



# CfgVehicles Config Reference

From Bohemia Interactive Community

**See also:** CfgMagazines Config Reference, CfgMoves Config Reference, CfgWeapons Config Reference and CfgAmmo Config Reference

## Introduction

### TokenName Config.cpp Reference

Simply put, Token Name & Value pairs are as follows

```
SomeName = SomeValue;
```

Token Names are used extensively throughout ofp 'text' files.

An example is, when creating a model using class statements inside a Config.cpp, to access it later, via the mission editor

```
scope=2;
```

This article details the token-names specifically available within the **CfgVehicles** class of a Config.cpp. The primary focus of this document is the TokenNames used for various aspects of controlling models within **Addons**.

Unlike a command reference, where one meaning fits all, Token names are **not** verbs. The meaning of the name, its effect, can be different, depending on context. You are advised therefore, that these names, as listed, apply *only* to the CfgVehicles class. They may, or may not, have identical meanings (if found), in other classes.

This article lists token names, their value ranges, and their type.

## TokenType

**Integer:** A signed value that will be held in 4 bytes.

**Boolean:** An Integer value of 0 or 1. This is a 'convenience type' in humanly readable text. The engine stores it as an integer.

**String:** Any string value.

For elite only (not ArmA, or OFP) **ALL STRINGS MUST BE ENCLOSED IN QUOTES**.

**Array:** An array [] = {...}

## Contents

Introduction

TokenType

Filenames and Paths

Pbo File conventions

p3d files

paa/pac

jpeg

Sound files

Vehicle vs 'Vehicle'

Config.bin vs Config.cpp

raP files

Alphabetical Order

A

...Action

cargoAction

...AngleX/Y

angle

access

accuracy

acceleration

airFriction

airFrictionCoefs...

ambient

animated

animPeriod

animation...

animationSource...

armor (integer)

armor (float)

armor...

attendant

audible

autocenter

float Array: An array of floats only

string Array: should be obvious

integer Array: ditto

Complex Array: A mixture of multiple types, including embedded arrays.

Variable Array: the number of elements for **this** array in **this** class model can have a varying number of elements (depending on context)

## Filenames and Paths

Filenames are case **insensitive**.

## Pbo File conventions

The root path of the filename is (generally) the name of the pbo

```
model= "\anyPbofile\''anywhere'\''\....";
```

note the pbo extension is inferred

**anywhere** is an (optional) (series of) folder(s) INSIDE the pbo.

Depending on the file referred to, eg a p3d file for a model, the extension (in this case .p3d) is not required.

## p3d files

```
model="\anyPbofilename\anyP3D(.p3d);
```

## paa/pac

wherever these file types are indicated, the default extension is paa

```
icon=\somePbo\anyIcon(.paa);
```

## jpeg

jpeg pictures are usable anywhere a pac (or pac) file is indicated. It is never default.

Exception: Operation Flashpoint Elite specifically, cannot handle jpg.

```
picture="\AnyPbo\AnyPicture.jpg";
```

## Sound files

wss is the default extension if not declared.

```
sound="\AnyPbo\AnySoundFolder\AnySound(.wss);
```

Otherwise use Ogg Vorbis format (extension .ogg).

axis

B

body

brakeDistance

bounding

brightness

C

...CanSee

...CargoAngleY

camouflage

can...

canBeShot

canDeactivateMines

canFloat

canHideBodies

cargolsCoDriver

cast...Shadow

Classes Embedded

cloud...

cobraLight

coefInside...

color

commanding

cost

count

crew

crewCrashProtection

D

dammage...

damageResistance

damper...

deltaT

density

destrType

disappearAtContact

direction

displayName

displayNameShort

driverIsCommander

driverForceOptics

...DustEffect

E

editorCategory

editorPreview

editorSubcategory

...Elev

ejectDamageLimit

ejectDead...

ejectSpeed

## Vehicle vs 'Vehicle'

In this document, a vehicle, is a true vehicle of some kind. Be it a tank, or a car.

A 'vehicle' on the other hand covers cfgVehicles as defined in a config.cpp which would include **men** or **buildings**, since ofp considers them *all* to be 'vehicles'

## Config.bin vs Config.cpp

By convention only, config.cpp is the humanly readable text equivalent of config.bin.

The engine will accept either or **both**.

When both are encountered, it is the text file that is predominant (the raPified file is ignored). (Which is the text file, and which is the 'binary' is moot, see below).

While unusual to have both, it is common for an addon, eg a pbo, to have **both** inside where the config.cpp is a 'work in progress' that might later be 'binarised' to raP, and the cpp removed.

## raP files

So-called 'binarised' or 'binary' files are encoded in raP file format. They are not in the conventional sense a "binary"—machine readable code or data—but rather a more convenient encoded form of the text file (from the engine's perspective). They represent the stripping out of comments and extraneous crud, with bells on. For example, save files are raP encoded, just like config.bin is.

By convention only, a config.cpp is a text file. The engine does not actually care; if it encounters a config.cpp with a raP signature, it is treated as per a config.bin.

The engine unconditionally converts text files to raP encoding before it uses it further. This (obviously) slows game or mission loading and (less obviously) is a source of 'undocumented features'. Prior to finishing your addon, it is a Good Idea™ to 'binarise' the config into raP encoding and rename the cpp file to .txt for documentation purposes.

## Alphabetical Order

### A ...Action

String : defaults ManActCargo

```
commanderAction = "ManActM60CommanderOut";
driverAction = "ManActShipDriven";
gunnerAction = "ManActJeepGunner";
commanderInAction = "ManActTestDriver";
driverInAction = "ManActM113Driver";
gunnerInAction = "ManActM1A1Gunner";
getInAction = ManActGetInTank; // default
getOutAction = ManActGetOutTank; // default
```

Embedded ClassNames

Hatch... Class  
Indicator... Class  
Light Class  
CargoLight Class  
Reflectors Class  
Smoke Class

Turret Class  
Embedded TokenNames  
enableSweep  
envelope  
EventHandlers... Class  
extCameraPosition[]  
extCameraParams[]

F

minFireTime  
fired  
flapsFrictionCoef  
forceHide...  
forceSupply  
formation...  
fov  
...Fov  
fuelCapacity  
fuelConsumptionRate

G

\_generalMacro  
...Gun...  
...GunnerMayFire  
...GunClouds Class  
...GunFire Class  
gearbox[]  
gearRetracting  
gun  
gunAimDown  
gunAxis  
gunBeg  
gunEnd  
gunnerHasFlares  
gunnerName  
gunnerOutOptics...

H

...Height  
has...  
...HeadTurnAI  
hiddenSelections  
hide...  
hitpoint  
hitSound...

