher Wheelbase)

Following values are exclusive to RF and AMS

AI Limiter: lower values, make the race more spread based in RCDs, meanwhile higher values make more close races. I recomend using very low values like 0.01000 to 0.50000, also take in account, when its set 0.0, this gona make them race always fast and make the MinRacingSkill from RCD pointless.

AI Mistakes: this make AI make more mistakes like missing braking points or taking wide a turn, i recomended very very low values like 0.1 or 0.01, due that still AI can commit some hardcore mistakes into the .exe.

Auto Line Smooth: this gona sound ilogical. But please put this line in 0, this was in reality a dev tool and should be used with caution, i don’t know why ISI leave this in 1 from default, but generaly make race really boring, 3 meanwhile make entertaining but with a lot of problems, im gona leave Mike Z explanation of this althrought come from a RF2 Forum

“This parameter is really intended as a development tool. Yes, we enable it for our public gold release... but it's operation should be considered "unsupported" for reasons I will get to in the long explanation. It's purpose is to give the track designer a tool to possibly smooth out an un-optimal path....which he would then save and review. It is possible that applying this algorithm will actually make the AI drive the path slower/worse
When we load up our tracks and cars, we have EACH CAR run a "simulated" lap to gather our best guess information about what lap time he should expect and min and max speeds he should expect as he (either player or AI) travels around the track. If this parameter triggers the smoothing code, we then, for each path specified, make a list of all waypoints along with the expected speed OF THIS PARTICULAR CAR on that path. We order than list from slowest to fastest and then go through, waypoint by waypoint and adjust it's position by a couple of centimeters left and right laterally along the path to see if that raises or lowers the speed of THIS PARTICULAR CAR through this waypoint, the waypoint before, and the waypoint after the current waypoint. We move the path to the position that "optimizes" those 3 speeds and then we move on to the next slowest waypoint in our list... and on and on until each waypoint has been processed. Obviously, this is not a bulletproof algorithm (I'm sure you can already see plenty of issues with it). For one thing, if you are racing multiple car types in your race, it would "optimize" the path for each car type...one after another....I could see the final result mutating beyond recognition, so be especially wary of using it for multi-class racing.”

As he explained, this should be used with caution by track designers (i need to test this if work with the AIWCAM versión of RF1), only in some tracks i found 3 make race much better, i thinks its due how they build the left and right patch, but if not the case then should be 0. Also maybe can help to your .ini files from Autocalibrate AI but more testing it’s needing if affect those files.

Setups lines: There are some lines regarding about AI using setups, jgf explained how you can set this for make AI use your setups in gtr2 and how make AI in rfactor use your defined setup without moving files (this last we gona see in the setup section)

“For getting the AI to use specified setups in GTR2, in addition to copying, renaming, and moving the svm files, you must edit this in the plr file:

Vehicle Specific AI Setups="1" // If 1, search specific vehicle directory for "<track>.<qual/race>.svm" or "<track>.svm" or "ai.svm" before falling back on game.gdb defaults

Not sure about this line but many sources say it must also be enabled_
Default Player Setup="1" // If 1, attempts to ensure player has default setup

In rfactor you must also enable this in the plr file: //use this if you want them to use your personal setup without brothering moving files, If you want more exclusive setups go to Setup Section that its almost below of everything

Fixed Setups="1" // use fixed setups specified in UserData
Fixed AI Setups="1" // whether AI use the fixed setups, only applicable if "Fixed Setups" is also enabled (and can be used in single player to have the AIs use your favorite setup)

And when you create the setup, select "set as default". You don't need to rename or move files anywhere.”

So this cover all PLR Section. We can move to:

Driver TALENT FILES (.RCD)

Now, this is going to be divided into 3 sections: The New RCD system (Rfactor 1 and Automobilista). The Old RCD system (F1 99-02 Challenge and Simbin) and NASCAR Games. Since they use different ranges and different variables, although it seems you can use variables that appear in F1C in Rfactor and vice versa, this is not confirmed due to a lack of debug tools.
Also something to keep in mind is that the AI always has a lower random factor when you load a session, so you will see that sometimes the AI can be a little better or a little worse than what you defined, also that they lose pace in the middle of the race (caused by the aeromin of the damage file and by composure). And for last, its not important the file name.rcd (don't use spaces) but its very important the first line of the rcd as should be same than the driver line from.veh/.car file. Example I have MSchumacher.rcd and have in first line Michael Schumacher and this Name its same in the .veh file in the driver line, then the game gonna load the rcd correctly.

New System RCD:
(thanks AMS and RF comunity)

Aggression: As this number increases the AI will give other cars less room, both while passing and while following. It also increases the frequency at which they try to pass and the increases the threshold they are willing to endure before giving up a pass. This interact with Aggresion slider UISlider*Aggresion Recomended values by AMS: 60 and 80, but i feel you can play with driver around 80-90 range if you want more fights (useful for Stock Car Racing, can be used for other series but have caution) and further 90 or 95-99 should be, "peculiar drivers looking a gap always or not caring of crash"

Composure: defines the frecuency of "bad zones" and also frecuency of mistakes (if AImistakes>0), lower values make worse driver (also lower value make them worse in they throtle), higher value make them recover faster from bad zones also (which can be caused by being overtaken) recomended ranges: 66-98 but AMS recommend stick around of 90 for average drivers.

Speed= the general speed of AI, very important, interact with the AIrange from AIW that from default are 0.1, so your speed values are limited with a 10% range. Unless you increase to 1.0 making speed depend 100% on rcd value. AMS range is 65-99 but remember how you set up the AI range, as if you use higher AI range or 1.0 then better to use a more close range like 80-99

Courtesy: (thanks SpeedNut) How respectful the AI drivers are when being overtaken or attempting to overtake others. Lower values make them leave less space, often resulting in contact or crashes. Higher values make them more cautious and more likely to avoid collisions. In AMS, this also interacts with the new blue flag behavior making them react more quick to blue flags. A good practice is to assign higher courtesy values to slower drivers so they yield more easily, while faster or top drivers should have lower values to encourage more aggressive overtaking attempts against slow cars (but not too much).

Crash: % of times a driver gona crash in the race, only used when you simulate the race (I still recommend to set up this)

Recovery: its seem this line it's just a left over, according to Niels this don't seem to be readied in Rfactor, i tried to see if in GTR2 and F1C but i dont see any change. Some people in AMS claim this make AI try to avoid any pontential crash but dont seem the case either. Other claim AI try to regain is position back but i really doubt about this, leave this in 5, 15, 60 or 99 unless you found what the actual purpose of this line.

