



TRADERPLUS DOCUMENTATION

By TheDmitri



Read me !!!

Hey! It's Dmitri! Nice to meet you future TraderPlus's user, I'm glad that you opened this document. I know your feeling, you may think about the number of pages (46) of this documentation and you're already feeling afraid, depressed, annoyed, or even angry about it.

"Dmitri, you suck! I don't have time to lose! I just want to install this mod in a couple of minutes and just play with it, how difficult is it to do ???"

Let me reassure you dear user, if you read a couple more lines of what I'm saying, I'll explain to you the reason and how to be SUPER-EFFECTIVE !!!

This documentation is big because I'm trying to answer all your questions and since this mod allows you to set up so many different features, it is huge! It's mainly because this mod can really adapt to your server's need **but don't panic**, basic default install doesn't take that long and looking at what you seek neither.

In order to find information in this documentation, I strongly recommend checking the Table Section page 3 and click on the title that seems to be what you're looking for.

For example, for installing the mod:

TABLE

I.	Everything you need to know	4
2.	How to resolve my issue?	4
3.	How to properly share an issue in the discord/to me.....	0
4.	How to report bugs in Trello	0
II.	Installation of TraderPlus.....	2
III.	How does the Bank system work?	3
1.	How to add a bank trader	4
2.	TraderPlusBankingConfig.json	5
3.	Bank account storage location and variables.....	5
IV.	How does the Receipt system work?	6
V.	How does the Garage system work?	8
VI.	How does the CarLock system work?	11
VII.	How to convert your TraderConfig.txt to TraderPlusPriceConfig	12
VIII.	Files's explanations	14
1.	TraderPlusGeneralConfig	14
2.	TraderPlusPriceConfig.....	19
3.	TraderPlusIdsConfig.....	24
4.	TraderPlusVehiclesConfig	27
5.	TraderPlusSafeZoneConfig.....	29
6.	TraderPlusStock_ID_X	31
7.	TraderPlusLogs	32
IX.	How to make static object works with TraderPlus.....	33
X.	Explanations regarding the UI Interface	34
XI.	Useful tools to have	37
XII.	Conclusion.....	37

And it directs you directly to the section. Well, that's the biggest trick you need to know! Apply it to each question you have about the mod! Now good luck!

TABLE

I.	Everything you need to know.....	5
2.	How to resolve my issue?.....	5
3.	How to properly share an issue in the discord/to me.....	0
4.	How to report bugs in Trello	0
II.	Installation of TraderPlus	2
III.	How does the Bank system work?	3
1.	How to add a bank trader.....	4
2.	TraderPlusBankingConfig.json.....	5
3.	Bank account storage location and variables.....	5
IV.	How does the Receipt system work?	6
V.	How does the Garage system work?	8
VI.	How does the CarLock system work?.....	11
VII.	How to convert your TraderConfig.txt to TraderPlusPriceConfig	12
VIII.	Files's explanations.....	14
1.	TraderPlusGeneralConfig	14
2.	TraderPlusPriceConfig	19
3.	TraderPlusIDsConfig	25
4.	TraderPlusVehiclesConfig.....	28
5.	TraderPlusSafeZoneConfig	29
6.	TraderPlusStock_ID_X	32
7.	TraderPlusLogs	33
IX.	How to make static object works with TraderPlus.....	34
X.	Explanations regarding the UI Interface.....	35
XI.	Useful tools to have.....	38
XII.	Conclusion	38

INTRODUCTION

That mod was created as a challenge to myself. A trader mod is a lot of work and a lot to think in order to fill every gap possible. It took me 6 weeks to complete it to release the first released. Now, 16 weeks has passed and a lot of new features has been added and many more are to come.

The main idea is simple: **“You can only buy what has been sold previously to the trader”**

I wanted to make it more deep than the current Trader available, made by Dr.Jones. I made that trader to have dynamic price based on stock among other cool features. I hope you'll enjoy it as much as I enjoyed making it. More will come over time, so it will be even better.

I want to thank:

- Dr.Jones for being the first in the community releasing a trader. He's the one who show us how a trading system is important for the community and he deserve our gratitude for that.
- Chubby for providing the currencies to the mod
- Morty for the nice feedback during testing
- All the tester that came testing the mod
- All the modding community that allow me to learn and be able to make that

I. Everything you need to know

This documentation is normally very complete. I can understand that my English can be a bit weird or hard to understand as a non-native speaker, but I try my best so all of you can understand it. In that documentation you should find everything you need about your questions so **PLEASE, read it!**

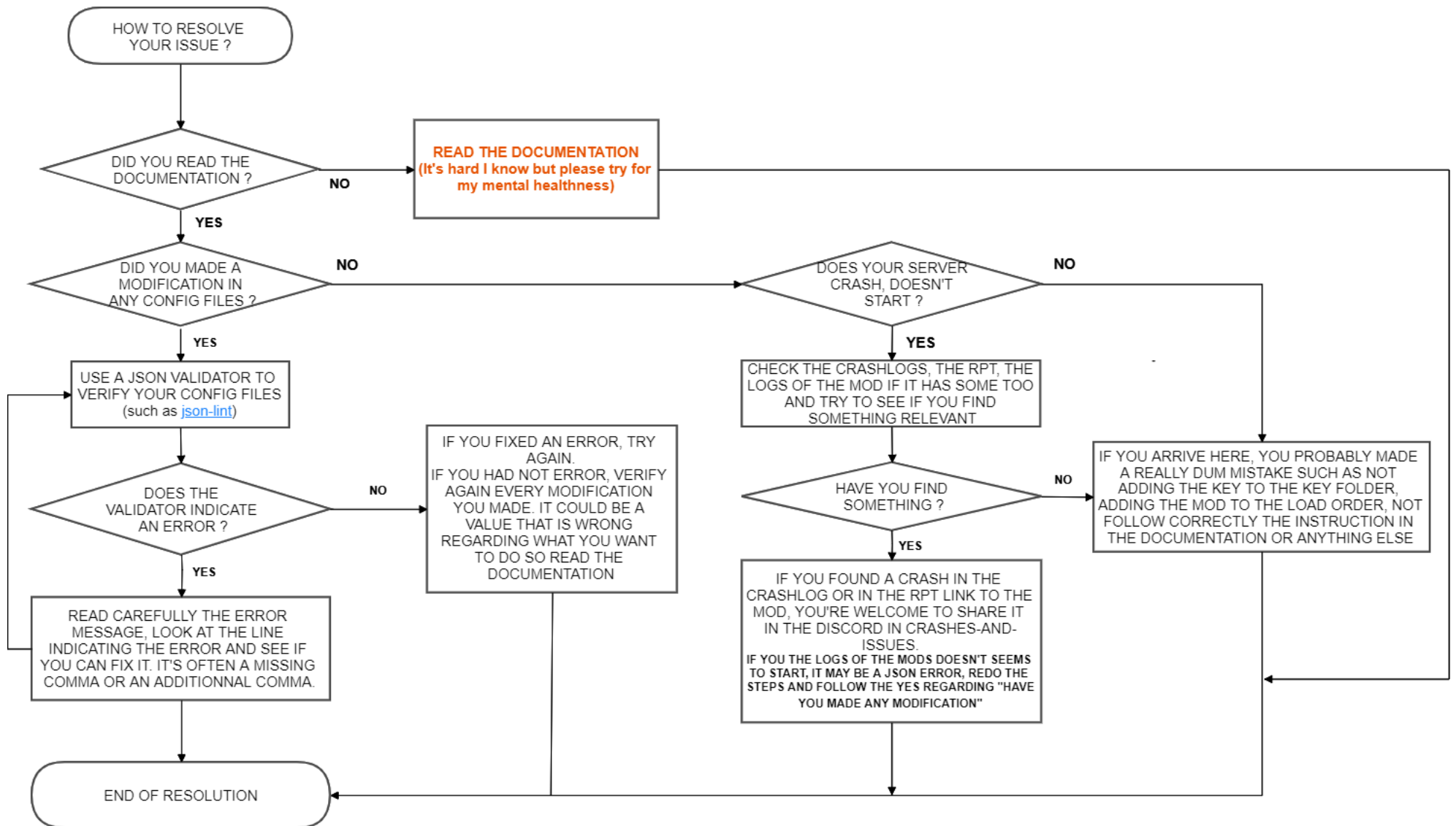
If you come to the discord chat asking a question already answered here, either I will not answer, or I will tell you RTM or RTFM. (Read the Manual or Read the Fucking Manual, depending on if you're the 10th person asking me today something like this 😊). That's not meant to be rude or anything but keep in mind that my time, as yours is precious.

1. How to get the information I need?

So always refer to the TABLE or use CTRL+F to look for key words and please read entirely until you find what you need. Once you are sure it is not in the documentation, you're welcome to ask your question in the discord.

2. How to resolve my issue?

While modifying, setting up the mod, you may do errors that conduct to an issue such as server not starting, mod not working correctly anymore, or more. In such scenario, you can use that diagram to know how to proceed. Once again, it's there to give you basics step on how to resolve issue on your own.



3. How to properly share an issue in the discord/to me

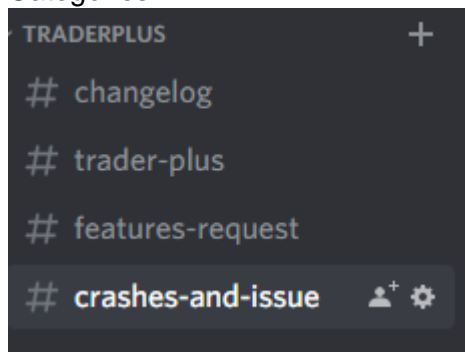
In order to report an issue related to the mod you must always try to follow the following template:

Title: In a short but complete sentence, try to explain the issue so I or anybody can understand what it is about in order to look into it.

Description: Explain exactly how you met that issue and try to give the different steps that conduct to that issue. Don't forget to mention every attempt you made to fix the issue on your own. That way, nobody will tell you to do it again.

Data: If you can, you're really invited to add screens, videos that can help describe the issue.