## cargoAction

Variable String Array: default {ManActCargo}

```
cargoAction[] = {"ManActM113Medic", "ManActM113Medic", "ManActM1
```

**Note** that unlike driverAction and gunnerAction this is an array.  
(for multiple positions)

The number of elements in this array correspond to the  
#transportSoldier value declared for this class.

see #cargolsCoDriver

## ...AngleX/Y

Float Degrees

```
initAngleX = 0;
minAngleX = -30;
maxAngleX = 30;
```

```
initAngleY = 0;
minAngleY = -100;
maxAngleY = 100;
```

```
initFov = 0.700000;
minFov = 0.420000;
maxFov = 0.850000;
```

used by car.ViewPilot class

## angle

Float: Degrees

```
angle = -240;
```

see #Indicator... Class Or #Hatch... Class

## access

Integer: Determines the manipulability of the class.

```
access=1;
```

0 ReadAndWrite additional values can be added 1  
ReadWrite only adding new class members is allowed 2  
ReadOnly no modifications enabled 3 ReadOnlyVerified no  
modifications enabled, CRC test applied

All B/S classes are **ReadOnlyVerified** and can only be inherited  
into a new class

hour

icon

in/out

init

initT

initYSpeed

insideSoundCoef

interval

irScanGround

irScanRange...

irScanToEyeFactor

irTarget

isBicycle

isMan

K

killed

L

ladders

laser...

landingAoa

landingSpeed

library

LODDriverTurnedIn

LODDriverTurnedOut

M

magazine

magazines

mapSize

marker

material

maximumLoad

maxT

min/max

memoryPointExhaust...

memoryPointGun

memoryPointGunnerOptics

memoryPointGunnerOutOptics

memoryPoint...Missile

memoryPoint...Rocket

memoryPointsGetIn...

memoryPointsGetIn...Dir

memoryPointSupply

memoryPointTrack...

microMimics

minute

model

moves

N

## accuracy

Float: Default value = 0.02.

Determines how easy an object is to identify. Smaller values are easier to detect.

Typically, buildings are 0.2, vehicles, 0.5

**A value of 1000 causes the underlying class to be identified instead.** Eg bushy trees, pink trees and yellow trees, can all be 'identified' as "trees". The description "bushy trees" is then used solely when Editing. See #VehicleClass

```
accuracy = 0.2;           // buildings
accuracy = 0.5;           // small vehicles
accuracy = 0.9;           // truck
accuracy = 3.500000;       // sniper
accuracy = 1.500000;       // Lawsoldier
accuracy = 1000;          // soldier // identify (most) so
```

Relationship to #camouflage:

**accuracy** defines how hard it is for the AI to recognise the correct class of a unit. But even if it can't recognise the correct class it can still recognise it as an enemy and engage it.

**camouflage** defines how hard it is for the AI to see that the unit is there.

Related TokenNames #scope, #accuracy, #displayName, #vehicleClass, #nameSound, #camouflage

## acceleration

Float: metres per second

```
acceleration = 7;
```

## airFriction

Array: 3 separate tokens, airFriction0[], airFriction1[], airFriction2[] that define the X Y and Z component respectively.

```
airFriction2[] = {25, 12, 2.500000};
```

## airFrictionCoefs...

Array: tokens for defining the friction behaviour of planes, introduced with Arma 3 Jets DLC. They define the airfriction for planes with simulation = "PlaneX". Untested if applicable to other objects as well. The values in each array define the X, Y and Z coefficients respectively. The friction force in a certain axis is calculated ingame by  $F_{friction} = v * abs(v) * airFrictionCoefs2 + v * airFrictionCoefs1 + (sign of v) * airFrictionCoefs0$

...NightLights...

name

nameSound

noseDownCoef

nightVision

O

...OpticsColor

...OpticsModel

gunnerOpticsShowCursor

P

passThrough

picture

position

precision

...preciseGetInOut

preferRoads

primary

primary...

proxyIndex

proxyType

R

...Radius

...RotorSpeed

...RotorDive

ReloadAnimations Class

rotL/R

reversed

rotor...

S

...Sensitivity

...Simul/Plan

scope

secondaryExplosion

sensitivity

sensitivityEar

selection

selection...

shape

showgunneroptics

showWeaponCargo

sound

SoundEnvironExt Class

Sounds Various

...Sound

sound...

sound[]

side

simulation

size

airFrictionCoefs0; where "v" is the speed of the plane in that particular axis.

```
airFrictionCoefs2[] = { 0.00100, 0.00050, 0.00006};
airFrictionCoefs1[] = { 0.100, 0.050, 0.006 };
airFrictionCoefs0[] = { 0.0, 0.0, 0.0 };
```

## ambient

Float Array:

```
ambient[] = {0.07, 0.07, 0.07, 1.0};
```

See #Reflectors Class, #Light Class, #CargoLight Class

## animated

Bool : Default Value true

Used by class models that inherit an animated 'vehicle' to turn those aspects of the vehicle off.

```
animated = false;
```

model is animated, or not.

when set true, the animation class, if any, happens, otherwise it doesn't

see animations class

## animPeriod

float: Seconds.

This TokenName is generally used inside AnimationSources class (ArmA only). It's use in OFP:R is as follows

```
animPeriod = 1.33; // a fountain
```

## animation...

String: These animation strings point to rtm files to create the movement.

```
animationFlag = "\AnyAddon\AnyRTM(.rtm)";           // to wiggle fla
animationOpen = "\AnyAddon\AnyRTM(.rtm)";             // to open or cl
animationDrop = "\AnyAddon\AnyRTM(.rtm)";
```

## animationSource...

source  
sourceAddress  
sourceSize  
...Speed  
scud...  
straightDistance  
submerged..

T

...Turn  
TokenNames Embedded  
turning  
turnCoef  
terrainCoef  
threat  
timeToLive  
transport...  
transportAmmo  
transportMax...  
transportSoldier  
transportVehiclesCount  
TransportMagazines Class  
transportVehiclesMass  
TransportWeapons Class  
turretAxis  
type (threat)  
type (animation)  
typicalCargo

U

...UsesPilotView  
uavHacker  
unitInfoType  
unloadInCombat  
upDownL/R

V

vehicleClass  
vehicleClass[]  
view...Shadow  
viewGunnerInExternal

W

...WaterEffect  
weapon  
weapons  
weaponsGroup  
weaponSlots  
wheelCircumference  
wounds

## String

```
animationSourceBody = "Turret_2";
animationSourceGun = "Gun_2";
animationSourceHatch = "hatchCommander";
animationSourceElevation = ""; //new in A3
animationSourceCamElevation = ""; //new in A3
```

## armor (integer)

Float: A strength value of the object, how much external damage can be taken before 'destruction'. It is calculated together with object volume size! (bounding sphere calculated in geometry) Below are general ranges to give some idea.

```
approx strength = (0.27/tgtRadius)^2 / armorInConfig

armor = 3;           // man
armor = 10;          // motorcycle
armor = 20;          // a small car
armor = 15..60;      // aircraft
armor = 150;         // buildings
armor = 150..300;    // bmp
armor = 400..900;    // tank
armor = 300;         // boat
armor = 10000;       // ship
```

If unspecified (or not inherited) the default value is 30

1 bullet will kill armor <=20

2 bullets <=40

3 bullets 50

## armor (float)

Float: The floating version of armor type is only used in Hit... classes, eg HitTurret HitBody

```
armor = 1.4; // a bee
armor = 0.8; // a turret
```