Reputation: Thanks to **Niels Heusinkveld**. This line along with Speed Calculate a Intimidation value, its seem higher values make the AI try to defend more his position, also lower values make them a little more cowardly to overtake. Stick around 99 for top drivers, 90 for average and 70-90 for bottom drivers. (But you can play with this range as you like because there not a lot of information avaible)

CompletedLaps: only used when you skip sessions, i just stick around 80 to 100%

MinRacingSkill: 0-100% When the value of the AI Limiter variable (from the player file) is > 0.0, the AI drivers go through cycles of optimal and suboptimal driving where their driving skill drops to MinRacingSkill * Speed. (Set the AI Limiter variable to 0.0, and the AIs will always drive as best as they can... on every lap.) So, for example, if you have a driver with Speed = 50, an AIRange in the AIW file = 0.2, and MinRacingSkill = 90, this type of AI will generally drive at (0.9 + 0.2 * 0.5) = 90% speed (really very, very slow... but this is just an example). However, sometimes it will drop to (speed * MinRacingSkill = (0.9 * 0.9) = 81% when having a bad lap... that's extremely low, but this is just an example with easy numbers. Stick around 95 value and dont go below of 80 because gona make AI exagerated slow when they are in “bad zone”

AMS range if you have curiosity: 60-99

QualifyingAbility: This actualy work in rf1 (posible in AMS also) and basicaly its, when you skip the qualy sesión, this gona calculate the average position of the driver, lower values make AI being generaly in first positions meanwhile higher make start more from the bottom (thanks to Whils)

AMS ONLY (thanks isiMotor2 Community AI Documentation)

TireManagement:

The lower the value, the more the AI will wear its tires.

Suggested range: 40-95 (80, 85 and 90 are the most common values)

StartSkill

The lower the value, the more chance AI driver will bog down at the start.

Suggested range: 20-90, being around 70 the most commom value. If you set this under 60, the AI driver will problably bog down very often.

Now this it's the old RCD system used by F1C and SB games. First we gona see GTR2 RCD and then we gona see F1C rcd (due that it's more simple), im gona use VonDutch tutorial as a base (Green lines) because he explained really well but im gona add some comment with // example

```
        Henrik Roos  //same name as in Driver from .car file
{
 //Driver Info
 Abbreviation=H. Roos  // Driver Name
 Nationality=Swedish   // Self explanatory
 NatAbbrev=SWE         // Self explanatory

 //Driver Stats
 StartsDry=-1.0          // This is the average number of places gained or lost at the Start,
positive #'s are places gained, negative numbers are places lost.
 StartsWet=1.0           // keep these numbers at -1, 0 or 1. This applies to both wet and dry
starts. This helps cut down on the chaos at a start. Far fewer crashes.
 StartStalls=0.0         // % of time that the driver stalls his car at the start //dont go very far with
this, its for simulate Stalls in the grid like hapened often in F1 in 2000s era
 QualifyingAbility=4.00  // This is the average qualifying ability of the driver. The number ='s
position. //however only seems used when you skip the qualy
 RaceAbility=0.70        // The lower the number is the faster the driver is. The range is 0.0 - 6.2
//actualy you can go beyond from 6.2, but you are risking that AI dont uses the full throtle,
especialy when you set more high than 15.00
Consistency=2.0          // The lower this number the less variation in lap times the higher the
number you see more variation in laptimes
 RainAbility=1.00        // Drivers ability to race in the rain. The range is 0.0 - 6.2//should be
always same or litle higher than RaceAbility, never should be lower because gona make really
OP in driying conditions or risk more crashes, Driver that are great in rain should have same
value as RaceAbility, normal driver should just sum a number less than 1, for example in this
```

case, 0.70 +0.30=1.00, and bad driver in rain sum a number higher than 1
taked from: https://cluisparacosm.blogspot.com/2012/10/tesis-tutorial-como-crear-archivos-rcd.html

Passing=97.0 // % of times driver completed a successfull pass, not including pit stops or lapped traffic. Raising this number helps the AI pass more.

Crash=1.41 //% of times driver crashed //used when you skip the session,
Recovery=94.8 // amount of times driver carries on after a crash // not know exactly, same theory as the new rcd system or it's just a leftover.

CompletedLaps%=99 // Self explanatory //only used when you skip practice/qualy

Script=default.scp // refers to a script(????) unsure on this have not modified//refers to a old script that appeared back in F1 games.

TrackAggression=0.99 // this is IMPORTANT. This numbers needs to be at least in the hi .70's for even the slowest/worst drivers. Higher for the better drivers this is a important key to getting the AI to pass each other and you.//you can use higher numbers than 1.00 but be beware can break the AI, so use Von Dutch range from 0.70-1.00

// Increase attempted low-speed cornering by adding a minimum onto calculated speed.
// Reduce attempted high-speed cornering by multiplying speed by a number less than 1.0.
// <adjusted speed> = CorneringAdd + (CorneringMult * <original speed>)

CorneringAdd=2.1 // Increase attempted low-speed cornering by adding a minimum onto calculated speed. Don't go too high with this number, make small adjustments from original values until satisfied

CorneringMult=.974 // Reduce attempted high-speed cornering by multiplying speed by a number less than 1.0. Same with this, don't go to high here the higher this number and the previous one the faster the AI will be up to a point then they have trouble staying in control of their vehicles. So small adjustments are best until you reach a point where you are satisfied. // i always try to leave Corneringmult alone or being almost all same for all drivers, then you can increase a little bit the cornering add if you want more slower driver, faster driver should have more lower cornering add value but have caution with your adjusting as Von Dutch says.

And this are comments from jgf, thanks to him.

"The Cornering Add and Cornering Mult can be very useful when dealing with mods much different from what the sim was designed for. For example, GTR2 tracks are designed for modern GT cars, using a well designed mod of older cars will have them attempting to run like modern cars ...and probably spinning out too often. The Cornering adjustments can compensate this; you may have to balance this with race ability to get realistic lap times. It is also easier to use this than to tweak the AI braking parameters, which affect all cars, to get the AI to behave in turns. (Note that this will affect this driver at all tracks, so is not a solution to one turn or one track, which is an AIW issue.)"

//AI Throttle Control - how good they are at their own traction control upon throttle application //I have not experimented with these//me neither but here the jgf explanation and its seems its the more accurate

TCGripThreshold=0.7 // Range: 0.0-1.0
TCThrottleFract=1.2 // Range: 0.0-???
TCResponse=0.0 // Range: 0.0-???

From jgf: Threshold seems to be at what point in the tire slip range they begin to modulate the throttle; "Fract" is how much they modulate; and "Response" how quickly they modulate. But this is conjecture based on my tweaking, and is not entirely consistent across all drivers/all cars.