You can post that in the discord under the section crashes-and-issues in TraderPlus Categories

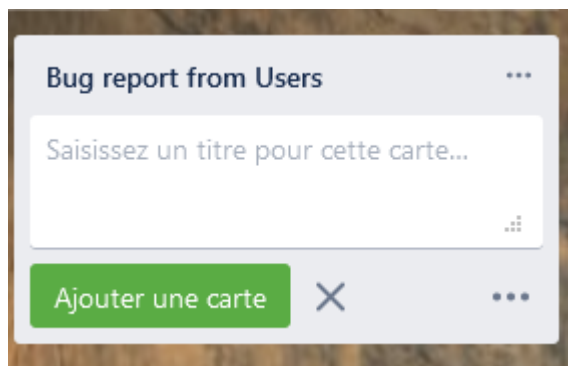


In addition, so I can get a trace and see if faster, you can create a card under Bug report from Users in the TraderPlus Trello: <https://trello.com/b/SOFdxrZ2/traderplus-bug-tracking> (it's not optional, please do it there too thanks in advance)

4. How to report bugs in Trello

Step 1: Click on the link above, register, connect to trello.

Step 2: Click on add a card in "Bug report from Users" like showed below and put the Title of your bug/issue then click on the button add a card.



Step 3: Click on the new card that you made and add in description the description of the issue.

Not show sellable item than can't be sell when sell = -1 ×

Dans la liste [Accepted](#)

ÉTIQUETTES

Mineur +

Description

Ajouter une description plus détaillée...

Activité Masquer les détails

DM Écrivez un commentaire...

DM **Dmitri Medeleiv** a déplacé cette carte de features request à Accepted
il y a 3 heures

DM **Dmitri Medeleiv** a ajouté cette carte à features request
Hier à 11:15

SUGGÉRÉES ⚙

Rejoindre

AJOUTER À LA CARTE

Membres

Étiquettes

Checklist

Date limite

Pièce jointe

Image de couvert...

POWER-UPS

Marker

+ Ajouter des Power...

Passez à Business Class pour obtenir un nombre illimité de Power-ups par tableau.

[Promouvoir l'espace de travail](#)

BUTLER ?

+ Ajouter un bouton

Step 4: Finally you can add screenshot or video with pièce-jointes (attached file) and close it.

All right, now you should have basic knowledge on how to help yourself and me



II. Installation of TraderPlus

This part is dedicated to the installation of TraderPlus.

Normally, you should already have subscribed to the mod in the Steam Workshop and opened the mod folder to get access to that documentation file. So what do we do next :

Step 1: We copy the mod folder @TraderPlus to your DayZ Server Root.

Step 2: We go inside @TraderPlus\key and we copy it to paste it to your keys folder in your DayZ Server Root.

Step 3: We go inside @TraderPlus\ServerProfile and we copy the TraderPlus folder to paste it in your profile folder located normally in your DayZ Server Root. (that's the folder where you can find RPT files, Logs files and Crashlog among other mod config files such as zomberry, vpp, expansion etc...)

Step 4: We have finished the basic step to make the mod work. The mod is ready to be launched for testing in Chernarus.

Step 5: Make sure that you have inside your TraderPlus folder the following folder:

Nom	Modifié le	Type	Tail
TraderPlusBankDatabase	28/04/2021 15:25	Dossier de fichiers	
TraderPlusConfig	28/04/2021 17:18	Dossier de fichiers	
TraderPlusDatabase	16/04/2021 17:48	Dossier de fichiers	
TraderPlusLogs	28/04/2021 17:18	Dossier de fichiers	

If you don't have some, you need to create them manually.

"All right Dmitri, but I don't run Chernarus, how can I make the mod work for my custom map ?"

So, dear user, if you're in that case, you'll have to edit my config file in order to set your traders' locations. I recommend to use any modding tools such as [DayZ Editor](#) or [Custom Offline mod](#) to create your traders' locations with objects.

III. How does the Bank system work?

Yes, indeed, there is a banking system in TraderPlus that allow players to store their money into a database. For now, the banking system works only with basics features such as:

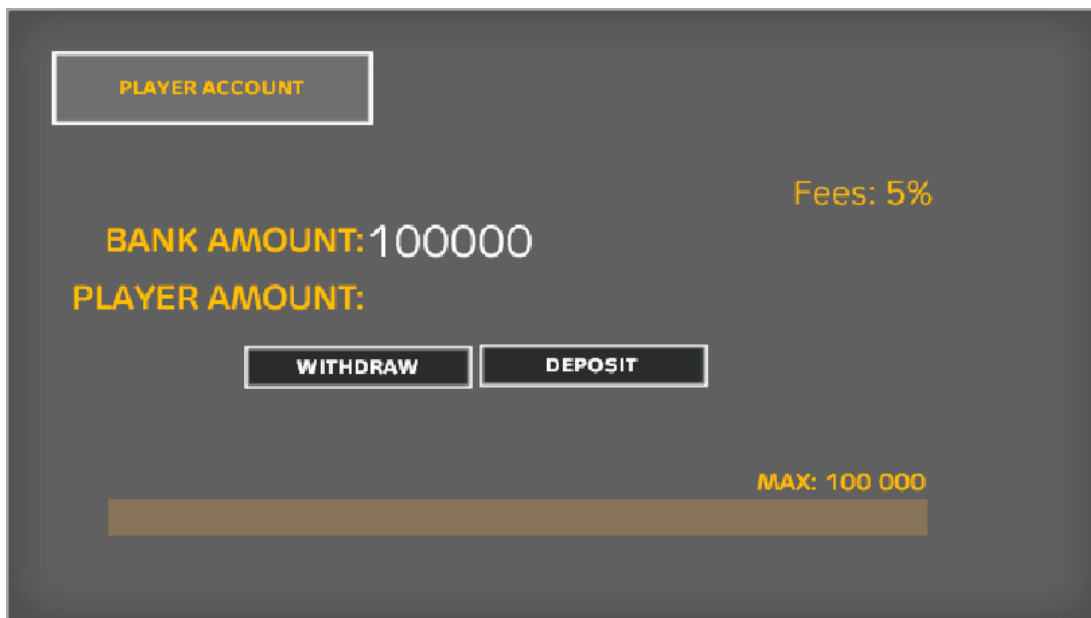
- Withdraw money from bank account
- Deposit money to bank account

Later, the following features will be included:

- Ability to create a clan account
(defined price to create one, permission settings for leader member to invite, kick, etc...)
- Ability to transfer money from one player's bank account to another

"Yes but Dmitri, there is already some bank mod in the workshop, what's the point of using this one ?"

Well, dear user, having a bank system inside the trader allows me to fully control it and makes it more adaptable for additional features if needed. Moreover, it reduces the amount of mod you need to load into your server and reduce the chance of incompatibility between mods.



This is a picture of the default interface, as you can see, you have:

BANK AMOUNT: Amount of money owned by the player in the bank

PLAYER AMOUNT: Amount of money owned by the player in his inventory.

TRANSACTION AMOUNT: You can edit it to enter the amount you want to deposit/withdraw.

A PROGRESS BAR: It will be filled based on the max capacity of your bank account.

FEES: Value in %, that will be deduce from deposit and lost. Call that fees from the bank.

Again, everything is configurable.

If you don't want the bank system, you can install in your server **@DisableBankTraderPlus**.

Make sure to load that addon before TraderPlus in your load order.

if you want to change the fees, that's possible in the TraderPlusBankingConfig.json

if you want to change the max money amount that can be stored in the bank, that's also in the config.

If you want to translate the UI, that's also possible.

1. How to add a bank trader

This part is gonna be easy if you already know how to add trader into the game.

If you don't know how to add a trader into the game, I invite you to check the VI.2 GeneralConfig.json. Then come back when you're familiar with that part.

All right, so, to add a trader bank into the game, you're gonna choose your object. It can be a static object such as a ATM or a real NPC (SurvivorM_Peter). Whatever you want to put, you can. (Keep in mind that static object must inherit BuildingBase to work, there is a tutorial in **VI.2 Traders**)

Then just do as you would do to add a Trader into the config.

There is only one change to make, that's the ID. **In order to make your trader a bank trader, you must set Id = - 2. (THAT'S A MINUS 2)**

```
{
  "Id": -2,
  "Name": "TraderPlus_BANK_ATM",
  "GivenName": "ChaChing",
  "Role": "ATM",
  "Position": [
    8207.0,
    471.0,
    9061.0
  ],
  "Orientation": [
    -117.0,
    0.0,
    0.0
  ],
  "Clothes": []
}
```

2. TraderPlusBankingConfig.json

So let's explain how this config file works:

"TransactionFees": That's a float variable dealing with the fees taken by the bank each time the player is doing a transaction. By default, it's set at 0.05.

A friendly reminder, the value to set here must be set between 0.0 and 1.0 because it will be used as a %.

So 0.05 = 5%, 0.0 = 0% and 1.0 = 100%.

"DefaultStartCurrency": That's an int variable dealing with the default amount a player when his account is created. By default, the amount is set to 0.

"DefaultMaxCurrency": That's an int variable dealing with the maximal amount a player can store inside his bank account. Keep in mind you can always change the max value for a specific player by going directly into their bank file. By default, the amount is set to 1000000.

"TextUI": That's a string variable dealing with the translation of the UI, Just replace the sentences if you need to.

"CurrenciesAccepted": This string array allow you to restrict the ATM to only accept some defined currencies. Keeping this variable empty will make the atm accept all the currencies define in **TraderPlusGeneralConfig.json**

"Hey Dmitri, How can I make the ATM only accept EUR money?"

Well, dear user, check the picture below:

```
"CurrenciesAccepted": [  
  "TraderPlus_Money_Euro100",  
  "TraderPlus_Money_Euro50",  
  "TraderPlus_Money_Euro20",  
  "TraderPlus_Money_Euro10",  
  "TraderPlus_Money_Euro5",  
  "TraderPlus_Money_Euro1_Coin"  
]
```

Keep in mind that each ATM will accept only the currencies defined above. If you are thinking about having the atm able to store all kind of currencies separately, that is not possible. At least for now. The work around I can see would be to restrict some trader with licenses so you can exchange some secondary currencies with the one accepted by the atm.