Related TokenNames: #armor (float), #material, #name, #passThrough

## armor...

Float:

```
// for vehicles general
armorStructural = 1; // ranges between 1 and 4.0, default 1
armorFuel = 1.4; // default
armorGlass = 0.5; // default
armorLights = 0.4; // default 0.4 in all models.
armorWheels = 0.05; // default
// for tanks
armorHull = 1;
armorTurret = 0.8;
armorGun = 0.6;
armorEngine = 0.8;
armorTracks = 0.6;
// for men
armorHead = 0.7;
armorHands = 0.5;
armorLegs = 0.5;
// helicopters
```

```
armorHull = 0.5;
armorEngine = 0.6;
armorAvionics = 1.4;
armorVRotor = 0.5;
armorHRotor = 0.7;
armorMissiles = 1.6;
armorGlass = 0.5;
```

## attendant

Bool: Default false

Used for 'vehicles' to repair other 'vehicles'. Thus a hospital has an attendant that can repair soldiers. Adds the action "Heal at XXX"

```
attendant = true;
```

Related TokenNames: #hiddenSelections

## audible

float: Default Value 1

```
audible = .05; // man
audible = 3; // motorcycle
audible = 6; // tank/ship
```

How loud you are. The bigger value, the better the unit is heard.

see #camouflage

## autocenter

bool: Default true

```
autocenter = false; // man
```

## axis

String:

```
axis = osa_poklop_driver;
```

see #Indicator... Class Or #Hatch... Class, IndicatorWatch Class

## B body

String:

```
body = "mainTurret";
```

see #Turret Class

## brakeDistance

Float: default 5 meters per sec

```
brakeDistance = 1;      // man
brakeDistance = 14;     // tank
brakeDistance = 500;    // plane
brakeDistance = 50;     // boat
```

## bounding

String: Memorypoint for Bounding point. Exact purpose unknown. Defined for base class of Tanks. Default memorypoint is the tip of the tank's barrel (usti hlavne).

```
bounding = "usti hlavne";
```

## brightness

Float:

```
brightness = 1.0;
```

See #Reflectors Class, #Light Class, #CargoLight Class

## C . . . CanSee

Integer

Used to describe which sensory inputs are available to which crew members.

Numeric value	CanSee.. define	Description
1	Radar	Enables top left radar (OA) for a vehicle (top centre in arma and OFP).
2	Eye	Effect unknown - maybe affects AI capabilities?
4	Optics	Effect unknown - maybe affects AI capabilities?
8	Ear	Effect unknown - maybe affects AI capabilities?
16	Compass	Enables top centre compass bar and digital compass in A2/OA.
32	Peripheral	Enables small colored blobs at the edge of the screen that indicate nearby units.

Multiple values can be added (or binary ored, giving the same result) together.

```
commanderCanSee = 31; // default
gunnerCanSee = 4+8+16; // default
driverCanSee = 2+8+16; // default
```

```
#define CanSeeRadar 1
#define CanSeeEye 2
```

```
#define CanSeeOptics 4
#define CanSeeEar 8
#define CanSeeCompass 16
#define CanSeeAll 31
#define CanSeePeripheral 32
#define CanSeeRadarC CanSeeRadar+CanSeeCompass
```

```
commanderCanSee = CanSeeAll;
gunnerCanSee = CanSeeOptics+CanSeeEar+CanSeeCompass;
driverCanSee = CanSeeEye+CanSeeEar+CanSeeCompass;
```

## ... CargoAngleY

Float Degrees

Found within vehicle classes.

Orientation of the man sitting in the cargo space.

```
initCargoAngleY = 185; // Truck5t
initCargoAngleY = 90; // M113
```

```
initCargoAngleY = 10; // UH60
minCargoAngleY = -60;
maxCargoAngleY = 120;
```

## camouflage

Float: Default Value = 2.

how difficult to spot. bigger = easier

```
camouflage = 0.6; // snipers
camouflage = 1; // man
camouflage = 4; // trucks
camouflage = 8; // tanks
```

Relationship to #accuracy:

**camouflage** defines how hard it is for the AI to see that the unit is there. The AI will not spot an enemy that has a camouflage setting of 0 (maybe this needs some more testing) even if it stands directly in front of it *as long as the enemy doesn't fire*.

**accuracy** defines how hard it is for the AI to recognise the correct class of a unit. But even if it can't recognise the correct class it can still recognise it as an enemy and engage it.

[Related TokenNames](#) #scope, #accuracy, #displayName, #vehicleClass, #nameSound, #camouflage

## can . . .

### canBeShot

Boolean: Declares whether bullets have any effect

```
canBeShot = true;
```

## canDeactivateMines

Boolean: Default false

part of the man class.

```
canDeactivateMines = true;           // SoldierEngineer
```

## canFloat

Boolean: Default value: false

Used to allow vehicles (such as BMP) not to sink !

```
canFloat = true;
```

## canHideBodies

Boolean: Default false

part of the man class.

```
canHideBodies = true; // SoldierWSaboteur
```

## cargoIsCoDriver

Variable Boolean Array : Default {false};

This token declares which (if any) cargo positions are 'in the front' (ie front windscreen)

```
cargoIsCoDriver[] = {true, true, false};           // 1st two passenger positions for a 5T truck
```

The number of elements in **this** array correspond to the #transportSoldier value declared for **this** class model.

## cast...Shadow

Boolean: Default false

```
castCargoShadow = false;
castCommanderShadow = false;
castDriverShadow = false;
castGunnerShadow = true;
```

Normally used to shadow 'soldiers' standing up in vehicle objects, such as the MG Jeep and Machine Guns.

## Classes Embedded

see #Embedded ClassNames

## cloud...

Various(Strings, Floats, Arrays)

```
cloudletDuration = 0.9;
cloudletAnimPeriod = 1.0;
cloudletSize = 0.1;
cloudletAlpha = 0.8;
cloudletGrowUp = 0.4;
cloudletFadeIn = 0.0;
cloudletFadeOut = 5.0;
cloudletAccY = -0.1;
cloudletMinYSpeed = 0.3;
cloudletMaxYSpeed = 1.5;
cloudletShape = "\ca\data\cl_basic";
cloudletColor[] = {1, 1, 1, 0};
```

See #Smoke Class

## cobraLight

String : Used as a core model for all other 'air' craft.

```
cobraLight = "AnyAddon\AnyP3d(.p3d)";
```

## coefInside...

Float:

```
coefInside = 2; // default
coefInsideHeur = 4.3; // default
```

Used by the static class for non moving objects such as buildings

```
coefInside = 1; // forest
coefInsideHeur = 0.25;
```

## color

Float Array: of floats.

```
color[] = {0.8, 0.8, 1.0, 1.0};
```

This is a fixed four dimension array consisting of the color values for a RGBA model.

See #Reflectors Class, #Light Class, #CargoLight Class, #Smoke Class

## commanding

Integer: part of Turret class

```
commanding = 1;
```

## cost

Float: Default Value 50,0000

This value reflects the attractiveness of the target to the enemy AI. When *all other considerations are equal*. A soldier eg is not interested in Air, despite it's highly attractive cost values. On the other hand, he is interested in a medic, since a medic (in normal configs) has a (while small) a higher cost, than others in the group, including the officer.

```
cost = 10000000;           // an air vehicle is typically this value
cost = 8;                  // a medic;
cost = 1;                  // a grunt
cost = 4;                  // an officer
cost = 0;                  // most buildings;
```

Related TokenNames: type, #cost, #threat

## count

String or Integer: (math formula).

```
weapon = M60;
count = "30*1";
```

this token is always associated in a weapon or magazine couplet, inside a TransportWeapons, or TransportMagazines class.