//AI skill mistake variables
MinRacingSkill = 0.98 // this refers to the minimum amount of skill that the driver will run at //same as i say in the new RCD system, stick around 0.95 value and dont go below of 0.80 unless you want drivers really going slow

Composure = 0.002 // The maximum amount of time the driver will make a mistake // Lower values make AI enter more frequent in bad zones, make lose more pace when the race

continue, make more mistakes like sbins, and for last take more time to AI to recover its pace after being overtaken. I Stick around 0.001-0.099 (0.0xx range) but you can use higher number if you wish.
//AI ColdBrain variables   // "Cold Brain" refers to the begining of a Race or Qualify session It tries to simulate the AI on cold tyres
RaceColdBrainMin=0.94   // This is the minimun amount of speed (based on a figure calculated by the game) the driver will run in a race
RaceColdBrainTime=100 // This is the amount of time the driver will run that speed in a race
QualColdBrainMin=0.94   // This is the minimun amount of speed (based on a figure calculated by the game) the driver will run in qualify
QualColdBrainTime=155 // This is the minimun amount of speed (based on a figure calculated by the game) the driver will run in

}

So remember talent files are one of the important parts to getting faster, more consistent AI and a better game experience. Use this as a guide for creating your own talent files or modifying others. Try to keep all of them fairly even, it helps to not have huge differences between the AI drivers. Time to move on.

F1 Challenge 99-02 RCD System (It is recommended to read the GTR2 System as it shares many similarities)

```
Michael Schumacher //same name as in Driver from .veh file
{
 //Driver Info
 Nationality=German
 DateofBirth=3-1-69
 Starts=161
 Poles=43
 Wins=53
 DriversChampionships=4

 //Driver Stats
 StartsDry=1.3           //Average number of drivers passed (-4 - 4), Same as GTR2, negative number= places lost.
 StartsWet=1.9
 StartStalls=0           //% of starts where driver stalled //self explanatory, use with caution
 QualifyingAbility=2.0   //Average qulifying position. Used when you skip the qualy session.
 RaceAbility=1.8         // The lower the number is the faster the driver is. The range is 0.0 - 6.2  //actualy you can go beyond from 6.2, but you are risking that AI dont uses the full throtle, especialy when you set more high than 15.00
RainAbility=1.5          //should be always same or litle higher than RaceAbility, never should be lower because gona make really OP in driying and wet conditions (more risk to crashes in rain due to perfoming much faster also), Driver that are great in rain should have same value as RaceAbility, normal driver should just sum a number less than 1, for example in this case, 0.70 +0.30=1.00, and bad driver in rain sum a number higher than 1.
Taked from: https://cluisparacosm.blogspot.com/2012/10/tesis-tutorial-como-crear-archivos-rcd.html
 Passing=88.0            //% of times driver completed a successfull pass, not including pit stops or lapped traffic //A higher number causes the AI to start swerving and attempting to overtake sooner.
 Crash=5.7               //% of times driver crashed //used when
you skip the session, still I recommend to set up this line.
 Recovery=0.1             //% of times driver continued after a
crash //mistery line that its not know its actual purpose
 CompletedLaps%=90 //used when you simulate the session.
 TrackAggresion=1.4 //recomended range 0.70-1.5, more test
recommended. Controls how much the AI defend his position and
how much space left to other car, help to overtakes too. Due
that F1C its limited to 100% aggression unlike GTR2 that allow
higher values, we can use higher number than 1.0 (but used it
with caution and maintain with the range i explained. Better If
we stick around 1.0 for TOP and MidTop driver) or don`t add this
line and let the game generate one for each driver
 Composure=0.010    // higher number make AI more stable, also
make them recover fast the pace after being overtaked, but I
recommend not add this line due that still it's a mystery how
work and can mess up some fined parameters

 // Increase attempted low-speed cornering by adding a minimum
onto calculated speed.
 // reduce attempted high-speed cornering by multiplying speed
by a number less than 1.0.
 // <adjusted speed> = CorneringAdd + (CorneringMult * <original
speed>)
 CorneringAdd=1.5
 CorneringMult=0.965
```

With this you can set up the speed of cornering of the AI, I recommend use the file from a similar rcd from your car, I stick with a same Cornering mult for everyone and make very small adjustment for CorneringAdd for drivers (slower driver have a little more higher value than faster driver). I recommend read the jgf comments in the GTR2 Section.

NASCAR Games RCD System (NASCAR Thunder 2003, NASCAR Thunder 2004 and NASCAR Simracing).

```
Dale Earnhardt Jr.
{
 //Driver Info
 DateofBirth=10-10-74
 TeamName=Dale Earnhardt Inc.
 HomeTown=Mooresville, NC

 //Driver Stats
 Speed=94.0 //same as in rfactor, if you have AI range in AIW in 1.0 then use
80-90 range, if you leave to 0.1 then use 65-99 range.
 Concentration=95.0 // refers to the composure line, lower values make AI enter
in bad zone more frequent and commit more mistakes.
 Fitness=98.0// MinRacingskill, when AI enter in bad zone, its speed it's reduced
to speed*fitness. So for that should not be lower than 80 (I stick around higher
than 90)
 Intimidation=99.0 // its seem to control how much he block other driver when
triying to pass him
```

```
 Aggression=93.0 // should be 80-90 for mid pack and some top driver, top driver 90-95 and higher if that specific top driver its a trouble maker.
 Status=98.0 // same as reputation in rfactor, higher values make him block more other car (interact with Intimidation) and try to overtake more
 Courtesy=60.0 // how much space he left to other car when overtake and brake, TIP: lower speed driver should have higher value.
 CompletedLaps%=90
 Script=default.scp
}
```

Finaly we ended RCD part, so now we can jump to the core of the car:

HIGH DEFINITION VEHICLE (.HDV or .HDC)

This file define the majority of physics of the car, but we gona focus only in the AI

Im gona use some MJQT and VonDutch bases in some variables and add my comments with //example

AIMinPassesPerTick: higher number make more realistic AI suspensión simulation but at the cost of more CPU Usage. Should be betweben 4 to 8 for most cars, i generaly stick around 6 for my preference (4 its fine for almost all softly sprung cars, but for cars that use a lot of Aero like F1 or hare really Stiff then its better increasing it). But never use exaggerate values because gona use a lot of CPU, and by consecuente AI actualy become worse (and dont use 0, this make the highest calculation but really make CPU work harder)

AIRotationThreshold: Incrase AIMinPassesPerTick if the car exceed the yaw rate, I only leave a value from a similar car but according to some devs; if you use **AIMinPassesPerTick** in 4 or higher its completely useless and should be leave in 1.0.

(another opinion): Acording to **SpeedNut**.: “*Refers to the threshold for the angle of the car's rotation in the air at which point the game engine will consider the car to be spinning out of control and trigger a crash or a spin. When a car becomes airborne, either due to hitting a bump or ramp, or due to a collision with another car, it may start to rotate or spin in the air. The AIRotationThreshold parameter determines how much rotation is allowed before the game considers the car to have lost control*.”