3. Bank account storage location and variables

As other bank system, you're able to get access to each player's bank account file. Thoses files are saved inside TraderPlus\TraderPlusBankDatabase\.

Nom	Modifié le	Type	Tail
TraderPlusBankDatabase	28/04/2021 15:25	Dossier de fichiers	
TraderPlusConfig	28/04/2021 17:18	Dossier de fichiers	
TraderPlusDatabase	16/04/2021 17:48	Dossier de fichiers	
TraderPlusLogs	28/04/2021 17:18	Dossier de fichiers	

Each file start by Account_ followed by the SteamID64 of the player.

Nom	Modifié le	Type
Account_76561198047475641.json	28/04/2021 17:22	Fichier JSON

So if you want to access someone account file, you'll need to know their SteamID64 or make a file search with their Steam Name.

Then inside, you'll find thoses informations:

```
{
  "Name": "TheDmitri",
  "MoneyAmount": 907898,
  "MaxAmount": 1000000,
  "Licences": [
    "Car Licence",
    "Weapon Licence",
    "Admin Licence"
  ]
}
```

"Name": that's the player's nickname on steam.

"MoneyAmount": that's the amount of money stored.

"MaxAmount": that's the maximal money that this player can store.

"Licences": that's a string array that contains all the licences own by the player.

You can add licence directly in that array if you don't want to let people buy it as a product. For example, you want to define a trader only for admins, you'll go inside each admins bank account and add the admin licence in it.

IV. How does the Receipt system work?

So as you know, trader sell items but also vehicles. Dealing with trading vehicle can be resolve from many points of view such as having a parking spot in order to park the car to trade it. I decided to make something a bit different. In order to sell or buy car, you'll need a

receipt that will act as a proof of ownership.

In order to make the receipt system work with your car, make sure your car is set in the VehicleConfig with its attachments. If you don't, the car will be popping on your head and you'll come to me saying my shit doesn't work 😊

To get a receipt from a car, you'll have to "pack it" with a reinforced wrench (classname: TraderPlusWrench). **Keep in mind you can't pack a car that have item in its cargo.**

So once packed, you'll be able to go to the car seller trader and sell it as a car.

There are multiple conditions regarding packing a car into a receipt:

=>There is a variable called IsReceiptTraderOnly in the GeneralConfig. If set to 1, you'll be able to pack and deploy your car only inside a safezone. If you set it to 0, that can be done from the entire map.

In addition to that:

- If you use the CarLock system, only the owner or the person who has the password can pack it. It will take between 15 and 30s to pack. *(Check V. for how the carlock system work).*
- if you don't use CarLock because you wanted to keep using your key system from MuchCarKey for example, then anyone can pack a car that is UNLOCK. (it means locked car can't be packed). It will take 30 min to pack.

Like mentioned above, you're also able to deploy it to get your car. There are script securities to make sure that we can't deploy a car on a player or on a roof.

You may think about the car's attachments, how does it works ?

Well, it's good and bad.

You have to make sure your car classname is well written in the VehicleConfig in order to be able to use the receipt system. Also, all attachments in that file will spawn everytime, no matter what. So it means that packing a car that miss attachments or fuel will give a full car once deployed again. That means players can actually try to glitch the system by sellings car parts. That's why I recommend not to allow players to sell car parts.

V. How does the Garage system work?

-The garage system is pretty like what you may have seen already in the modding community. It allows users to store vehicle into an external database to “protect” the vehicle from the server environment such as raiding, server database wipe, days bugs and mainly to improve server performances. An then, when the user wants it, to get back the car on a parking location near the garage npc.

This garage system **is for now compatible with TraderPlus CarLock, Trader keys and MuchCarKey.**

Now that we know how the garage system works, let's talk of how to set it up.

In your TraderPlusConfig folder, you should find a new config folder called:
-TraderPlusGarageConfig.json

Here is the file's explanations:

"VehicleMustHaveLock": That Boolean allow you to define if you want to be able to store vehicle without the need of locking the vehicle by any mods compatible with this system (TP, Trader, MCK)

"SaveVehicleCargo": That Boolean allow you to define if you want to be able to store items that are in the vehicle cargo. **IT DOESN'T WORK PROPERLY FOR NOW. ITEM ARE SAVED BUT SPAWN ON THE GROUND WHEN YOU GET BACK THE CAR.**

"SaveVehicleHealth": That Boolean allow you to define if you want to be able to save the vehicle health's state.

"SaveVehicleFuel": That Boolean allow you to define if you want to be able to save the vehicle fuel's level.

"ParkInCost": That integer variable allow you to define the amount of money needed to store the vehicle. **Keep in mind, the money used is the same than the money set up for the bank ATM.**

"ParkOutCost": That integer variable allow you to define the amount of money needed to get the vehicle. **Keep in mind, the money used is the same than the money set up for the bank ATM.**

"PayWithBankAccount": That Boolean allow you to define if you want to be able to use bank account to pay for garage fee instead of real money.

"WhitelistedObjects": That string array allow you to define object that can be bypass if detected in the parking spot.

"NPCs": That class array allows you to define each Garage NPC location and parking spot.

"ClassName": Set the object that will be used as NPC. It can be a static object of a survivor.

"Position": Vector position of the npc

"**Orientation**": Vector orientation of the npc

"**ParkingPosition**": Vector position of the parking spot of the npc

"**ParkingOrientation**": Vector orientation of the parking spot of the npc

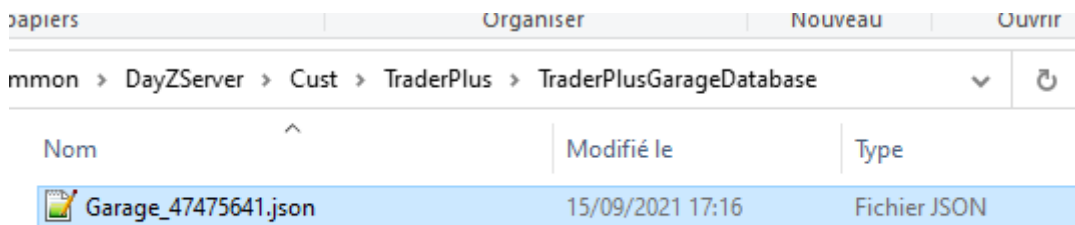
"**Clothes**": String array of classname that a survivor can have to be dressed

Now that we know about the garage config in general, let's take a look at location of the garage database.

In your TraderPlus folder in the profile folder, you should have a folder called:

-TraderPlusGaragedatabase

Inside, you'll find a json file for each player's garage.



The numbers 47475641 are the low part of the steamID64 of the player.

Let's look inside:

```
{
  "OwnerName": "Dmitri",
  "OwnerID": "47475641",
  "ParkingPosition": [
    11612.20703125,
    57.90999984741211,
    14693.2568359375
  ],
  "ParkingOrientation": [
    -33.7625846862793,
    0.000008328365765919444,
    -0.000005567482276092051
  ],
  "Vehicles": [
    {
      "VehicleName": "OffroadHatchback",
      "VehicleID": -1,
      "EngineHealth": 1000.0,
      "FuelAmmount": 42.0,
      "ItemsInCargo": [
        {
          "ItemName": "SparkPlug",
          "Amount": 1,
          "Health": 20.0,
          "ExtraData": -1
        },
        {
          "ItemName": "CarBattery",
          "Amount": 50,
          "Health": 200.0,
          "ExtraData": -1
        }
      ]
    }
  ]
}
```

So we've got inside a couples of variables such as:

"OwnerName": the owner name

"OwnerID": the owner low steamID64

"ParkingPosition": NOT USED FOR NOW

"ParkingOrientation": NOT USED FOR NOW

"Vehicles": the vehicles data array that contains all the vehicles information.

So, this is where the vehicle are store per player. There isn't much to config yourself in this file. Actually, you should not have to edit them to make the garage work. It was just a simple review to show you that.

Now you may wonder how the garage system interacts with TraderPlus?
Here is what you need to know:

- All the vehicles stored in the garage are accessible in the Vehicles category in the inventory panel in the trader.
- You can sell them directly from that category if the trader buy them.

-You can buy vehicles that goes directly into your garage if you enable the variable

"UseGarageToTradeCar": 1, in TraderPlusGeneralConfig.json. if not, vehicles are given as receipt. (see how the receipt system works for me informations).

That's all for this part, thanks for reading this 😊

VI. How does the CarLock system work?

-You'll need to attach an item called **CarLock** in order to secure and own the car. Once a 8 digits (only numbers, not char) password set, you'll be able to lock, unlock your car forever. Password need to be set only once. Your friend can also enter the password once to lock unlock the car.

Password data are stored into a client database accessible in Documents\DayZ\CarLock\CLDatabase\CarLock_X.json.

you may find different json file. There is actually one for each car owner and all the password possible that worked once are inside. You're free to remove thoses files from time to time if it's too big. That will generate a new file but you'll loose all your passwords.

=>To get your own password file, you need to compare the right part of your SteamID64 to the id number of the file to find yours.

So regarding the CarLockSystem:

the CarLock can be detached with a plier if you're the owner or an admin. (to set an admin, check SafeZoneConfig explanations).

-A CarLockPick can be used to attempt to steal the car by removing the CarLock. According to LockPickChance (0.0 to 1.0 ~ 0% to 100%), you'll get the CarLock and a alarm will be displayed for 5 min that can be heard in a 500 meters radius.

-When buying a car at the trader, the car will be already locked and only a password will be required to finished the procedure.