The value is mostly represented as a string as above (a math formula), or it can be (unusually)

```
count = 2;
```

See At the End of the Day in the TokenNameValueTypes page for an explanation how this duality is possible.

## crew

String: Default Civilian

Reflects whether vehicle is manned, and what with.

```
crew = <ClassName>
crew = "SoldierWPilot";
```

'SoldierWPilot' is a class declared in same file.

## crewCrashProtection

Float

Multiplier of damage to crew of the vehicle. Lower number means better protection.

```
crewCrashProtection = 0.05;
```

## D damage . . .

Variable String Array:

```
damageHalf[] = {"\AnyAddon\AnyPAA.paa", "\AnyAddon\AnyOtherPAA.paa", ...};  
damageFull[] = {"\AnyAddon\AnyPAA.paa", "\AnyAddon\AnyOtherPAA.paa", ...};
```

When 'vehicle' is considered at least half way to destruction, the series of pictures is applied to the model.

## damageResistance

Since ArmA 1.00 Only for AI, to know when to shoot and when not.

```
damageResistance = 0.004;
```

You can calculate it from vehicle armor and hit value of weapon that should be able to harm the vehicle. tgtRadius is radius of virtual bounding sphere calculated in vehicle geometry.  
CfgVehicles\_Config\_Reference#armor\_.28integer.29

```
damageResistance = (HIT*HIT)/Armor*((0.27/tgtRadius)*(0.27/tgtRadius))
```

## damper . . .

Float:

```
damperSize = 0.1;           // default  
damperForce = 3;           // default
```

## deltaT

Float

```
deltaT = -500;
```

See #Smoke Class

## density

Float

```
density = 0.5;
```

See #Smoke Class

## destrType

String: Default Value = DestructDefault

Used for animation, sound, and final 'look' of vehicle.

```
destrType = "DestructNo";           // nothing happens
destrType = "DestructBuilding";    // smoke, explosion
destrType = "DestructEngine";      // smoke only
destrType = "DestructTree";        // smoke, crushing, falls over
destrType = "DestructTent";        // smoke, crushing, flattens
destrType = "DestructMan";
destrType = "DestructDefault";     // =building
destrType = "DestructWreck";
```

Tent and Tree are no longer identifiable when destroyed if not class 'things'

## disappearAtContact

Bool: Default false;

```
disappearAtContact = true;         // FxCartridge
```

## direction

string:

```
direction = "konec L svetla";
```

see #Reflectors Class

## displayName

String: Default value = "Unknown" or "Vehicle"

Selects object uniquely in the Mission Editor, and separately, is used to identify object in game according to #accuracy. For instance when pointing your men to an object while in a mission, it is this *displayName* that is used.

It is good practice to use a stringtable.csv that should accompany your addon so that

1. All names and labels are in one place instead of hunting for them.
2. For language differences

```
displayName = "$STR_DN_AH1Z";
```

Note, that will careful crafting of classes, different colored 'rocks' can all be called the same displayName (rock) in the mission itself. See #scope and #accuracy for details.

See #nameSound for definition of what is actually 'said'

[Related TokenNames](#) #scope, #accuracy, #displayName, #vehicleClass, #nameSound, #camouflage

## displayNameShort

String:

```
displayNameShort = "$STR_DN_UH60";
```

## driverIsCommander

Bool : Default false. When not set, the turret commanding values are used to determine who is the commander. (source (<https://dev-heaven.net/issues/27614#note-10>))

```
driverIsCommander = true;
```

For boat, MG Jeep and helicopters

see #has...

## driverForceOptics

Bool **Description:** Determines if the driver can freely look around inside the vehicle or if he is locked to looking forward through the hatch/optics.

```
driverForceOptics = true;
```

## ...DustEffect

String

Defines the effect that will be spawned at the position of the LandContact points.

```
leftDustEffect = "vbs2_fx_lowTrackDustEffects";
rightDustEffect = "vbs2_fx_lowTrackDustEffects";
```

see #...WaterEffect

## E editorCategory

Eden Editor main category. See Eden Editor: Object Categorization for more details.

```
editorCategory = "EdCat_Structures";
```

## editorPreview

Eden Editor preview picture. See Configuring Asset Previews for more details.

```
editorPreview = "\A3\EditorPreviews_F\Data\myObject.jpg";
```

## editorSubcategory

Eden Editor sub-category. See Eden Editor: Object Categorization for more details.

```
editorSubcategory = "EdSubcat_Drones";
```

## ...Elev

Integers:Degrees

**Description:** How far the mounted weapon and viewpoint (through optics only) can be lowered or raised, and how much the initial elevation is. Values of max and min = 90 effectively allows a weapon to be lowered into the ground/straight into the air.

```
initElev = -80;
minElev = -60;
maxElev = 10;
```

part of any #Turret Class (tanks eg)

## ejectDamageLimit

Float: If the damage value of the vehicle is higher than the ejectDamageLimit, AI will disembark the vehicle.

```
ejectDamageLimit = 0.75;
```

## ejectDead...

Boolean : Default false

```
ejectDeadGunner = false;
ejectDeadCargo = false;
ejectDeadDriver = false;
ejectDeadCommander = false;
```

Causes that unit to 'unmount' the vehicle. Such as an MG nest, or motorcycle.

## ejectSpeed

Array: Speed vector of ejection seat. Therefore only applies to planes.

```
ejectSpeed[] = {0, 0, 0}; // cant eject in a Cessna
```

## Embedded ClassNames

Some Token Names are encountered only within an embedded classname, rather than the main 'vehicle' body.

An embedded classname conveniently adds a unique feature to the 'vehicle'. Such as smoke, such as, a turret for a tank.

Embedded class names are used extensively in animated buildings for the quite logical reason that identical Token Names for door1 cannot be the same as Door2.

Below is a list of embedded classnames that (in most cases) use a TokenName uniquely. It is not found elsewhere and is better served describing that Token Name in the context it is found in, ie, an embedded class.