I recomend use a value from a similar car that have good setting

AIEvenSuspension: Averages out spring and damper rates to improve stability (0.0 - 1.0). The use of this variable would increase the disparity between AI and player physics. // I always try to use 0.0 because gona make AI have some hacks for make them more stable, increasing the disparity physics than player, use it only if you don't have any other option

AISpringRate: Spring rate adjustment for AI physics is used to soften or stiffen the suspension for AI vehicles. Decrease this to soften spring rates to keep the AI stable. 1 is equivalent to the setup file or default setup. //I try to use 1.0 or 0.9 to be more similar as player unless AI it's really unstable, then use 0.5

AIDamperSlow: Contribution of average slow damper into simple AI damper. This simplifies dampers when used for AI. //act when AI have weight transfer like when they brake or steer

AIDamperFast: Contribution of average fast damper into simple AI damper. //act when the weight move really fast like when they hit a bump

Lower values make them be more softness and 1.0 make them be same as player setup.

You should use a value from a mod that work well, i try to only touch the dampers one so they are more close to human driving, but if dont work well, you can touch the AISpringRate and AIevenSuspension.

From Yoss from AMS comunity:

AIDownforceZArm: Hard-coded center-of-pressure offset from vehicle CG.
AIDownforceBias: Bias between setup and hard-coded downforce value (Range: 0-1). 0 is most realistic, 1 is least realistic. A value of 0 will ensure the AI do not have more downforce than the player, this can be useful to improve AI stability though, especially if they have problems at high speeds.

As say, this make AI more stable at high speed, but the best way its make proper setup and leave it a 0.0

Examples values

Gran Prix Evolution 1998

```
AIMinPassesPerTick=7
AIRotationThreshold=0.10
AIEvenSuspension=1.0
AISpringRate=0.5
AIDamperSlow=0.4
AIDamperFast=0.0
AIDownforceZArm=0.05
AIDownforceBias=0.0
```

Reiza V8

```
AIMinPassesPerTick=6
AIRotationThreshold=1.0
AIEvenSuspension=1.0
AISpringRate=0.6
AIDamperSlow=0.6
AIDamperFast=0.0
AIDownforceZArm=0.0125
AIDownforceBias=0.00
```

Reiza Boxer (this one its curius the negative AIDownforceBias)

```
AIEvenSuspension=1.0
AISpringRate=0.6
AIDamperSlow=0.5
AIDamperFast=0.0
AIDownforceZArm=0.0
AIDownforceBias=-0.15
---------------------------------------------------------------------
```

AITorqueStab: The numbers represent how much force is needed to deflect the car from its course, longitudinally, laterally, and vertically

jgf recommends:

Run some practice sessions with the AI in the same cars as you and do a little "paint swapping"; if your car reacts more than the AI, lower the values, if the AI react more than you, raise the values. Set this too low and the slightest nudge will send the AI out of control.

// Follow the jgf recomendation, but remenber this also interact with the previus values we touched

From isiMotor2 Community AI Documentation

FeelerFlags=15: This will create the most possible accurate collision detection system, but you'll need to use the collision feelers. This part would be optional, since you'll need 3DSimEd to get the real car dimensions for each competitor. But, if you willing too, it's a worth effort.

Procedure to follow to improve collision detection of AI cars:

> Delete the // (comment marker) from the collision feelers lines and remove the line FeelerFlags=X from the top of it, since it's already informed at the beginning of HDC file.

You'll notice there's a bug in most of cars I've seen so far which is the wrong signal for FeelerTopFrontLeft, FeelerTopFrontRight, FeelerTopRearLeft and FeelerTopRearRight.

> Note that the coordinate system is strange (at least for me): +x = left, +y = up, +z = rear. The lines below are already fixed, so pay attention on it. Left is always positive and right is always minus!

To know the real car dimensions, you'll have to open it in 3DSimEd. Take note of dimensions, subtract 10 cm from each one (helps to prevent the AI to AI crash without significant visual glitches), change the values above (double checking the signals + and -)

- Remember to leave the rear-wing out of the box!
- It's not nice to see the car turning over based in the rear-wing position, it's preferable to leave the wing going through the tarmac than the strange "rigid rear-wing" effect when turning over

If you have patience enough, the amount of AI to AI crashes and the physics reaction will improve a lot by following this procedure. If you don't use feelers, the game will create a automatic box around the car which could be pretty rough for calculations.

The section of the HDC file relevant to collision detection might look something like this:

Example:

```
//FeelerFlags=15 // or remove this second instance or simply leave the
"//" infront
FeelerOffset=(0.0, 0.0, 0.0) // leave it alone, it won't be used.
FeelersAtCGHeight=0 // Set it to zero. You'll provide all real corners
coordinates for the car.
FeelerFrontLeft=(1.003,0.384,-2.554) //front-left corner collision
feeler
FeelerFrontRight=(-1.003,0.384,-2.554) // front-right corner collision
feeler
FeelerRearLeft=(1.003,0.384,2.586) //rear-left corner collision feeler
FeelerRearRight=(-1.003,0.384,2.586) // rear-right corner collision
feeler
FeelerFront=(0.064,0.384,-2.639) // front side collision feeler
FeelerRear=(0.064,0.384,2.587) // rear side collision feeler
FeelerRight=(-1.039,0.384,-0.247) // right side collision feeler
FeelerLeft=(1.039,0.384,-0.247) // left side collision feeler
FeelerTopFrontLeft=(0.478,1.540,-0.298) // top front-left collision
feeler
FeelerTopFrontRight=(-0.478,1.540,-0.298) // top front-right collision
feeler
FeelerTopRearLeft=(0.652,1.511,2.323) // top rear-left collision
feeler
FeelerTopRearRight=(-0.652,1.511,2.323) // top rear-right collision
feeler
FeelerBottom=(0.064,0.249,-0.247) // bottom feeler
```

Seem AI use the feelers to calculate they colision avoidance, also mods should need to have good coll files to help this.

RearBrakeSetting: define the brake bias, for example in 36 gona be 64-36 brake bias, why it’s this important? Because AI prefer more stable braking, so this can help AI to have more stable braking, but its not needed to change this in the hdv because you can make AI setups as we gona see in other section and in that place you can use a really forward brake bias. Also in Rfactor and AMS you can make exclusive AI upgrades where you can adjust this.