Warning: That system needed variables on CarScript class so it will corrupted all your cars at first load. It may remove others key mod system so if you don't want to use the CarLock system, you'll be able to deactivate it by adding/repacking a small mod to your server list. It's actually a define value that disable all CarLock feature. Keep in mind that without the car lock system, car won't get locked after being deployed at the trading zone so carefull !! link of the mod:

<https://steamcommunity.com/sharedfiles/filedetails/?id=2458852374>

VII. How to convert your TraderConfig.txt to TraderPlusPriceConfig

That part is dedicated to user who already use Dr.Jones trader mod and would like to convert their config to TraderPlus config.

!! Make sure to follow **II. Installation of TraderPlus** before doing that part!!!

I'm happy to announce that most of the trader config can be converted:

What can be converted:

- TraderConfig.txt
- TraderVehiclesParts.txt
- TraderObjects.txt (partially)

What can not be converted:

- TraderVariables.txt

So basically the most important file that is TraderConfig.txt is converted so you won't have to do the price config again. And that, I bet, is making you happy because you're as lazy than I am 😊.

Let's stop the talking and get into the subject:

The conversion will regenerate 4 files located at

DayZServer\profile\TraderPlus\TraderPlusConfig: TraderPlusGeneralConfig.json, TraderPlusPriceConfig.json, TraderPlusIDsConfig.json, TraderPlusVehiclesConfig.json.

The TraderPlusGeneralConfig will get all trader ids, name and positions. I strongly recommend comparing for each trader which npc/object is supposed to be so you can change it by hand. Same goes for the orientation. It must be added by hand.

The TraderPlusPriceConfig will take your TraderConfig.txt, add each classname, trade quantity, buyprice and sellprice to add it to TraderPlusPriceConfig to the dedicated category it was put in. In addition to that, new variables will be added for each products:

-**a coefficient**: between 0.0 and 1.0

-**a maxstock**: it will be either -1 as infinite stock if that's what you want **or** the nominal value of that item indicated in your types.xml

(for ammunition, it will be multiplied by the average number of ammo that spawn in map so it can be higher than current nominal value)

So from that line for example:

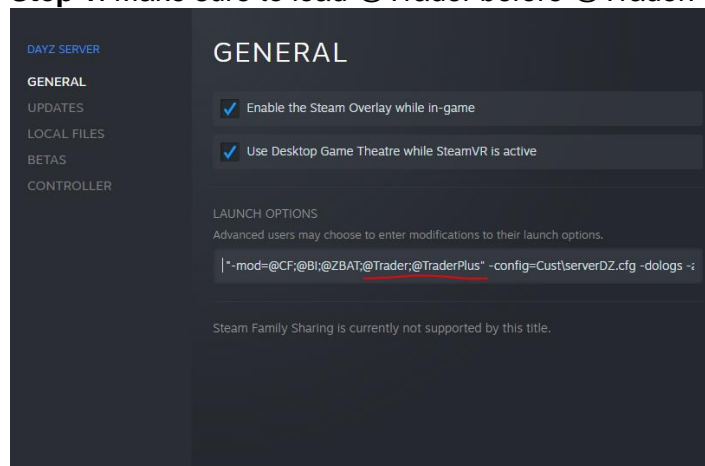
classname,	tradeqty,	buyprice,	sellprice
CombinationLock,	*	700,	300

it will be converted to something like this:

classname,	coefficient,	maxstock	tradeqty,	buyprice,	sellprice
CombinationLock,	0.97309	,60	,0	,700	,300

Now that you understand a bit more what's happening, let's see how we can convert Trader config to TraderPlus config.

Step 1: Make sure to load @Trader before @TraderPlus in your start load order.



Step 2: Go to the TraderPlusConfig folder and open TraderPlusGeneralConfig.json and set to 1 the first variable called : **"ConvertTraderConfigToTraderPlus": 1**

It should look like this:

```
{
  "ConvertTraderConfigToTraderPlus": 1,
  "ConvertTraderConfigToTraderPlusWithStockBasedOnCE": 0,
  "DisableHeightFailSafeForReceiptDeployment": 0,
  "EnableShowAllCheckBox": 1,
  "EnableStockAllCategory": 1,
  "IsReceiptTraderOnly": 1,
  "LockPickChance": 0.30000001192092898,
  "LicenceKeyWord": "Licence",
  "Licences": [
    "Car Licence",
    "Admin Licence"
  ],
  "AcceptedStates": {
    "AcceptWorn": 1,
    "AcceptDamaged": 1,
    "AcceptBadlyDamaged": 1
  }
},
```

Also, you must choose if you want your price config to be converted with infinite stock or with stock value based on each nominal value in your types.xml.

In order to choose, you will have the variable:

-"ConvertTraderConfigToTraderPlusWithStockBasedOnCE": 0,

If set to **0**, the TraderConfig.txt will be converted with infinite stock, if set to **1**, it will be converted with each nominal value define in your types.xml.

Step 3: Start your server, go back to the TraderPlusConfig folder and wait to see TraderPlusGeneralConfig, TraderPlusPriceConfig.json, TraderPlusIDsConfig.json and TraderPlusVehiclesConfig.json been modified (check the modification date). It should be done 1 to 2 minutes after server started. Once it's done. Stop your server and set to 0 **"ConvertTraderConfigToTraderPlus": 0,**

Step 4: Biggest part is done, congrats ! Now I suggest that you take a look at those newly modified files and see if everything seems to be ok. Normally every category you set in TraderConfig can be found in the PriceConfig, every traders ids can be found in IDsConfig and every vehicles parts can be found in VehiclesConfig.

Step 5: Now only thing remaining it to make sure your traders are located in the right place, in addition to your objects if you have some. You're invited to check for each Config file, the section dedicated to it in the documentation to understand all of it 😊

VIII. Files's explanations

In this part, I'll detail how to set and the purpose of each variables for each config files. You're welcome to go directly to what you're looking for but I strongly advise you to fully read the document at least one time so you've got the big picture of how the mod works.

1. TraderPlusGeneralConfig

Let's begin with the general config file, you'll be able to set everything that is linked to the general's mod behavior such as TraderMoney, UI Translation, Accepted States, Traders Positions and Objects Positions.

So once you've opened the file, you'll see something that contains:

ConvertTraderConfigToTraderPlus	Boolean (true : 1 ; false : 0)
ConvertTraderConfigToTraderPlusWithStockBasedOnCE	Boolean (true : 1 ; false : 0)
IsReceiptTraderOnly	Boolean (true : 1 ; false : 0)
LockPickChance	float
AcceptedStates	<Boolean,boolean,boolean> class
Currencies	Array <String, integer> class
Traders	Array <Integer,string,string,string,vector, vector> class
TraderObjects	Array<string, vector, vector> class
EnableShowAllCheckBox	Boolean (true : 1 ; false : 0)
StoreOnlyToPristineState	Boolean (true : 1 ; false : 0)

So let's start the explanations of the purpose of each one:

"ConvertTraderConfigToTraderPlus": That Boolean var allow you to convert Dr.Jones TraderConfig.txt and TraderVehiclesParts.txt to

TraderPlusPriceConfig.json, TraderPlusIDsConfig.json and TraderPlusVehiclesParts. You're invited to check out II. Regarding that part.

"ConvertTraderConfigToTraderPlusWithStockBasedOnCE": That Boolean var allow you to decide if the price config from Dr.Jones TraderConfig.txt must be converted with stock defined from each item's nominal value in your server types.xml or if the price config must have all product with infinite stock. Check the part How to convert trader to traderplus config if you want to use it !

"DisableHeightFailSafeForReceiptDeployment": that Boolean disable the failsafe to avoid cars to be deployed on roof and stuff. Useful for custom map such as Takistan. Set to 1 to disable, set to 0 otherwise.

"UseGarageToTradeCar": that Boolean variable allow you to get your cars directly in your garage if enable. If not, the cars are given as a receipt.

"EnableShowAllCheckBox": that Boolean variable allow you to hide the SHOW ALL check box if you don't want your players to be able to see goods not available in stock. Set to 1 to show the checkbox, 0 to hide it.

"EnableShowAllCategory": that Boolean variable allow you to hide the ALL check category if you want to avoid freeze when updating a trader with large stock file. Set to 1 to show the it, 0 to hide it.

"IsReceiptTraderOnly": That Boolean allow you to define if you want the receipt system for the car to work in the entire map or only inside a trading area.

"LockPickChance": If you want to you the carlock system included in the mod, you'll have the possibility to define the chance value of successful lockpicking a car.

"LicenceKeyword": That's a very important variable. It's a string variable that will make the licence system to work. Because all of you have your own language, you also have your own word to say "Licence". You need to define the word "Licence" in your own language. Then you need to have that key word in every licence you define. If you don't, licences won't be recognized by the mod.

"Licences": That's where you're going to add all the license you want to create. Keep in mind that as mentioned above, it must contain the key word.

Example:

```
"Licences": [  
    "Weapon Licence",  
    "Car Licence",  
    "Admin Licence"  
],
```

As you can see, all the licence contains the key word "Licence".

At this point you probably want to know how to link it to traders and how to allow players to buy the licences.

So don't forget to check **TraderPlusIDsConfig.json** for linking it to traders id.

To allow player to buy them, check **TraderPlusPriceConfig.json**

"AcceptedStates": {

"AcceptWorn": boolean var that allow you to decide if you want to accept item is such condition to be trade. 0: disable ; 1: enable

"AcceptDamaged": same than above

"AcceptBadlyDamaged": same than above

}

"StoreOnlyToPristineState": That Boolean allow you to sell items no matter the state and store it only as pristine. Set it to 1 to enable, set it to 0 to disable.

"Currencies": That array of <string, integer> allows you to decide what kind of currency you want to use for the mod. You can add a line if needed.

Example:

I want to add Lat_25 as a currency:

25 is between 20 and 50 so I'll add a line there as so:

```
{
    "ClassName":
"TraderPlus_Money_Dollar50,TraderPlus_Money_Euro50,TraderPlus_Money_Ruble
50 ",
    "Value": 50
},
{
    "ClassName": "Lat_25",
    "Value": 25
},
{
    "ClassName": "TraderPlus_Money_Dollar20,TraderPlus_Money_Euro20",
    "Value": 20
},
},
```

Now that you understood how to add a currency value that wasn't already set, let's check how to add a currency that already is set.