### Hatch... Class

```
class HatchDriver {
    selection = poklop_driver;
    axis = osa_poklop_driver;
    angle = -100;
};
```

### Indicator... Class

```
class IndicatorSpeed {
    selection = ukaz_rychlo;
    axis = osa_rychlo;
    angle = -240;
    min = 0;
    max = 60 / 3.6;
};
```

### Light Class

```
class Light {
    ambient[] = {0.3, 0.15, 0.0, 1.0};
    brightness = 0.08;
    color[] = {1.0, 0.5, 0.0, 1.0};
    position = ohniste;
    shape = koulesvetlo;
    size = 0.3;
};
```

### CargoLight Class

```
class CargoLight {
    ambient[] = {0.6, 0, 0.15, 1};
    brightness = 0.007;
    color[] = {0, 0, 0, 0};
};
```

### Reflectors Class

```
Reflectors ::LandVehicle {//old
    ambient[] = {0.1, 0.1, 0.1, 1.0};
    brightness = 0.25;
    color[] = {0.9, 0.8, 0.8, 1.0};
    position = L_svetlo;
    size = 0.5;
    direction = konec_L_svetla;
    hitpoint = L_svetlo;
    selection = L_svetlo;
```

```

};

Reflectors { //new
    class Light_1 {
        ambient[] = {10,10,11};
        color[] = {1000,1000,1100};
        size = 1;
        dayLight = 1;
        useFlare = 1;
        intensity = 1;
        coneFadeCoef = 2;
        innerAngle = 89;
        outerAngle = 90;
        position = "light_start_memoryPoint";
        direction = "light_end_memoryPoint";
        hitpoint = "";
        selection = "";
    };

    class Attenuation { //https://community.bistudio.com/wiki/setLightAttenuation
        start = 0;
        constant = 0;
        linear = 0;
        quadratic = 0;
        hardLimitStart = 9;
        hardLimitEnd = 10;
    };
};

}

```

See [setLightAttenuation](#) notes for explanation of attenuation values.

## Smoke Class

```

class Smoke {
    density = 0.5;
    deltaT = -500;
    in = 0.0;
    out = 0.0;
    initT = 1000;
    initYSpeed = 1.7;
    interval = 0.01;
    size = 0.1;
    timeToLive = 100000002004087730000.0;
    //
    cloudletDuration = 0.9;
    cloudletAnimPeriod = 1.0;
    cloudletSize = 0.1;
    cloudletAlpha = 0.8;
    cloudletGrowUp = 0.4;
    cloudletFadeIn = 0.0;
    cloudletFadeOut = 5.0;
    cloudletAccY = -0.1;
    cloudletMinYSpeed = 0.3;
    cloudletMaxYSpeed = 1.5;
    cloudletShape = cl_basic;
    cloudletColor[] = {1, 1, 1, 0};
    class Table {
        class T1 {
            maxT = 0;
            color[] = {0.8, 0.8, 0.8, 1};
        };
        class T2 {
            maxT = 900;
            color[] = {0.3, 0.3, 0.3, 1};
        };
        class T3 {
            maxT = 1000;
            color[] = {1, 0.5, 0, 0.5};
        };
    };
};

```

## Turret Class

(Tanks eg)

```
    class TurretBase {
        body = OtoCvez;
        gun = OtoChlaven;
        gunAxis = OsaHlavne;
        gunBeg = usti hlavne;
        gunEnd = konec hlavne;
        minElev = -4;
        maxElev = 20;
        minTurn = -360;
        maxTurn = 360;
        soundServo[] = {Vehicles\gun_elevate, 0.031623, 1.000000};
        turretAxis = OsaVeze;
    };
}
```

## Embedded TokenNames

see #Embedded ClassNames

## enableSweep

Bool: Default true. Used by Helicopter class

```
enableSweep = false; // /UH60MG
```

Is sweeping over the target a valid technique for given helicopter?

## envelope

## Float Array

```

// Lift (G) based on speed
envelope[] = {
    // speed relative to max. speed -> Lift
    // 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 // rel. speed
    0.0, 0.2, 0.9, 2.1, 2.5, 3.3, 3.5, 3.2, 2.5, 2.0, 1.5, 1.0, 0.9, 0.7, 0.5 // l1
};

```

Info by RKSL-Rock (<http://forums.bistudio.com/showpost.php?p=2037721&postcount=9>);

The top speed is set via the mass. So you will need MLODs to alter it. So it is the same issue as the original QP.

The config maxspeed value is only really used by the AI and the engine as a base variable in some calculations.

You can change the acceleration though by using the envelope command. This will change the vertical speed relative to the maxspeed in the config in steps of 10% of the stated max speed.

This is the envelope from the BKS1 Lynx AH7:

```
envelope[] = {1.7, 2.50, 3.50, 4.50, 5.50, 5.50, 5.15, 5.15, 5.15, 5.15, 4.15, 3.15, 2.15, 2.15}
```

There are normally 15 values representing the vertical lift in m/s at 10% steps from 0 to 140% of the max speed.

So if the max speed is 500 and the first value is always 0:

2nd value = 50kph and 2.5 m/s vertical speed 3rd value = 100kph and 3.5 m/s vertical speed ... 11th value = 500kph and 4.15 m/s vertical speed ... 15th value = 700kph and 1.15 m/s vertical speed.

You wont affect the top speed but it will make it easier to handle.

## EventHandlers... class

Pre-defined object Event Handlers. Property with class matches the handler name.

```
class EventHandlers
{
    init = "myAddon_object = _this select 0;";
};
```



Arma 3 introduced **extended event handlers**. It's possible to define event handlers within unique sub-classes, which prevents compatibility issues when multiple addons wants to set the same handler type to single vehicle class.

```
class EventHandlers
{
    class MyAddon
    {
        init = "myAddon_object = _this select 0;";
    };
    class YourAddon
    {
        init = "yourAddon_object = _this select 0;";
    };
};
```

## extCameraPosition[]

float Array: Default {0, 2, -20};

```
extCameraPosition[] = {0, 5, -30};                                // plane
extCameraPosition[] = {0, 1, -10};                                 // cars
extCameraPosition[] = {0, 1.500000, -9};                         // tanks
extCameraPosition[] = {0, 0.300000, -3.500000}; // man
```

## extCameraParams[]

float Array: Default {0.5, 10, 50, 0.5, 1, 10, 30, 0, 1};

Smoothing of external camera movement

```
extCameraParams[] = {multFactor, speedMin, speedMax, factorSpeeedMin,
factorSpeeedMax, heightMin, heightMax, factorHeightMin, factorHeightMax};
```

```
factor = multFactor * interpolate(speed, speedMin, speedMax, factorSpeeedMin,
factorSpeeedMax) * interpolate(heightAGL, heightMin, heightMax, factorHeightMin,
factorHeightMax)
```

factor == 0 -> camera follow horizon

factor == 1 -> camera follow vehicle heading

factor (0,1) interpolation between horizon and vehicle heading

```
extCameraParams[] = { 0.5, 10, 50, 0.5, 1, 10, 30, 0, 1 }; // helicopters & UAV
```

## F minFireTime

Integer: Default 20 seconds

```
minFireTime = 20;
```

Minimal time spent firing on single target.

## fired

String: Event Handlers

```
fired = "_this exec \"\AnyAddon\AnySQS.sqs\""; // note the ""
```

## flapsFrictionCoef

Float: Default 0.5

```
flapsFrictionCoef = 2; // can be integer too.
```

## forceHide...

Boolean : Default false

```
forceHideDriver = true; // shilka  
forceHideGunner = true; // tank  
forceHideCommander = true;
```

Disables the turnout option for that particular crew member

## forceSupply

Boolean : Default

```
forceSupply = true;
```

Found only in Weapon holders. Is used to make the weaponholder disappear when empty.

## formation...

Integer:

```
formationX = 10;           // default meters
formationZ = 20;           // default meters
formationTime = 10;         // default seconds
```

One or both X Z values are used to keep objects separated (in meters) depending on wedge, echelon, V, single line, formations.