RFACTOR AND AMS VARIABLES ONLY

AISlipReaction= (50.0, 30.0)   // Prediction factor for front wheel grip loss (higher numbers increase sensitivity), how quickly AI increase throttle after grip loss has occurred //nothing new need to be added here. Test and see how AI behave, generaly faster car requires higher number but those values already work well. Lower the second number if you see the AI do a quick snap when they acelerate. Reiza default value seems to be: `AISlipReaction=(15, 0.3)`

AICornerReductionBase=120.0   // (pointspeed/this number) = % deceleration we can expect through a point // Prediction of how much AI need to slow down, also acording to some guys this interact with AIAimSpeedsPerWP but not confirmed yet. I recomend use this value as a default and play with it. (Generaly slower cars use lower values meanwhile higher speed car requires higher value)

AIAimSpeedsPerWP= (35,70,90,110,125,140,160,180)

// Speeds at which to look ahead X waypoints (spaced roughly 5 meters apart)

This one its tricky, basicaly make AI start to skip waypoint when they go more faster than the speed specified in each **Step**, each Step skip 5 meters of waypoint, we have 8 step, so its stepX5, first value represent 5 meter, second 10, third 15, four 20, five 25, six 30, seven 35 and eight 40 meter, this help to calculate they Throtle and braking and can improve lap times. **Its recomended use a value from a base car, and only finetuning it, for example if you notice AI cut too much a turn, then increase the first 2 values. The ones above are the reiza more commons value**

The speeds are in M/S so you might wondering, why almost all cars use exagerated highers values (making them never use those steps), its because (atleast my theory) guarantee that they dont gona cut the track if they reach that speed, more noticiable in oval tracks, generaly making them using only 4 or 5 of the steps instead of the 8 that allow to use. So you maybe want to choose the amount of step the car gona have allowed to use considering the speed they reach in corner (unless you want to do a lot of testing with the 8 steps, in this case make sure that the last 5 steps are really close to the maximun speed of the car to guarantee they dont use it in a corner, but have caution with this aproach, can end bad), the last step you gona use should be the maximun m/s speed of car including the Draft the rest you gona acomódate it according that speed can reach ensuring dont cut turns (remenber first two value should be configured for slow and mid turns). I highly recomend do two test, one in Monza or Silverstone style track (have high speed turns and some slow turns) and another in a oval fast track like Indianapolis or Talladega (if you consider the car can race in oval).

AIFuelMult: PLR file override for AI fuel usage - only positive value will override

AIPerfUsage: PLR file overrides for (brake power usage, brake grip usage, corner grip usage) used by AI to estimate performance - only positive values will override

AITableParams: PLR file overrides for (max load, min radius) used when computing performance estimate tables - only positive values will override.

Look in PLR section again to see how should be this adjusted, or leave it in -1.0 if you want to use default ones.

SpinInertiaAI: AI pitch inertia, used to deal with their lower physics sampling frequency, just paste the same value as SpinInertia for each suspensión, this ensure and force AI use same Inertia as player, if this dont found, then Game automaticaly *2 the Human Spin inertia, in some specific mods i see they do SpinInertia*8=SpinInertiaAI

BrakeTorqueAI: Different brake torque for AI, as they are not currently affected by cold or faded brakes, just use a Little lower value compared to player so they get a Little nerfed brakes. Example: BrakeTorque=3900 BrakeTorqueAI=3700

Tire Behavior Configuration (TBC/TYR)

I will be very brief in this part, as tires are a world of their own, so we will focus only on what is related to the AI. I must say that this is what I recommend touching the least, as it can lead to very poor results, so modify with caution.

Thanks to Yoss from AMS Comunity.

AISens: Simplified load sensitivity for the use of AI controllers. Linear instead of exponential like the player's car. It reduces the use of the CPU. Because this is linear, it would probably be prudent to establish this for a lower drop than the player's car, this is because most of the time is spent in the middle range of the loads instead of the maximum loads. This can be used to improve the stability of AI cars.

AIGripMult: Grip multiplier for AI. If the AI is too slow or spinning, these values can be modified to give them speed and / or stability. (1.0 Its same grip as player). I noticed some mods also decrease front grip and increase/decrease the rear one to alter the way the take the turns (for example lower grip seems to make them take turns more smoothly), so you maybe want to take this into consideration to adjust they grip.

AIPeakSlip: Slip of the peak AI tire, similar to LatPeak / LongPeak only for the AI. This does not change depending on the load, if the AI slips around a lot and is unstable, as a result, this value is reduced, especially in the back. Otherwise, if you do not have such problems, try to find the player's maximum average grade and use those values.you can also use more higher values to make AI go more faster or you want to force them to drift, but better leave same value as player or a little more lower (for wet tyres maybe can be good to leave it more lower to force them not drive to the limits)

AIWear: this define the constant of wear that AI gona suffer, should be less than player, generaly i see its player wear/3 or playerwear/4 depending of the car. But some exagerated case use player wear/10 (not recomended). I also found (thanks CMT) that Wear*0.29 give good results too. With the AIPitThreshold and the AMS variables of Tire wear in rcd, its better first use the human wear rate and then test when you need to pit and when AI need to pit in those games.

Softness: this tell the game how soft the tire are, calculate a DurabilityRating, its seem dont make anything, apart only work when you put a value betweben 0.5 and 1.0 (thanks to Niels for all this information). Any way softer tires should be always higher number, also 0.75 onward are configured for wet tyres generaly

Following values are Rf/AMS exclusive

AITireModel: 0.0 = original AI tire model in terms of slip, 1.0 = more similar to player tire model. Try to be 1.0 or 0.9 but if you notice problems, then leave it in 0.30 or 0.40

AIPitThreshold: this define when the tire have less grip (defined in WearGrip) than this value, then they should pit for new tyres, i recomend sticking around with 0.94 because it’s AMS recomended value.

AiHeatRate=6.6e-6: this line was hidden but discovered thanks to AMS community and Niels. This it’s the rate that define of how long take to warn them tyres. They start with the start temperature as same the player and have the same grip penalty (obviusly with they simplified physics), AiheatRate define how long take from start temperature to reach the optimun temperature. For now due that there not a lot of information I stick around the default value (I also read that it is measured in 1 m/s)

And with this cover all physics part of AI, now we can focus on other aspect

AMS EXCLUSIVE PARAMETERS

Really Thanks to Matt that he passed me everything he know, the best values and the information below here.

In HDV

AIQualLapsperRun=5 //laps he gona do in each qualy stint
AIWingDamageLift=0.30 //how much he gona lift when he lose his wing

AIPassLatScalar= 1.5 - Amount of extra space in meters left at 0 aggression when trying to pass. I personally do not use this, it is unnecessary.
AISuddenMoveScalar= 0.6 - (estimated guess) Amount of extra space in meters left a 0 aggression when trying to defend a pass.
AIStartOffset= 2 - How long in seconds before the AI will merge into the normal racing line during a race start
AIStartMerge= 8 - How long in seconds merging should take during a first lap

In SRS file (series file)

(Not recommended to touch the values from here)

OvertakingRadiusRiskBypass = 600 - radius in meter, from this radius and above the AI will be bypassing, below it will not
OvertakingWaypointsAhead = 15 - Number of waypoints ahead of the current one to calculate the radius

BlueFlags = 1

AIBlueFlagCurveRadius = 600 ////in meters, curves with a radius under this value won't affect AI under blue flag

AIBlueFlagCourtesyThreshold = 0.5 //// from 0 to 1 the amount of courtesy the AI need to lift when not obeying the blue flag is not punished. Blue flag is not punished with srs BlueFlag=1

AIBlueFlagReactionTime = 20 //// max amount of seconds for the AI to react to the blue flag

AILiftUnderBlue = 0.25 //// bonus of the throttle for the lack of courtesy of the driver to add to the final throttle of the AI

And now we can move to:

Artificial Inteligente Waypoint (AIW)

Ok, we gona cover some basic things About AIW because everything in AIW its related to AI in some way, especialy because this define the route AI gona take, the widht of road, ETC.
all waypoints things related should be covered in a apart tutorial, also maybe some things dont gona work in x game, the information it's very spraded and what im saying, maybe dont gona work in your favourite game. So we can start

(Use ctrl+b, very recomended to search each line, if not, then add it).