I like having multiple currencies so I can either pay with Dollars or Euros or Rubbles. I want to add Lat_50 to the already existing currency.

I just have to add to the string Lat_50 as so:

```
"TraderPlus_Money_Dollar50,TraderPlus_Money_Euro50,TraderPlus_Money_Ruble
50,Lat_50",
```

Keep in mind, that the first currency set in each string is the primary currency. It

means that if you sell an item, the trader will give you that primary currency. In our case, that's dollars !

"Traders": This variable is the traders array. That's where you're gonna set up where and what spawn as trader in the map.

You can either set a Object npc (a sign for example) or a real player like SurvivorM_Peter.

So first of all, I need to define the ID: **"Id":**

That id will represent what the npc is selling. Check out TraderPlusIDsConfig.json to get the id value.

"Hey Dmitri, can I have different trader with same Id ?":

Yes you can, it will mean they will all share the same stock.

How to know the id value:



Once you know what the trader should be selling, you need to define it's classname:

"Name": string var that correspond to the classname of the entity you want to use as trader, can be a static object or survivor.

(if you've got a special static object you want to use, make sure that static object extends BuildingBase)

"All right Dmitri, but how do I make my static object extends from BuildingBase ? I'm a noob in scripting :/." **Check VII.**

"GivenName": string var that correspond to the name shown in the trader UI.

"Role": string var that correspond to the role shown in the trader UI
You can put Weapon Trader, Car Trader, whatever you want.

"Position": vector var that correspond to the position in the map of the trader.
(Can be obtained with any modding tool such as DayZ Editor or Community offline mod)

"Orientation": vector var that correspond to the orientation in the map of the trader.
(Can be obtained with any modding tool such as DayZ Editor or Community offline mod)

"Clothes": array <string> that allows you to add Attachments to the trader is it's a Survivor.

Example:

```
"Clothes": [  
    "NBCHood",  
    "NBCJacket",  
    "NBCPants",  
    "NBCBoots",  
    "M4A1"  
]
```

*(For my sake and yours, please **never add a comma at the last element in the array**, in addition don't forget to use a json validator if something doesn't work)*

"TraderObjects": array <string, vector, vector> var that allows you to add custom object to spawn with the trader.

ObjectName: string var of the classname of the object you want to spawn

"Position": vector var that correspond to the position in the map of the object.
(Can be obtained with any modding tool such as DayZ Editor or Community offline mod)

mod)

"Orientation": vector var that correspond to the orientation in the map of the object.

Now we're done explananing the GeneralConfig, let's move on the big one: PriceConfig.

2. TraderPlusPriceConfig

That file is really important because he's the one that will allow you to edit your prices, your stock, how the dynamic price works, etc...

So let's start by opening the file and check how it looks.

You'll find at the top, two variables:

"EnableAutoCalculation": it's a bool variable that allows to calculate automatically the coefficient based on the stock. Check below to get more information about how it works!

"EnableAutoDestockAtRestart": It's a bool variable that allows to enable the destock feature once the stock is reached per product.

"EnableDefaultTraderStock": It's a int variable that allows to automatically fill each trader's stock during a server restart.

You have 3 options:

- "EnableDefaultTraderStock": 0 => no auto filling during a server restart.

- "EnableDefaultTraderStock": 1 => each product will be added to the stock based on the max stock value.

- "EnableDefaultTraderStock": 2 => each product will be added to the stock based on a random value between 0 and max stock value.

!/ Once you set this variable to 1 or 2, it will fill the stock and then the variable "EnableDefaultTraderStock" will be automatically set to 0 so you don't have to do anything. 😊

You want to know more about the **destock variable**? Here is an example:

Example:

In one of your traders, you noticed that the stock of LAR/FAL Mag is full like below:

30rd SG5-K Mag	pristine	9/12
20rd LAR Mag	pristine	32/32
5rd Vaiga Mag	pristine	8/15

It means player can't sell any more fal mag unless someone buy some to free some space in the stock.

That's where the destock feature come into place. When the **"EnableAutoRestockAtRestart"** is set to 1, you'll have the possibility to define for each product how many (in %) it can be reduce after a server restart.

Let's say you want the Fal Mag stock to be reduce by 75% when the stock is full, you'll need to add in the Fal Mag product line, the coefficient 0.75 (= 75%) at the end of the string, just after the sell price like below:

Classname, coefficient, maxstock, trade quantity, buyprice, sellprice, **destock coefficient**

Mag_FAL_20Rnd,0.78,32 ,0 ,100 ,50 ,**0.75**

Keep in mind that if you do not use the destock feature, you don't have to define a coefficient for it. The 7th element in the product line is optional!!!

Now let's talk about the big part, the products :

Normally you should see that products are located in different categories. Each category can contains as many products as you want.

Each product contains 6 elements that are :

-Classname -Coefficient -Max Stock -Trade quantity – BuyPrice – SellPrice

Keep in mind that you'll be able to do:

- Product that have dynamic price based on max stock value
- Product that have a static price and a max stock value
- Product that have a static price and unlimited stock (like Dr.Jones trader)

So let's start by trying to make a dynamic price based on max stock value

1.1. Make a product that have a dynamic price based on max stock value

Again, you've got two possibilities:

- You can let the mod calculate for you the maxstock value based on the nominal value in your types.xml in addition to the coefficient based on the lowest price desired when the stock is reached.
- You can set it manually by playing with the PriceCalculator.xls

If you're lazy, like me, this is how you're gonna make it auto calculate:

let's take an example: I want to add a Weapon form Morty's Weapon: the HK 416A7

Classname, coefficient, maxstock, trade quantity, buyprice, sellprice

HK 416A7, 0.4, -2, 1, 10000, 0.8

So let's try to understand what I set.

-I want the HK to have a **maxstock** based on the nominal value so I put **-2**. (don't try to understand that, just put -2)

-I want the lowest price for the HK to be 40% of the buyprice. (buyprice is always the highest price when the stock is low), so I put **0.4** (= 40%) in the **coefficient**.

-I set to 1 the **trade quantity** because it's a gun and that's actually the max quantity I can set. (You can't put a tradeqty bigger than the max qty of the item or superior to the max count for ammunition => 20 for 7.62x39mm)

You can also set tradeqty to **-1** if you want it to be max qty of the item or to a coefficient such as **0.75**; that mean you can trade if the product has at least 75% of quantity.

Let's talk a bit about the different scenarios we can meet while working on a priceconfig based on the products for the Tradeqty.

There is products without quantity such as **weapons**:

Tradeqty **must** be equal to **0** or **1**.

There is product with quantity such as **ammunitions**:

Tradeqty **can** be equal to **1** (minimal value) to **maxqty (= -1) (20 for 7.62x39mm)**

There is product without quantity but that contains ammunitions such as **mag**

Tradeqty **must** be equal to **0**.

There is product with quantity such as **consumable** like apple, meats etc...

Tradeqty **can** be equal to **1** or **maxqty(= -1)** or a **coefficient (0.0 to 0.99)**.

coefficient will represent a % of maxqty. (useful when you want to sell a % of wolfmeat).

There is product with quantity that contains water such as cooking pot or canteen.

Tradeqty **can** be equal to **0** (we don't care of the qty inside) to **maxqty (item filled completely with water)** **Keep In mind that value between 0 and maxqty will sell portion of the item until quantity reach 0.**

-I set the **buyprice** to **10000** as the highest price. To disable the ability to buy, you can set it to **-1**.Keep in mind that a buyprice below 8 can't be dynamic

-I set the **sellprice** to **0.8**. That mean it will be equal to **80%** of the buyprice. You can also put a number such as **8000** or **7500** or whatever. The two are possible. To disable the ability to sell, you can set it to **-1**.Keep in mind that a sellprice below 8 can't be dynamic in case buyprice is disable.

Once done, I can either config a new product or make the auto calculation happen.

To activate the autocalculation, here is what you have to do:

Step 1: There is a variable called "**EnableAutoCalculation**": in the price config. You need to set it to 1.

Step 2: Then start your server and wait a minute. Once the price config has been modified you can stop your server.

Step 3: Disable the "**EnableAutoCalculation**" with **0**.

Step 4: Check out the line you added and normally you should see that it has changed 😊

Now, if you want to set your product with a custom stock and a custom coefficient then you can open and run the PriceCalculator.xls located in the mod folder in the Folder called Tool.

Once open, you'll be able to set it up to your product values and put your own max stock and your own coefficient (between 0.0 and 0.99999999), I suggest to start with 0.993 and increase with a step of 0.001 to get interesting result.

Once done, you'll have three more variables called

stock = n	DynamicBuyPrice	DynamicSellPrice
20	1	1

You can define for a potential stock of n and get the buyprice and sellprice when the stock is equal to n.

Once your satisfied, put the chosen value into the priceconfig, start your server and enjoy !

1.2. Make a product that have a static price with a max stock value

So let's be back to our previous example with the the HK 416A7

Classname, coefficient, maxstock, trade quantity, buyprice, sellprice

HK 416A7, 1, 10, 1, 10000, 0.8

As you can see, the coefficient is set to 1, that way the price will always be buyprice (10000) and sellprice ($0.8 \times 10000 = 8000$).

You can define the stock based on your nominal value, you can either do a. and then put coefficient equal to 1 or go directly to your types and look for nominal value.

As before, tradeqty equal to 1 because that's the max we can get.
keep in mind that thoses are the possibilities for the trade qty variables:

There is products without quantity such as **weapons**:

Tradeqty **must** be equal to **0** or **1**.

There is product with quantity such as **ammunitions**:

Tradeqty **can** be equal to **1** (minimal value) to **maxqty (= -1) (20 for 7.62x39mm)**

There is product without quantity but that contains ammunitions such as **mag**

Tradeqty **must** be equal to **0**.

There is product with quantity such as **consumable** like apple, meats etc...

Tradeqty **can** be equal to **1** or **maxqty (= -1)** or a **coefficient (0.0 to 0.99)**.

coefficient will represent a % of maxqty. (useful when you want to sell a % of wolfmeat).