## fov

Float

```
fov = 0.85;           // Cessna
fov = 0.85;           // man
fov = 1.0;            // truck
```

## ... FOV

Float

```
minFov = 0.420000;      // man
maxFov = 0.850000;
minFov = 0.600000;      // truck
maxFov = 1.400000;
```

also used by ViewPilot Classes

## fuelCapacity

Float: Default Value 0. Has nothing to do with liters.

Arma 3: for a vehicle using tankX simulation consumption seems to be exclusively determined by engine rpm. With a 700 idle-rpm and 3000 max-rpm engine - a fuelCapacity of 1 will give you 640 seconds operation time at idle rpm, and 300 seconds of operation at maximum rpm. If the idle rpm was 200, you would get 865 seconds of operation, the max rpm time would not change. The amount of thrust from controllerinput is completely irrelevant.

A2 examples:

```
fuelCapacity = 50;          // motorcycle
fuelCapacity = 100;         // car
fuelCapacity = 700;          // tank
fuelCapacity = 1000; // air
```

The required fuelCapacity value to achieve a specific operation TIME (in seconds) at a specific engine RPM (revolution per minute) can be estimated with:

```
fuelCapacity = (8E-7 * RPM + 0.001) * TIME
```

## fuelConsumptionRate

Integer: Default Value 0.01 Has no unit. Has no effect on tankX vehicles in Arma 3

## G \_generalMacro

String: in Arma 3. Has no purpose whatsoever. It is generated automatically by BI developer tools for debugging See here for explanation (<https://forums.bistudio.com/topic/179378-what-does-generalmacro-do/>)

### ...Gun...

Integer:degrees

```
minGunElev = -60;
maxGunElev = 60;
minGunTurn = -5;
maxGunTurn = 5;
minGunTurnAI = -30;
maxGunTurnAI = 30;
```

### ...GunnerMayFire

Boolean: Default false

```
outGunnerMayFire = true;           // m113
inGunnerMayFire = false;
```

### ...GunClouds Class

This is a reference to the external WeaponCloudsGun classes

most models simply inherit the defaults

```
class GunClouds: WeaponCloudsGun{};
class MGunClouds: WeaponCloudsMGun{};
```

here is an example of over-rides for class man

```
class GunClouds:WeaponCloudsGun {
    cloudletGrowUp = 0.200000;
    cloudletFadeIn = 0;
    cloudletFadeOut = 0.400000;
    cloudletDuration = 0.200000;
    cloudletAlpha = 1;
    cloudletAccY = 2;
    cloudletMinYSpeed = -10;
    cloudletMaxYSpeed = 10;
    interval = 0.020000;
    size = 0.300000;
    sourceSize = 0.015000;
};
```

### ...GunFire Class

This is a reference to the external WeaponFireGun classes

most models simply inherit the defaults

```
class GunFire: WeaponFireGun {};
class MGunFire: WeaponFireMGun {};
```

## gearbox[]

Array

**Description:** Changes acceleration and max speed values of certain Vehicles. If changed, impact to the acceleration is minor (if notable at all!) to catastrophic (worsened acceleration/maxspeed). Best to leave as-is.

```
gearbox[] = {-18, 0, 110, 16.15, 14.44, 13.33};
```

## gearRetracting

Bool: Default true

```
gearRetracting = false; // Cessna
```

more correctly viewed as HasRetractingGear.

## gun

String:

```
gun = OtochLavon;
```

see #Turret Class

## gunAimDown

Float

**Description:** Aircraft only. How far a hardpoint-mounted gun "aims down"; high values allow for strafing runs without the need to nose-dive down too much, but also shift the targeting reticle in the aircraft's HUD; if too high, targeting reticle is only visible in 3rd Person (if available).

```
gunAimDown = 0.07;
```

## gunAxis

## gunBeg

## gunEnd

see #Turret Class

## gunnerHasFlares

Boolean

```
gunnerHasFlares = false;
```

## gunnerName

String

```
gunnerName = "$STR_POSITION_CREWCHIEF";
```

## gunnerOutOptics...

String

```
gunnerOutOpticsModel = "";
```

Array

```
gunnerOutOpticsColor[] = {0, 0, 0, 1};
```

Boolean

```
gunnerOutForceOptics = false;
```

Boolean

```
gunnerOutOpticsShowCursor = false;
```

## H

## ...Height

Integer:

```
minHeight = 5; // Min height in metres above sea-level.  
maxHeight = 50; // Max height above sea level.  
avgHeight = 10;
```

## has...

Bool:

```
hasDriver = true; // default  
hasGunner = false; // default  
hasCommander =true; // default
```

Depending on vehicle model.

Controls the 'get in' commands.

see #driverIsCommander

## ...HeadTurnAI

Integer Degrees

```
minHeadTurnAI = -70;      // man class
maxHeadTurnAI = 70;
```

## hiddenSelections

String Array: Default none.

Here you can define selections on the model, which are not shown at mission startup. This is useful for creating variations of one model, where the different selections are left out with hiddenSelections. (ex: Soldiers have the hidden selection "medic", as they should not have a red cross at their body):

Aside from this, the hidden selections are used for runtime texture assignment. Every element in the hidden selection-array corresponds to an index, with the first element being 0. Only 10 elements are allowed in the array and are numbered 0-9.

```
hiddenSelections[] = {"pruh"}; // bmp/tank
```

Try `this setObjectTexture [0, "\data\duha.pac"]` in the initline of a BMP and you'll see which parts are meant by the selection "pruh".

Don't forget to add the model and the selections in the CfgModels, as the hidden selections won't work otherwise (at least not after binarizing).

## hide...

Bool

```
hideProxyInCombat      = false;           // default (true for tanks); if true, disable proxy
hideUnitInfo          = false;           // default see [[#unitInfoType]]
hideWeaponsCargo      = false;           // default
hideWeaponsCommander  = true;            // default
hideWeaponsDriver     = true;             // default
hideWeaponsGunner     = true;             // default
```

## hitpoint

String:

```
hitpoint = "L_svetlo";
```

See #Reflectors Class

## hitSound...

```
hitSound1[] = {voices\Hit11, 0.056234, 1};
...
hitSound20[] = {voices\Hit30, 0.056234, 1};
...
hitSounds[] = {hitSound1, 0.05, hitSound2, 0.05, ...}
```

hitSounds is used by the engine to reference user generated hitSound...s

the number of user generated sounds, is limited only by the author.

## hour

part of the IndicatorWatch class

```
class IndicatorWatch {
    hour = hodinova;
    minute = minutova;
    axis = osa_time;
    reversed = 0;
};
```

## icon

String: Default Value = "unknown\_object.paa"

This value is used by the map editor to show the building or vehicle when editing. It is not normally visible during game play. (but can be)

The icon can be any jpg, paa, or pac file. paa is default. Note that Elite cannot handle jpegs.

```
icon = "\AnyAddon\AnyPAA(.paa);
```

Related TokenName(s): #mapSize

## in/out

Float: These TokenNames are used inside the #Smoke Class

```
in = 0.0;
out = 0.0;
```

## init

String: Event Handlers The Event Handler receives an array **\_this** (the exact content of which still has to be documented), of which the first element contains the vehicle (e.g. **\_this select 0**).

```
init = "[(_this select 0)] exec ""\AnyAddon\AnySqs.sqs""";
```

## initT

Integer: This TokenName is used inside the #Smoke Class

```
initT = 1000;
```

## initYSpeed

Float: This TokenName is used inside the #Smoke Class

```
initYSpeed = 1.7;
```

## insideSoundCoef

Float: default 0.5

```
insideSoundCoef = 0.05; // air vehicles are 2%
```

## interval

Float: This TokenName is used inside the #Smoke Class

```
interval = 0.01;
```

## irScanGround

Bool: Default true.