WaypointSpan: the distance in meters of each Waypoint, modify it with caution or sometimes its better not add it. Generaly i see it's around 4 to 6, some Friends say they feel tracks with 4.x or lower great as AI corner really smooth.

DrivingLines: when you use 2 or higher, change the behaviur of the AI, acting more aggresive and doing bumpdrafting, also making them race in 2 lanes (or the ones you specefied). **Its recomended only use 2 or higher in ovals in tracks for StockCar Racing only, because if you used it with CARTS or IndyCars; gona end really badly.**

FuelUse: This one calculate the Fuel usage that track used, this its very important as its this tell you how much laps you have of fuel and for the AI. People tested this very much and there are multiple formulas you can use using track lenght

track lenght*16 or trackLenght*15.5 (someones do tracklenght*21 but not tested)

AICautiousness= (0.9900): (appeared back in F1 2001 and GTR2 HQ Mod bring this in his mod): This controls how cautious the AI is on track. Lower values make them race more aggressively and fight for positions more, while higher values make them more cautious, especially noticeable in large packs of cars. The recommended range is 0.30 to 0.80. I suggest starting with a value of 0.50 — if you see the AI making too much contact, increase it; if not, you can try lowering it.

AIBrakingStiffness=(1.0000,1.00000,0.9000): This tell how aggressive AI should brake, each number for each type of car (Rear Wheel drive,4WD,Front Drive), 1.00000 tells to brake into the last moment, you can lower a little this number for like (0.8500,0.8500,0.8000) and this gona tell AI to brake a Little more earlier, lower more this number and AI gona brake much earlier, Its very important and you can use it along with brake power/grip to find the perfect AI braking balance (better if we try to modify this only)

Slowwhenpushed: this control when the AI is touched/bumped, to brake the number of the force, its very recomended to ALWAYS add this line and should ALWAYS set in this range of 0.05 to 0.35, but you dont want to be very low because can end with chaos of AI crashing and not slowing down, also some tracks like ovals its fine to put 0.00.

LaneSpacing: this define the distance betweben the left and the right patch, noticiable when formation lap its enabled. Also some say this define the widht of the AI fast line but im not sure about this.

InsideAdjustment
OutsideAdjustment

Those lines are a fine turne of how wide/shallow they take inside/outside lines. I recomend stick around this
InsideAdjustment=-1.5000

OutsideAdjustment=-0.5000

Its not know what the diference if you use positive numbers, however outside need to have always lower (in negative) number than inside as the example for indianapolis the best value its -5.00000 for inside and -0.50000 for outside.(here my experiment using Indy Cars 2003 in Indianapolis)

```
First test
Outside Adjustment = -5.500000
Inside Adjustment = -10.000000

It seems the AI tends to use the track more.
The AI tends to make 3-way and even 4-way shifts on the straight. This
has led to more accidents due to the aggressiveness of the shifts, and
in fact, the RCDs are very aggressive, making it very easy for them to
end in chaos.

Second test
Outside Adjustment = -10.500000
Inside Adjustment = -5.500000
Basically, the values are reversed.

The AI seems to follow the outside line more and stay in line. Only in
classic corners do they head for the inside line to try to overtake.
Only some 2-way and 3-way shifts disappear.

Now, if both are set to 0, the AI will have 2-way, but 3-way will be
almost nonexistent.

And finally, the default value that comes with this mod: Track
Outer Adjustment = -0.500000
```

```
Inner Adjustment = -5.000000
```

It has really good values. Aside from trying to overtake using the track more, the AI allows overtaking in two lanes and sometimes in three lanes without being too chaotic.

```
WorstAdjust=(0.8000)
MidAdjust=(1.0000)
BestAdjust=(1.1500)
```

Those lines define the speed of the AI depending of your strenght setting. Worst correspond to 80% of strenght, mid to 100% strenght and best for 120% strenght
here its how work. When you select a dificult <=80, you are using the worst adjust, if you use a value <=100 you using the mid one and if you use a strenght>100 you are using the best adjust
took an example, if you select 95% strenght and you have in mid adjust 1.000 then you are doing 95*1.000*speed of the Waypoints and this gona define the actual speed of AI. Its very important to first adjust everything in 1.0000 and select 100% strenght, then you calibrate the mid adjust for being around 0.3 second of the best lap time of the series in that track and then you calibrate the best and worst adjust according to your preference. And important to being balanced so user dont gona need to adjust the strenght for each track.(thanks jgf for all this information and recomendation)

AISpec: I think this line dont work in GTR2/Rfactor, but its seem work in AMS. Basicaly this act when you set strenght<100% AI gona be limited with this values. The first one Limit the aceleration, the second one the speed, the third the cornering and the four they deceleration. (need further testing)

also according to **SpeedNut** in race07 do this (i never played Race 07, so i can`t test this sorry):
Refers to a set of parameters that define the behavior and performance of the game's artificial intelligence (AI) opponents.
The parameters are as follows.

- Grip level: This parameter defines the level of grip or traction available to the AI opponents. Higher grip levels will result in faster speeds and more aggressive driving behavior from the AI opponents.
- Path width: This parameter defines the width of the racing line or path that the AI opponents will follow. Narrower paths will result in more cautious driving behavior from the AI opponents, while wider paths will encourage more aggressive overtaking maneuvers.
- Speed: This parameter defines the target speed that the AI opponents will aim to achieve. Higher speeds will result in faster lap times and more aggressive driving behavior from the AI opponents.
- Braking behavior: This parameter defines the aggressiveness of the AI opponents braking when braking into corners.

AIDraftStickiness: This one its a pain because there not actual best value, this dont gona fix the AI dont overtake in straight problem but can mitigate it, and im gona talk with my preference because some people preffer the value 3 to 5, this line define if the AI gona want to change the car that he drafting or he should maintain with that car (or pull a overtake)

I prefer using a range of 0.00 to 2.50, sticking around 0.90 to 1.00, sometimes i feel better racing deleting this line (lol), but some people say that they have better racing with a value betweben 3 to 5. So its all to you to test this, i would take into consideration that AI maintaing the draft until about close to the mid of the straight and also that AI dont wana stay on the draft of a lapped car very much and start to overtake him

also its seem in AMS this was changed and most of track use a value of 3, so take that also in consideration.

The way its work its add this amount to the speed of the car infront (it's in M/S) and when the car that it's following it surpas this value, then try to overtake it, or to change from draft partner.