There is product with quantity that contains water such as cooking pot or canteen.

Tradeqty **can** be equal to **0** (we don't care of the qty inside) to or **maxqty (= -1) (item**

filled completely with water) Keep In mind that value between 0 and maxqty will sell portion of the item until quantity reach 0.

-I set the **buyprice** to 10000 as the highest price. To disable the ability to buy, you can set it to -1. Keep in mind that a buyprice below 8 can't be dynamic

-I set the **sellprice** to 0.8. That mean it will be equal to 80% of the buyprice. You can also put a number such as 8000 or 7500 or whatever. The two are possible. To disable the ability to sell, you can set it to -1. Keep in mind that a sellprice below 8 can't be dynamic in case buyprice is disable.

1.3. Make a product that have a static price with unlimited stock

So let's be back to our previous example with the the HK 416A7

Classname, coefficient, maxstock, trade quantity, buyprice, sellprice

HK 416A7, 1, -1, 1, 10000, 0.8

If you looked at the two previous a. and b. you'll see that maxstock is set to -1. So you can conclude than -1 will make the stock unlimited.

As before, tradeqty equal to 1 because that's the max we can get.
keep in mind that thoses are the possibilities for the trade qty variables:

Let's talk a bit about the different scenarios we can meet while working on a priceconfig based on the products for the Tradeqty.

There is products without quantity such as **weapons**:

Tradeqty **must** be equal to 0 or 1.

There is product with quantity such as **ammunitions**:

Tradeqty **can** be equal to 1 (minimal value) to **maxqty (= -1) (20 for 7.62x39mm)**

There is product without quantity but that contains ammunitions such as **mag**

Tradeqty **must** be equal to 0.

There is product with quantity such as **consumable** like apple, meats etc...

Tradeqty **can** be equal to 1 or **maxqty (= -1)** or a **coefficient (0.0 to 0.99)**.

coefficient will represent a % of maxqty. (useful when you want to sell a % of wolfmeat).

There is product with quantity that contains water such as cooking pot or canteen.

Tradeqty **can** be equal to 0 (we don't care of the qty inside) to **maxqty (0.0 to 0.99) (item filled completely with water). Keep In mind that value between 0 and maxqty will sell portion of the item until quantity reach 0.**

-I set the **buyprice** to 10000 as the highest price. To disable the ability to buy, you can set it to -1. Keep in mind that a **buyprice below 8 can't be dynamic**

-I set the **sellprice** to 0.8. That mean it will be equal to 80% of the buyprice. You can also put a number such as 8000 or 7500 or whatever. The two are possible. To disable the ability to

sell, you can set it to **-1**. Keep in mind that a **sellprice below 8 can't be dynamic** in case buyprice is disable.

If you want that item to automatically be added to your stock, You'll have to enable AutoCalculation and follow step like 1.2.

1.4 How to allow player to buy a licence

If you're using the licence system, you're probably wonder how to allow players to buy licences. Keep in mind, that licences aren't physical items. You won't get anything when you buy one. It will be added to your bank account file from the server database. Only admins can access that.

Example:

So in whatever category, add the following line:

Car Licence, 1, -1, 1, 10000, -1

As any other item, you add the licence name,
Then 1,
Then -1,
Then 1,
Then the buy price,
Then -1.

You cannot sell it so don't try something weird please 😊.

"Hey Dmitri, I get that, but I would like to restrain the ability to give a licence to only specific player, how do I do that ?"

For that, dear user, you'll have to directly add the licence to the player bank account file located in the **TraderPlusBankDatabase** folder.

You open the file of the player based on his steamID64. You've got the nickname into each file to find it better 😊.

Once you find it, you open it and add it like this:

```
{
  "Name": "TheDmitri",
  "MoneyAmount": 908949,
  "MaxAmount": 1000000,
  "Licences": [
    "Car Licence",
    "Weapon Licence",
    "Admin Licence"
  ]
}
```

That should be all for that file, hope it was understandable, let's keep up and check how the TradeIDsConfig works.

3. TraderPlusIDsConfig

All right, now that we have seen general config and price config, let's take a look at the TraderPlusIDsConfig.

That file shouldn't be too hard to understand. So let's go step by step to explain you everything you need to know about it:

First of all, you need to understand the purpose of this file. It is used for setting up each trader ID so we can know which items the traders are trading, the licenses needed to access the trader and finally the currencies accepted by the trader.

So let's talk about each variables:

"Categories": That's a string array where you're going to define each category that can be trade by a trader ID. If you remember our talk in the general config on how to know the ID we're currently in, here is a reminder:

```

{
  "IDs": [
    {
      "Categories": [
        "Licences",
        "Vegetables",
        "Packaged Food",
        "Meat",
        "Animal Parts",
        "Drinks",
        "Medical Supplies",
        "Medications",
        "Money Exchange"
      ],
      "LicencesRequired": [],
      "CurrenciesAccepted": []
    },
    {
      "Categories": [
        "Melee",
        "Sidearms",
        "Rifles",
        "Shotguns",
        "Submachine Guns",
        "Assault Rifles",
        "Sniper Rifles",
        "Grenades",
        "Ammunition",
        "Magazines",
        "Handguards",
        "Buttstocks",
        "Optics",
        "Suppressors",
        "Compensators",
        "Bayonets",
        "Flashlights"
      ],
      "LicencesRequired": [],
      "CurrenciesAccepted": []
    },
    {
      "Categories": [
        "Tools",
        "Heavy Tools",
        "Electronics",
        "Fire & Lights",
        "Cooking & Hunting Supplies",
        "Hardware Supplies",
        "Seeds & Lime"
      ],
      "LicencesRequired": []
    }
  ]
}

```

Handwritten red annotations in the image:

- A bracket next to the first ID object is labeled "ID = 0".
- A bracket next to the second ID object is labeled "ID = 1".
- A bracket next to the third ID object is labeled "ID = 2 ...".

It means that first element in array is ID:0, then 1, then 2 and so on.

Normally just by seeing the file you should get it but in case you're tired because it's late when you're reading this, let's do an example:

I want to add a category called Morty's Weapons to my weapon trader. So I check where all the guns are. It seems by default it's trader id 3.

So I'll add it this way:

```

{
  "Categories": [
    "Melee",
    "Sidearms",
    "Rifles",
    "Shotguns",
    "Submachine Guns",
    "Assault Rifles",
    "Sniper Rifles",
    "Grenades",
    "Morty's Weapons"
  ],
  "LicencesRequired": []
}

```

(For my sake and yours, please **never add a comma at the last element in the array of a json**, in addition don't forget to use a json validator if something doesn't work)

"LicencesRequired": that's the array that allows you to define a required license to use the trader.

For example, let's say you want to restrict the ability to trade cars to only those who bought the license, you'll need to add in the variable array:

```
"LicencesRequired": [  
  "Car Licence"  
]
```

Keep in mind you need to define the licenses in **TraderPlusGeneralConfig.json**

Before adding them in LicencesRequired.

If you want to add more, that's possible.

```
"LicencesRequired": [  
  "Car Licence",  
  "Admin Licence"  
]
```

"CurrenciesAccepted": that's the array that allows you to restrict the currencies accepted by the trader. (nice isn't it ? 😊)

```
"CurrenciesAccepted": []
```

By default, the array is empty. It means the trader accept all the currencies define in the GeneralConfig file.

Now if you want to restrict the trader to only accept USD ,for example, here is how you can do it:

```
"CurrenciesAccepted": [  
  "TraderPlus_Money_Dollar100",  
  "TraderPlus_Money_Dollar50",  
  "TraderPlus_Money_Dollar20",  
  "TraderPlus_Money_Dollar10",  
  "TraderPlus_Money_Dollar5",  
  "TraderPlus_Money_Dollar1"  
]
```

I recommend adding them from the highest to the lower value, but it's not a must.

In this case, only the bills defined will be working to trade.

I want to underline something about this feature, if have some prices that are under the minimal value set in the currencies accepted, you won't get the money when trading those. So make sure to make your price accordingly to the money you want to use for that trader.

All right, that's all for that file, let's keep up and see how the TraderPlusVehiclesConfig works.

4. TraderPlusVehiclesConfig

So after we opened the file, we should see array of vehicles along side their attachments. Keep in mind, that all attachment filled in that file will be spawn when you deploy a car with a receipt. As I'm not checking and saving parts when packing, players could actually try to glitch it buy selling parts and packing then deploying to get then again. What I advise is to disable the sell of all car parts. That way they won't glitch it.

So let's see how we can add a vehicle in that file.

```
{
  "VehicleName": "CivilianSedan",
  "Height": 0,
  "VehicleParts": [
    "SparkPlug",
    "CarBattery",
    "CarRadiator",
    "HeadlightH7",
    "HeadlightH7",
    "CivSedanDoors_Driver",
    "CivSedanDoors_CoDriver",
    "CivSedanDoors_BackLeft",
    "CivSedanDoors_BackRight",
    "CivSedanHood",
    "CivSedanTrunk",
    "CivSedanWheel",
    "CivSedanWheel",
    "CivSedanWheel",
    "CivSedanWheel"
  ],
  {
    "VehicleName": "Dmns_MRAP",
    "Height": 2,
    "VehicleParts": [
      "SparkPlug",
      "CarBattery",
      "CarRadiator",
      "HeadlightH7",
      "HeadlightH7",
      "MRAPDriverDoor",
      "MRAPCoDriverDoor",
```

```
        "MRAPCargo1Door",
        "MRAPCargo2Door",
        "MRAP_wheel",
        "MRAP_wheel",
        "MRAP_wheel",
        "MRAP_wheel"
    ]
}
```

Not really difficult, copy one and paste it at the end of the current array and change VehicleName to the classname of the vehicle and put the attachment needed for that car.

About the “**Height**”:

0 means height is default set by the hologram during the deployment.

[0;x] means heights is increase by the defined value to fix the deploy above ground.

Normally, cars should be deployed correctly above ground. For vanilla cars and most of them it's working well.