Probaly used to stop ai looking down

```
irScanGround = false; // tanks
```

## irScanRange...

Integer: Default 0

```
irScanRange = 4000; // outdated (since Resistance)
irScanRangeMin = 500; // tanks general
irScanRangeMax = 4000;
irScanRangeMin = 2000; // air
irScanRangeMax = 10000;
irScanRangeMin = 4000; // a vulcan
irScanRangeMax = 10000;
```

## irScanToEyeFactor

Integer: Default 1

```
irScanToEyeFactor = 2; // air
irScanToEyeFactor = 5; // shilka
```

## irTarget

Bool: Default true.

Used for (some) buildings so that they don't show up on tank radar. Save for very limited exceptions, this token should be enabled for all vehicle units.

Disabling it will make the vehicle NOT to be engaged by infantry anti-tank gunners nor vehicle weapons. It will only be engaged by small arms, and only if its armor value is low enough for AI infantry to think it can damage it by shooting at it.

```
irTarget = false; // man
```

## isBicycle

Bool: Default false.

this value inside the motorcycle class is used to turn it on when inheriting to a real bicycle

## isMan

Bool: Default true.

```
isMan = 1;
```

Defined within the man class

## K killed

String: Event Handlers

```
killed = "[(_this select 0), 1, 1, 0, 0] exec ""\AnyAddon\AnySQS.sqs"""; // note the '
```

## L ladders

Embedded Array: The ladders array is used to declare one or more ladder pairs inside the model.

```
ladders[] = { {"start", "end" } };
ladders[] = { {"start1", "end1" }, {"start2", "end2" } }; // two Ladder building
//...etc
```

Note that unfortunately, the 'start' and 'end' labels are arbitrary for each model. You cannot have a generic ladder building as such.

## Laser...

Boolean: Default false

Determines if a vehicle has laser capability.

```
laserScanner = true;
laserTarget = true;      // used as a generic class of All {} to default things true
```

## 1andingAoa

MathFormula String Default  $10 * 3.1415 / 180$

Landing Angle of Approach. Used by plane class

```
landingAoa = 7*3.1415/180;      // Cessna
```

## 1andingSpeed

Integer: Default 0 Kph.

```
landingSpeed = 75;      // plane vehicles
```

## library

String: This TokenName is associated with the library class of a vehicle (if any). It is used to give a 'memo' style full screen info on the given object. Example of use:

```
class Library {
    libTextDesc = "$STR_LIB_AH1Z";
};
```

## LODDriverTurnedIn

Integer Defines the LOD that is to be used when the Driver is turned in. Value=-1 seems to default to the standard LOD (for Turret's that is Gunnerview).

```
LODDriverTurnedIn = -1
```

Value: -1=Default LOD; 1=first resolution LOD; 1000=Gunnerview; 1100=Pilotview; 1200=Cargoview.

## LODDriverTurnedOut

Integer Defines the LOD that is to be used when the Driver is turned out. The same rules as with LODDriverTurnedIn apply.

```
LODDriverTurnedOut = -1
```

## M

## magazine

Array: used in TransportMagazines class this specifies the type of ammo in the magazine and the amount.

```
{
    magazine = "30Rnd_556x45_Stanag";
    count = "30*1";
}
```

Each of these couplets are contained within a 'weapon' classname within the magazines class itself. Example

```
class TransportMagazines {
    class _xx_M16 {
        magazine = M16;
        count = 30*1;
    };
    class _xx_M60 {
        .....
    };
    class _xx_PK {
        .....
    };
    ...
};
```

## magazines

Variable string Array: default {}

see #weapons for proper description

## mapSize

Float: Default Value = 10 (meters).

Used for the map editor to show the icon associated with this model class.

There is no x y component to this, the measurement units are meters.

To create a circular, rectangular or other 'non square' image, you need to make a pac file for the icon with, a transparent layer.

```
mapSize = 0.7; // small object such as a tree
mapSize = 4; // a smallish building
```

The Mission Editor map is not very good in this regard (as noted above). However, the main problem is the base classes of the engine. The mapSize= for most objects, *particularly* buildings, is a one shoe fits all. A forest, eg, is the same mapSize= as a tree !!!

Later Oem addons, \*generally\* are better and more specific in this regard, making position and fitting a little easier.

## marker

String : used as a core for all other air craft when looking on map.

```
marker = "\AnyAddon\AnyP3d.(p3d)";
```

## material

Integer:

Used in Hit... classes, eg HitTurret HitBody

```
material = 50;
```

Related TokenNames: #armor (float), #material, #name, #passThrough

## maximumLoad

The maximumLoad value defines the number of units of cargo storage.

```
maximumLoad = 2000;
```

An item's storage value (the amount of space it takes up) is defined by its mass value. For example:

```
//CfgMagazines >> "Titan_AT"
mass = 100;
```

The following parameters are used by just Eden:

```
transportMaxMagazines
transportMaxWeapons
transportMaxBackpacks
```

## maxT

Integer: This TokenName is used inside the #Smoke Class

```
maxT = 0;
```

## min/max

String Degrees Math Formula

```
min = 0;
max = 60 / 3.6;
```

Note for readability, these values are *normally* expressed as strings which the engine itself will convert.

see Indicator Class

## memoryPointExhaust...

String:

```
memoryPointExhaust = "exhaust_start";
memoryPointExhaustDir = "exhaust_end";
```

## memoryPointGun

String:

```
memoryPointGun = "machinegun";
```

## memoryPointGunnerOptics

String

```
memoryPointGunnerOptics = "gunnerview";
```

## memoryPointGunnerOutOptics

String

```
memoryPointGunnerOutOptics = "commander_weapon_view";
```

## memoryPoint...Missile

String:

```
memoryPointLMissile = "Missile_1";
memoryPointRMissile = "Missile_2";
```

## memoryPoint...Rocket

String:

```
memoryPointLRocket = "Rocket_1";
memoryPointRRocket = "Rocket_2";
```

## memoryPointsGetIn...

String:

```
memoryPointsGetInCargo = "pos_cargo";
memoryPointsGetInCoDriver = "pos_codriver";
memoryPointsGetInCommander = "pos_commander";
memoryPointsGetInDriver = "pos_driver";
memoryPointsGetInGunner = "pos_gunner";
```

## memoryPointsGetIn...Dir

String:

```
memoryPointsGetInCargoDir = "pos_cargo_dir";
memoryPointsGetInCoDriverDir = "pos_codriver_dir";
memoryPointsGetInCommanderDir = "pos_commander_dir";
memoryPointsGetInDriverDir = "pos_driver_dir";
memoryPointsGetInGunnerDir = "pos_gunner_dir";
```

## memoryPointSupply

String:

```
memoryPointSupply = "supply";
```

## memoryPointTrack...

String:

```
memoryPointTrackFLL = "tyreTrack_1_1l";
memoryPointTrackFLR = "tyreTrack_1_1r";
memoryPointTrackBLL = "tyreTrack_1_2l";
memoryPointTrackBLR = "tyreTrack_1_2r";
memoryPointTrackFRL = "tyreTrack_2_1l";
memoryPointTrackFRR = "tyreTrack_2_1r";
memoryPointTrackBRL = "tyreTrack_2_2l";
memoryPointTrackBRR = "tyreTrack_2_2r";
```

## microMimics

String: Man class

This token refers to the class within the external cfgMimics class

```
microMimics = Micro;
```

## minute

part of the IndicatorWatch class

see #hour

## model

String: Default Value= "empty.p3d"

Location in the addon where the p3d model resides.

```
model = "\AddonName\anyP3D(.p3d)";
```

Where a class is not a visible 'object', such as a BaseClass. You can specify

```
model = "";
```

this cuts down on load and access.