AIRange: this line define the diference betweben the AI drivers, remenber this interact whith the speed/raceability variable from rcd, if you decide to adjust each rcd then i recomend put this value in 1.0 or, a good practice i see from race07 smart AI its this: "0.5 for Nordschleife, 0.8 if ref time is superior to 2min, 0.9 if between 2min and 1min30".

RaceQualRatio: appears in both Rfactor and Automobilista game. However seems only the last one work. Lower values make that AI qualy its more slower than race, higher make the oposite more faster AI in qualy than in race recomended its to put in 1.005 for example so AI its slightly more faster in qualy compared to race (remenber you can do exclusive qualy setups for AI when we reach to Setup Section). Also something to note in F1C this variable also appears but not work as same, this slowdown the race compared to Qualy. So a lower value make AI in race more slower thant qualy and viceversa. Should be realy realy close to 1.0 like 0.990 for example in that game.

SpeedOffset: this seems to act as a offset for the AI Cornering speed, its unknow its original purpose as its seems do a lot of iterative process (thanks Niels for that information), so i think its better leave this to 0.0000 or deleted this line, unless you know if this can improve AI in a way with this.

CheatDelta: dont seem to work in Rfactor and GTR2, Unknow if this work in automobilista.

Now according to SpeedNut in Race 07 do this (as say i dont played Race 07 so i can`t test this):

Refers to a parameter that controls the amount of time penalty that the AI-controlled cars receive if they cut a corner or go off-track during a race.

The CheatDelta parameter specifies the amount of time penalty that the AI-controlled cars will receive if they cut a corner or go off-track. The value is specified as a number of seconds.

The three parameters used in a CheatDelta formula would be:

1. DeltaStart: This parameter specifies the start value of the time penalty for cheating or breaking the rules.
2. DeltaFactor: This parameter specifies the factor by which the time penalty should be multiplied, based on the severity of the infraction.
3. DeltaMax: This parameter specifies the maximum amount of time penalty that can be imposed on the player or AI-controlled car, regardless of the severity of the infraction.

SB EXCLUSIVE: im gona copy paste what ISIMOTOR 2 COMMUNITY AI Documentation because explain this perfectly.

RaceRatio

RaceRatio is a percentage determining how much faster AI are during the race – compared to their baseline speed (e.g. in practice) as determined by WorstAdjust, MidAdjust, and BestAdjust. So, for example, RaceRatio=1.04 means that AI will be 4% faster in the race than in practice.

QualRatio

QualRatio is a percentage determining how much faster AI are during qualifying – compared to their baseline speed (e.g. in practice) as determined by WorstAdjust, MidAdjust, and BestAdjust. So, for example, QualRatio=1.04 means that AI will be 4% faster in qualifying than in practice.

GarageDepth=(3.0000)

This its the amount of meter that AI gona run when he leave its garage spot, increase this if AI steer to early or decrease it if AI crash into the pit wall due to not steering in time.

jfg recommendations. Again thanks to him for this

“Ever have a weekend where you ran circles around the AI in practice but they left you in the dust in the race? Easily fixed.

First, I always set the in-game AI slider to 100% before any AI tweaking so all adjustments are made to a common baseline.
Now start a practice session, accelerate time if you wish, and note the AI times after a few laps; tweak "MidAdjust" to get these where you want them. Start a qualifying session, allow it to run out (accelerate it if you wish but do not end it early), adjust "QualRatio" to get these times where you want them. Now start a race, accelerate if you like, let them run enough laps to get stable times and, you guessed it, adjust "RaceRatio" to put those times where you want them. AI times will now be consistent through the weekend and when you adjust the AI slider in-game, all session times will track accordingly.

At this point RaceRatio and QualRatio may be adjusted without affecting anything else, but if you change MidAdjust it will affect both Ratios, so you may have to tweak them again.”

And this cover all AIW aspect, now we can go to:

Game Data Base (GDB File)

GDB file are used for define the property and information of the track and the rules of the series you runing (SB and F1C this last one)

Im gona cover all related to AI plus some extra variables that seems are not very know (and work in almost all games)

Attrition = 30: this define the wear in component of cars gona suffer when race in the track, its not very clear how does work exactly and i dont know if this affect the human player, interact with the LifeTimeAvg from the engine file so depend of how you set that file (the general rule its to multiply the duration of the race with 1.5 and LifeTimeVar generaly its LifeTimeAvg/3 or /4)

Here a good formula from F1C that really work well if you have the lifetime values well setted:

(100/amount of cars)*amount of cars you want to failure

Here a example, in 1999 in Australian Gran Prix, 22 cars participate in the race, 10 cars DNF due to mechanical failure (the rest were due to accidents) so we do: (100/22)*10=45
so when you race, aproximated 10 cars gona have a mechanical failure (not always, can vary, can be more cars or less cars but this give a good number for that GP specificaly)

Also i not know if this value work with the mechanical failue in normal, only i tested it with scaled mechanical failure.

PitWindow = 42 This line appears in Nascar Thunder games, tells the AI to pit each specified amount of laps (this case in 42), i tested it in Rfactor and AMS and seems to not work, unknow if this work in GTR2 (not necesary the thruth, dont add this line)

GarageDepth = 2.00 do the same as in the SB AIW files: specify the distance (in meters) that the car should travel from its garage spot before turning into the pit lane. Increase this value if the car starts steering too early, and decrease it if the car crashes into the pit lane wall.

AIDryGrip = 1.01
AIWetGrip = 1.05

Those control how much AI grip gona have in the track, higher values make them have more grip and race the corner more harder, only adjust it if you notice AI are slow (or fast) in turns (balance it with the adjusts from AIW), also you can use the AIWetGrip to nerf the AI in wet conditions (very useful in F1C if you dont want to adjust the wet lines from RCD)

PlayerFuelMult= 1.00
AIFuelMult= 1.00

Those make player/AI to use more fuel (or less), useful if you want to adjust it for a specific track

PlayerTireWear = 1.00
AITireWear = 1.00

Those lines as the name say control the wear of tires of the player and AI in this track, useful if you want to balance the wear for this specific track.

TireHeatMult= 1.0
RearTireHeatMult= 1.0
FrontTireHeatMult= 1.0

Control how much your tires heat, i dont recomend use this unless you have an unrealistic mod with unrealistic tires heat.
(Something to note, those was tested in Rfactor and Gtr2, don’t work In F1C)

I see some mods like Race 07 Smart AI mod or GTR2 HQ mod add this line:

```
Drafting
{
  BaseDropoff=0.199 // higher number -> more drafting effect
  LeadingExponent=1.8 // higher number -> lower effect on leader
  FollowingExponent=1.9 // higher number -> lower effect on
followers
}
```

With this you can control the draft effect, you can add to each track or to the game.gdb for default value, remenber Rfactor and Automobilista added more options and can appear in the .hdv of the car.