However, since deploying a car like a pokemon seems a bit weird, some modders didn't think of that and something is missing in the vehicle geometry making the hologram of the car appearing in the ground.

So only way I found to make them deploy above ground is to increase the height. So if you see that your modded car is being deployed in the ground, increase the height until it's not.

Keep in mind that the actual Hologram will still appear in the ground, only the final result is fixed that way !

Last thing, if you don't put all the cars you're selling and buying in this files, that will make a full car appear instead of a receipt and fucked up the trader so only sell a car that is in that file !!!

All right, let's take a look on how to set up a safezone for your traders.

5. TraderPlusSafeZoneConfig

First of all, I want to talk a bit about the safezone and its features:

That mod include safezone that disable most of the annoying action that can happen in the trader such as:

- Injection on target

- Desinfect on target
- Force consume on target
- Pack tent
- GiveBlood to target
- Collect blood from target
- Deploy Object
- Restrain yourself or target
- Force feed someone
- Lock doors
- Unpin grenades
- Lock pick car from MuchCarKey

In addition, zombies will be cleaned when they see a player. Animals that comes to the player will also be deleted in each 20s. (the check is in the client, so don't worry, it won't affect server performances)

You're also able to set item that will be cleaned over time and set admins that can bypass disable action while in safezone !

Finally, as server owner, you've got the possibility to place barrel hole or fireplace in the safezone. (don't place them in general config, place them by hand or with an admin tool directly in the safezone).Thoses objects can be ignite on even without wood. That way, it won't make the fire burn for nothing !

That file should be easy to get, You want to add a safezone here is how:

With default file (before adding a new safezone):

```
{
  "EnableNameTag": 0,
  "MsgEnterZone": "YOU HAVE ENTERED THE SAFEZONE!",
  "MsgExitZone": "YOU HAVE LEFT THE SAFEZONE!",
  "MsgOnLeavingZone": "LEAVING ZONE IN:",
  "SafeAreaLocation": [
    {
      "SafeZoneStatut": "Trader",
      "X": 11595.0,
      "Y": 14688.0,
      "Radius": 200.0,
      "Countdown": 30
    }
  ],
  "CleanUpTimer": 1200,
  "ObjectsToDelete": [
    "Paper"
  ],
  "SZSteamUIDs": []
}
```

After adding a new safezone:

```
{
  "EnableNameTag": 0,
  "MsgEnterZone": "YOU HAVE ENTERED THE SAFEZONE!",
  "MsgExitZone": "YOU HAVE LEFT THE SAFEZONE!",
  "MsgOnLeavingZone": "LEAVING ZONE IN:",
  "SafeAreaLocation": [
    {
      "SafeZoneStatut": "Trader",
      "X": 11595.0,
      "Y": 14688.0,
      "Radius": 200.0,
      "Countdown": 30
    },
    {
      "SafeZoneStatut": "Trader 2",
      "X": 1595.0,
      "Y": 4688.0,
      "Radius": 100.0,
      "Countdown": 45
    }
  ],
  "CleanUpTimer": 1200,
  "ObjectsToDelete": [
    "Paper"
  ],
  "SZSteamUIDs": []
}
```

So let's take a look at the variables:

"SafeZoneStatut": that's the name of the safezone, the title in the notification when you get in. Make you to set a different name per safezone or you'll get issues !

"X": X coordinate of the safezone

"Y": Z coordinate in dayz (third value in the position vector) or 2nd value in Izurvie.

"Radius": integer value of the area size of the safezone

"Countdown": Integer value that will be decrease in a timer after leaving the safezone

"EnableNameTag": ~~if you want to see player nametag in the safezone, set it to 1. Set to 0 to disable it. Each player can disable by pressing 'N' when its enable.~~

Discontinued for now. Too much fps taking.

"EnableAfkDisconnect": Set to 1 to enable the afk kicker system, set to 0 otherwise. Basically, it allow to kick from your server player that are afk for too long in the safezone. (Meaning, if they are not moving for a certain amount of time, they will be notified to move or get kicked and 30s later, they will be kicked if they didn't move)

"KickAfterDelay": Set to whatever value in **minute** that will be used to check if the player is afk or not. If you set it to 30, then, every 30 **min**, we'll check if the player moved from last position saved (which was 30 min ago) and notify him to move to avoid getting kicked.

"MsgEnterZone": string var that correspond to the notification message when a player get in the safezone

"MsgExitZone": string var that correspond to the notification message when a player get out of the safezone

"CleanUpTimer": time in second where items selected gets deleted inside the safezone.

"ObjectsToDelete": array<string> that allow you to add items that will be cleaned over time in the safezone

"SZSteamUIDs": array<string> that allow you to add steamID64 of admins that aren't concern about all the blocking action in the safezone such as unpinning a grenade, placing object, etc...

```
"SZSteamUIDs": [  
    "76561198047475641",  
    "76561198047475857"  
]
```

We're getting to the end, nice. I'm getting tired of writing here. All right, let's move on to a little explanation on the Stock files.

6. TraderPlusStock_ID_X

If you're using that mod, it's probably for the stock feature, so it means you'll have the possibility to see what's in each trader stock id. In your DayZServer\profile\TraderPlus\TraderPlusDatabase, you should see a stock file for each id you set in TraderPlusIDsConfig.json. By default, they will be empty because my way of thinking is that it will be filled by player.

In addition to that, you may already see a couple of line in the stock regarding product with unlimited stock because when you enable the autocalculation in TraderPlusPriceConfig, it automatically add them in the stock. If they aren't added, that's not bad. The player will be able to see that he can sell that product and then after the first sell it will be added to the stock.

If you want to add a stock for a couple of items here is how:

```
{  
  "TraderPlusItems": [  
    "WolfSteakMeat 47 0",  
    "Apple 40 0",  
    "Plum 11 0",  
    "WolfSteakMeat 45 3"  
  ]  
}
```

Keep in mind that the last value is the health of the product:

0: pristine

1: worn

2: damaged

3: badly damaged

Also, the stock value, the middle one doesn't always mean it's the quantity of the product. For example, here, the wolfsteakmeat is set at `tradeqty = 0.75` in my price config, so each stock unit will give you or request that you have at least 75% of the quantity of the product.

Also, if you're messing by hand any of the stock file, keep in mind that the format is really important. You need to have EXACTLY one space between each variable, and you must make sure your json is valid. Again, use a json validator like <https://jsonlint.com/?code=> before coming to me saying you do not have anything in stock anymore.

All right, that should be all for that part. Finally let's talk briefly about the logs.

7. TraderPlusLogs

In order to get informations about how the mod is doing, you'll have some logs to help you see if config is loading correctly and also a trace of each players transaction.

Thoses logs can be found in *TraderPlus\TraderPlusLogs*

Keep in mind that you may find answer when something is wrong in there so don't forget to check that out !

For example, for no *fucking* reason, your trader doesn't work anymore. So of course, you think it's because dmitri's shit is buggy so you come to me for help saying my shit is broken. (I'm not saying it's not possible, I'm saying to make sure it's not you first 😊)

So before coming to the discord to me because it doesn't work, open your logs and check the couple of start lines:

```
Creation Time: 2021_4_28-18-41-46
18-41-46 | [INFO] SafeZoneCore - Started !
18-41-46 | [INFO] TraderPlusCore - Started !
18-41-46 | [INFO] DEBUG MODE ENABLED
18-41-46 | [INFO] TraderPlus file exist ! loading...
18-41-46 | [INFO] TraderPlusJsonLoader::LoadFromFile Loaded file: $profile:TraderPlus\TraderPlusConfig\TraderPlusGeneralConfig.json
18-41-46 | [INFO] TraderPlus file exist ! loading...
18-41-46 | [INFO] TraderPlusJsonLoader::LoadFromFile Loaded file: $profile:TraderPlus\TraderPlusConfig\TraderPlusVehiclesConfig.json
18-41-46 | [INFO] TraderPlus file exist ! loading...
18-41-46 | [ERROR] TraderPlusJsonLoader::LoadFromFile ERROR Parsing %1JSON ERROR:
Missing a comma or ']' after an array element.
18-41-46 | [INFO] CheckForSpacesToRemove - Started
18-41-46 | [INFO] CheckForSpacesToRemove - Ended
18-41-46 | [INFO] TraderPlus file exist ! loading...
18-41-46 | [INFO] TraderPlusJsonLoader::LoadFromFile Loaded file: $profile:TraderPlus\TraderPlusConfig\TraderPlusIDsConfig.json ( Fi
18-41-46 | [INFO] TraderPlusServer - Started
18-41-46 | [INFO] TraderPlusBankingCore - Started !
18-41-46 | [INFO] DEBUG MODE ENABLED
18-41-46 | [INFO] BankingSettings file exist ! loading...
18-41-46 | [INFO] TraderPlusJsonLoader::LoadFromFile Loaded file: $profile:TraderPlus\TraderPlusConfig\TraderPlusBankingConfig.json
18-41-46 | [INFO] TraderPlusBankingServer - Started
18-41-46 | [INFO] CarLockCore - Started !
18-41-46 | [INFO] CarLockServer - Started
```

“All right Dmitri, but I’m no scripter, I don’t know this stuff.”

*Well, dear user, I want you to look for the keyword **[ERROR]**. As you can see in the screenshot, there is config file that is showing an error. You need to find that file. You can deduce it’s the PriceConfig because it’s the only one not showing in the logs or verify each traderplus json with <https://jsonlint.com/?code=>*

It will allow you to find the line where the mistake is in order to fix it 😊

IX. How to make static object works with TraderPlus

During your configuration attempt, you may have tried to had static object has trader in the GeneralConfig. And maybe, after starting your server and joining the game, you did not found thoses static objects. It probably means they need a quick fix to make them compatible with the mod.