PitStopStrategies

This line defines the amount of pitstop a specific AI driver gona do, and in what lap. Here a example from a overtake fórum

```
PitStopStrategies
{
Driver1 = 2 - 20,30
Driver2 = 2 - 25,50
Driver3 = 3 - 15,30,45
}
```

Driver names are the names that appears in the .veh (or .car) files, the number after = its the amount of pit stops he gona do, and the number after – its the lap the driver gona stop

Something to take into consideration, this gona make AI fill its tank for the amount of laps for each pit stop, so if you have Michael Schumacher= 2 – 5,30. He gona start the race with fuel for 5 laps, and then gona refuel so he can reach to 30 laps, after that gona fill the tank for the amount of laps before the race ends. I not tested this with time based race so this maybe dont gona work well in those scenarios

Also for mods that dont allow refuel, should have this set in 0 and depend only on tire wear. Like CMT mods

here an example

```
PitStopStrategies
{
Driver1 = 0
Driver2 = 0
Driver3 = 0
}
```

So this ensure AI race with the tank full (or for the amount of laps the race gona be) and only stop for tires.

```
  SettingsFolder = Track
  SettingsCopy = Grip.svm     // first "SettingsCopy" becomes
your default!
  SettingsCopy = track.svm
  SettingsAI = AItrack.svm
```

This tell the game what folder create for save your setups, also what setups to load for each specific track, the most important one is the SettingsAI as this tell what setup AI gona load for this track and we gona cover that in a specific section

```
  Qualify Laptime = 95.246
  Race Laptime = 100.926
```
Those lines define the Lap time AI gona do when you skip the session.

And those was the most important things you can adjust GDB, there more options but they are self explanatory. We can move finaly to other things but first we gona stop with a variable that appear in SimBin games

COMPARATIVE TIME (SB ONLY)

In each .Car File, there a line at the end called `ComparativeTime`

In sumary. It's a percetage that multiply the Qualify and Race Laptime when you skip the session, adjust it for each car group

all top team should have a value like 90, Mid group like 92 and lower group 94, also if this its for multiclass Racing, use more higher number like GT class 100 and NGT in 104.

SETUP FILES FOR AI (SVM)

One great advantage we have is the ability to assign specific setups for each AI. This is very important, as it allows us to control how they brake, their aero, tire compound, gear ratios, and other parameters.

You can assign a global setup by creating a setup file named ai.svm and placing it in the same folder as the .hdv/.hdc files or in the same .veh/car folder if you want a specific driver or team to use that setup.

It's also possible to assign track-specific setups. To do this, check the SettingsAI line in the track's .gdb file and name your setup file exactly as specified there. Then place it in the same folder as the .hdv/.hdc or .veh/car files, depending on whether the setup is meant for a specific car or team.

Something to note is that you can make a setup exclusive to qualifying or race sessions by adding .qual or .race before the .svm extension. For example:
Brianza_Italian_RFE.qual.svm

So in summary, the format is:
trackname.qual.svm or **trackname.race.svm**

Also, when creating a setup, the svm file gona save also the upgrade you selected in that moment (this applies to rFactor and AMS, where the upgrade system is available). This allows you to fix issues like AI running with a full tank during qualifying in rFactor by creating an upgrade that limits fuel capacity.

When making a set up for AI, the most important parameters to adjust are **aerodynamics** and **gear ratios**. Give the AI a slightly longer gear ratio than yours to prevent them from hitting the RPM limit constantly, which can damage their engines.

Also, remember you can now adjust **brake bias**. AI drivers generally prefer more forward brake bias. After that, feel free to test and tweak other parameters.

For qualifying setups, you might want to reduce the **radiator size** and **brake duct size**, and use a higher **turbo boost** setting. (Just remember to revert these changes for the race setup.)

Finally, you can also assign the **tire compound** the AI will use. However, keep in mind that the AI will first use the tire compound defined in the .hdv/.hdc files. After making a pit stop, they will switch to the compound specified in the .svm setup file.

CALIBRATING THE AI (AMS/RF ONLY)

An exclusive feature of rFactor/AMS is that you can train the AI on the racing line, allowing them to improve their lap times and their pace. I recommend creating a new player file with both fuel, wear, and failure turned off, and with a private session enabled.

Now, go to your PLR file and find the **Autocalibrate AI Mode** setting and set it to 1. You might also want to set the **AI Autoline Smooth** value to 1 or 3 to make their racing line smoother. Additionally, set **AI Mistakes** and **AI Limiter** to 0. After that, save the file.

Next, start a testing session and enter the track you want to train the AI on. Add an AI driver, and they will display a message like "Stay apart, human." This is when they start learning. After the AI completes 1 or 2 laps, press **Ctrl + T** to accelerate time. After a while, the AI will either return to the pits or indicate that they are finished with their training. They will then show you a meter indicating how much further they think they can improve their performance. The ideal result should be 0.0x meters, but depending on the AIW file of the track and the car's behavior, this might not always be possible.

Now, remove that AI and add another one. The new AI will reuse what the previous driver learned and continue from there. Once the second AI finishes, you can exit the game.

Go to the .veh folder of the car that completed the training, and you'll find a trackname.ini file. Copy this file and paste it into the car's .hdv folder. Now, every driver will use this .ini file and perform better on that track.

If, for any reason, you want to race with the same player profile you used to train the AI, return to the .plr file and disable **Autocalibrate AI**, as this setting makes the game not read the .rcd files.

General Considerations

You want to set the AI strength to your liking the one that makes you fight for positions and finish where you intend to. The **Aggression** setting depends on the series you're racing in and how you've configured the .rcd files. Start with higher aggression (like 100% or 90%), and then lower it by about **-20%** for each test if you feel the AI is overly aggressive. Remember to reduce it further for tracks with a poor AIW or tracks that tend to end in crashes, like Monaco.

Now, when you're in the game, don't skip qualifying. Simply press **Ctrl + T** and let the AI complete their lap times. This ensures they finish learning the track and have an appropriate starting position. I also prefer to **Ctrl + T** through the warm-up session, just to be safe. After that, during the race, I recommend not racing too aggressively with them in the start, as they tend to be in groups and "pretend to have cold tires." Once you've passed that phase, race them like you would against human opponents, and you'll notice they race well and provide a good challenge.

Conclusions

And that's everything covered in this guide. We've gone through every aspect that can be adjusted. If you made it this far, thank you so much for taking the time to read it I truly hope it helps you.

Unfortunately, some things are impossible to achieve, like making the AI always overtake on straights or getting them to use more than two tire compounds. But with what we currently know, this is everything we can do. From here on, it's all about fine-tuning each variable to make sure everything works properly.

As I mentioned at the start of this guide, thanks to everyone who helped me, to the community that has done so much deeper research, to my friends that support me, and to the developers who created these great games that we’re still enjoying today.

That’s all for now—hope you have a great race.
Goodbye! ^-^