In order to make thoses static objects compatible you need to make sure they **extends BuildingBase**

For example, I want to make ZT Vending Machine works with the mod:

I need create a file (StaticTrader.c) to add into my server mod in 4_World:

And add in there:

```
class ZT_Vending_Crafting_rusty extends BuildingBase{};
class ZT_Vending_Hunting_rusty extends BuildingBase{};
so on...
```

Here are the instruction:

First of all, I’m assuming you know how to pack and publish a mod, if you don’t know how to do that, find someone who does so you can do last step.

Step 1: Download that mod template: [download link](#)

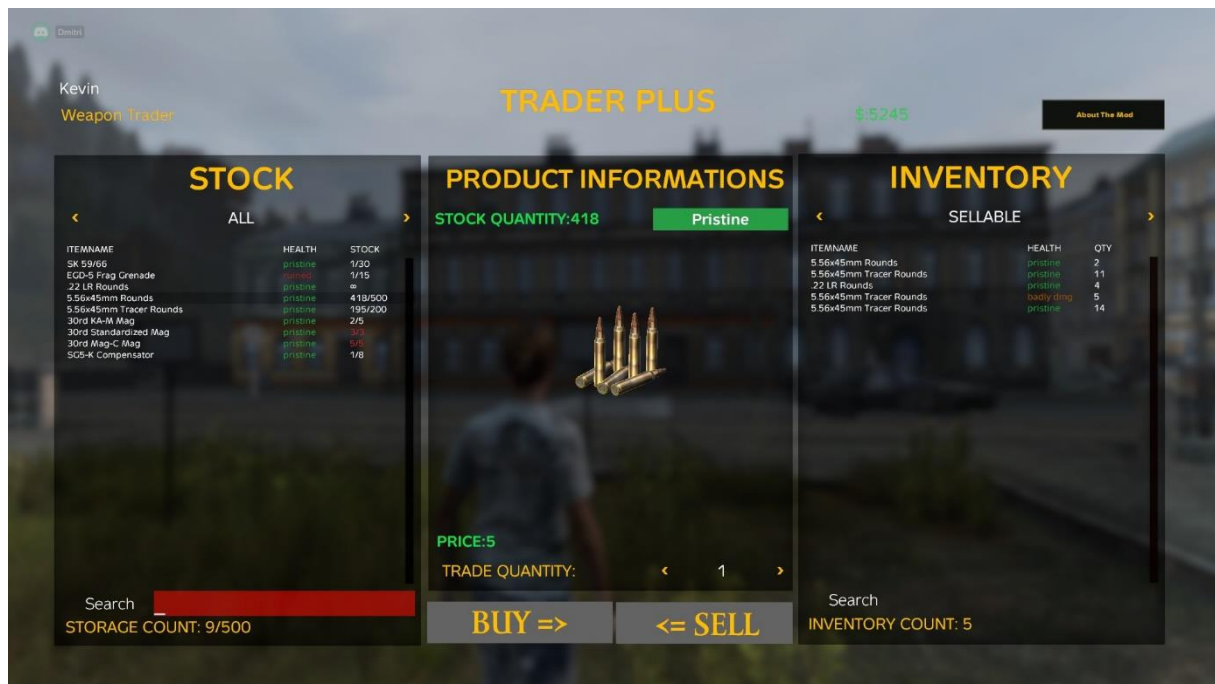
Step 2: Unzip it, go inside scripts\4_World\entities` and open the .c file called *Static Object*.

Step 3: Follow the commented instruction, again, Replace CLASSNAME with your classname of your static object you want to make compatible.

Step 4: Pack and publish the mod into your modpack or standalone, I don’t really care at that point what you do. Just make sure to load that mod before TraderPlus so it will make your static object compatible with the mod.

Now you can’t say that you don’t know. 😊

X. Explanations regarding the UI Interface



The interface will give you three main frames: - STOCK – PRODUCT – INVENTORY

LEFT FRAME:

You'll be able to see what the trader has in stock in the left frame.

Items that a **red STOCK** in the list means the **stock limit is reached**.

Also, you'll be able to see multiple times the same item but in different health condition. (if you want to limit the trade for only pristine or worn check GeneralConfig). It also means each item with a different state can be filled with the defined stock filled in the priceconfig. So your stock can be multiply by 4 if you allow pristine, worn, damaged and badly damaged. Keep in mind that the price is reduce by 25% for each condition.

an item that is worn will be buy/sell = 75% of the pristine one.

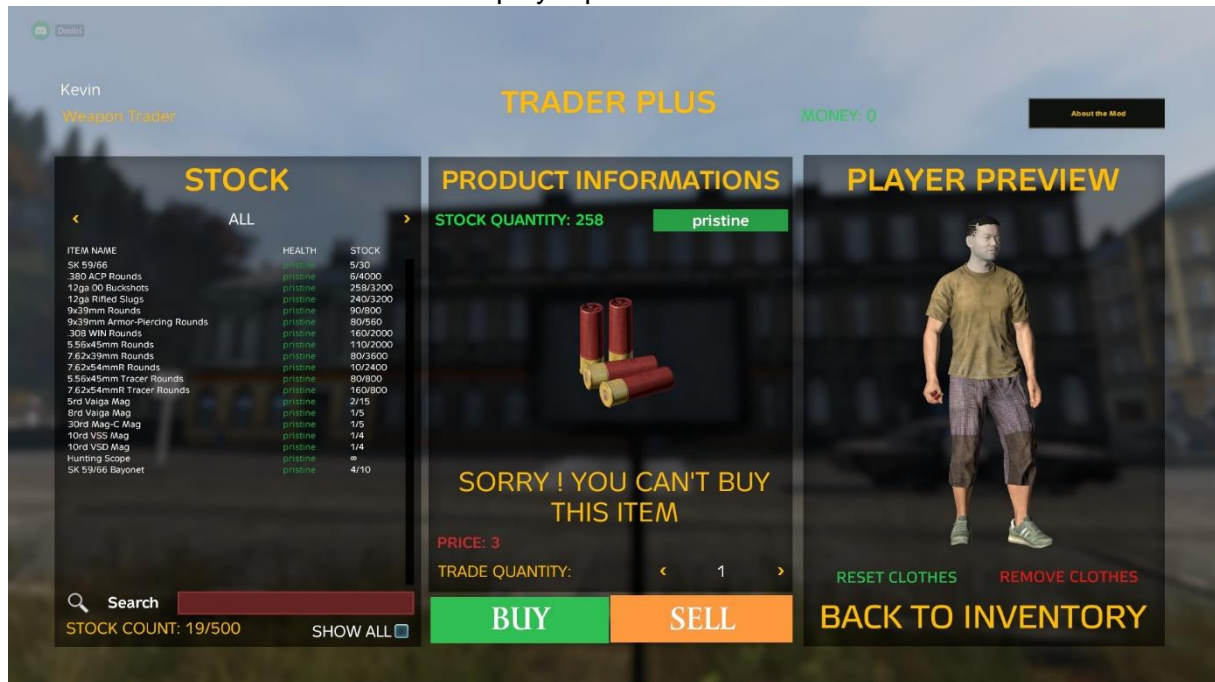
an item that is damaged will be buy/sell = 50% of the pristine one.

an item that is badly damaged will be buy/sell = 25% of the pristine one.

Also, when clicking in a product in the stock list, the inventory frame will be replaced with the player preview frame so you can see in real time the clothes/items you want to wear.

If you look at the player review, you can see that you have 3 buttons:

- RESET CLOTHES will show clothes wear by the player
- REMOVE CLOTHES will remove all clothes of the dummy
- BACK TO INVENTORY will hide the player preview and show the INVENTORY frame



At the bottom of the left frame, you can see the STORAGE COUNT: that's the number of line that you can see in stock. (max value is equal to MaxStockPerID in General Config)

You also now have a checkbox called SHOW ALL so the player can see all the potential item available to trade. Can be hide in general config

MIDDLE FRAME:

You can see the product selected from either the STOCK LIST or the INVENTORY LIST. You'll see quantity of the item or stock quantity on the top, the health condition, the price based on the trade quantity value.

You've got the two buttons BUY and SELL, I don't think that needs more explanations.

RIGHT FRAME:

You're able to see by default all the items that you can sell to that trader. Keep in mind that to avoid some loss, there is a few features:

- Weapons or Tools in the back (shoulder) can't be seen in the list, so you don't accidentally sell your primary gun. Use your hands to sell a gun or put them in your backpack.

- Weapons attachments are given back to the player if he sell a gun that has attachments.
- Clothes, Bag that have items inside, can't be sold. That way you don't accidentally sell your bag or your jacket will all your money.
- Clothes weared by the player can't be seen. It needs to be in hand or in inventory to be sold.
- A third category called HANDS ATTACHMENTS allows you to see your weapon's attachment or the cargo inventory of your item in hand.

NEW FEATURES:

- You're able to quick buy/sell by double clicking in the list
- You can now sell multiples items at a time
- Bullets inside your mag are automatically removed when selling them => thanks to MaD for that
- A basic bank system has been integrated
- You can show all the items that the trader buy/sell by clicking on the checkbox **SHOW ALL**
- You have the ability to see what's in a kit when you click on the kit button that appear when the classname contains "kit".
- When having a weapon in hand, compatibles items are highlight in orange in the stock list

XI. Useful tools to have

To make sure your json syntax is correct: <https://jsonlint.com/?code=>

To add object or trader in the map:

-DayZ Editor: <https://steamcommunity.com/sharedfiles/filedetails/?id=2250764298>

(how to use it: <https://www.youtube.com/watch?v=GJRfJPqANdw>)

-Custom offline mod: <https://github.com/Arkensor/DayZCommunityOfflineMode>

(how to use it: <https://www.youtube.com/watch?v=27KElpExhKM>)

~~To set your TraderConfig or TraderPlusPriceConfig with an interface:~~

~~<https://trader-editor.iwapp.fr/> - <https://traderplus-editor.iwapp.fr/>~~

XII. Conclusion

This is the end of the traderplus documentations, I hope you found an answer to your questions. It took me a full day to make that file so hope it was worth it. Enjoy the mod and have a good time in DayZ.

Donate a coffee: <https://ko-fi.com/thedmitri>

Patreon: <https://www.patreon.com/thedmitri>