Related TokenName: #simulation, #reversed

## moves

String: Default = NoDefaultMoves

applies to the man class and refers to a cfgMoves class

```
moves = CfgMovesMC; // civilian, in fact, there isn't another
```

## N

## ...NightLights...

Float:

```
spotableNightLightsOff = 0.05; // default
spotableNightLightsOn = 4; // default
visibleNightLightsOff = 0.1; // default
visibleNightLightsOn = 0.2; // default
```

## name

String:

```
name = "motor";
```

Used in Hit... classes, eg HitTurret HitBody

Related TokenNames: #armor (float), #material, #name, #passThrough

## nameSound

String: Default value: "Target"

Namesound is used by the AI audio to indicate verbally where to go, what 'vehicle' to attack. Thus the audio speech "move to 'rock' 3 o'clock" is derived from the *nameSound*.

```
nameSound = "rock";
```

Many different nameSounds can be selected. Here are just a few.

```
nameSound = crew;
nameSound = target;
nameSound = tank;
nameSound = house;
```

The actual descriptive text accompanying the 'sound' can be different see #displayName

Typically, a collection of buildings (eg) will be configured as follows

```
class GenBuilding : NonStrategic {
    scope=private; // internal use only
    VehicleClass="Some Editor Group Name";
    namesound="house";
```

```
};

class YellowBuilding: GenBuilding {
    scope=public;
    name="YellowBuilding";
    model="Some yellow building.p3d";
};
```

Related TokenNames #scope, #accuracy, #displayName, #vehicleClass, #nameSound, #camouflage

## noseDownCoef

Float: Default 1.0 used by Plane class

```
noseDownCoef = 0.025; // Cessna
```

How much the nose drops when banking.

*Note: Not used in ArmA any more*

## nightvision

Bool: Default false

```
nightVision = true;
```

present in lawsoldier, sniper, saboteur, tank, and air crew

*Note: obsolete, not used, NVG item is used instead*

## O ...Opticscolor

Float Array : Default {0, 0, 0, 1};

```
driverOpticsColor[] = {0, 0, 0, 1};
gunnerOpticsColor[] = {0.910000, 0.230000, 0.230000, 1};
commanderOpticsColor[] = {0.910000, 0.230000, 0.230000, 1};
```

## ...OpticsModel

String: Default ""

```
commanderOpticsModel= "\AnyPbo\AnyP3d(.p3d)";
driverOpticsModel = "\AnyPbo\AnyP3d(.p3d)";
gunnerOpticsModel= "\AnyPbo\AnyP3d(.p3d)";
```

## gunnerOpticsShowCursor

Bool

```
gunnerOpticsShowCursor = true;
```

## P passThrough

Bool: Default true

If true, destroying this part of the vehicle will destroy the whole vehicle. If false, damage will only affect this part of the vehicle.

This value is only taken into account in ArmA 2. It was ignored by the game engine in previous games (that includes OFP).

```
passThrough = 1;
```

Used in Hit... classes, eg HitTurret HitBody

Related TokenNames: #armor (float), #material, #name, #passThrough

## picture

String: Default: iaston.paa

I think this is for briefing Information icon click

```
picture = "\AnyAddon\AnyPAA(.paa)";
```

"Picture" is used in config.cpp to display the 128 x 64 graphic file of the unit in the HUD and the bottom of the screen while in game. A .pac file can be used also. If using a .pac make your alpha channel pink - RGB 255, 0, 255. Pink is invisible in .pac file.

## position

String:

```
position = "ohniste";
position = "L svetlo";
```

See #Reflectors Class, #Light Class

## precision

```
precision = 1;           // man
precision = 200;         // air
precision = 50;          // boat
```

## ... preciseGetInOut

Integer: Defines location of getin point. 0 = use memoryPointGetIn , 1 = use model center as get in point, 2 = use proxy of crew position as get in point. Default is 0.

```
preciseGetInOut=0; //for vehicle crew
cargoPreciseGetInOut=0; // for passengers
```

## preferRoads

Bool: Default false

```
preferRoads = true; // all vehicles
```

## primary

Boolean: For models that can have, (but don't necessarily do have) multiple internal objects of the same type (turrets eg). One of them, is declared the main, or primary turret. There may indeed be only one 'turret'. (Note: this entry has been replaced by the **primary...** entries. Please use those for your turret definitions to ensure maximal compatibility.)

```
primary = true;
```

## primary...

Boolean A vehicle can have multiple turrets, but one of them normally is the primary gunner turret and one the primary vehicle commander turret (this can be the same turret or two different turrets, but you cannot have several primary gunner or commander turrets). Please use these entries instead of the more general **primary** entry.

```
primaryGunner = false;
primaryObserver = true; // Commander
```

## proxyIndex

Integer

```
proxyIndex = 2;
```

## proxyType

String

```
proxyType = "CPCommander";
```

R

## ...Radius

Float:

Represents a circular area (radius in meters from **centre** of object for action to take place

```
getInRadius=2.5; // default
getInRadius=3.5; // tank
getInRadius=10; // ship
'''supplyRadius''' = 2.5; // for fuel dumps ammo trucks etc
```

## ...RotorSpeed

float: Helicopters

```
mainRotorSpeed = 1.0; // default
backRotorSpeed = 1.5; // default
//
mainRotorSpeed = 1.0; // ch47d
backRotorSpeed = -1.0;
```

## ...RotorDive

float: Helicopters, Default 0.0

```
minMainRotorDive = -9; // CH47D
maxMainRotorDive = 15;
neutralMainRotorDive = -5;
minBackRotorDive = -15;
maxBackRotorDive = 9;
neutralBackRotorDive = -5;
```

## ReloadAnimations Class

General usage in Helicopters, this class is 'looked for' by the engine. Various models, but normally only one, are specified as follows

```
class ReloadAnimations {
    class SomeThingA {
        ...
    };
    ...
    class SomeThingZ {
        ...
    };
}
```

here is an example for an Mi24 helicopter

```
class MachineGun30E {
    weapon = MachineGun30E;
    angle0 = 0;
    angle1 = -2 * 3.141592654;
    multiplier = 500;
    type = rotation;
    animPeriod = 0.500000;
    selection = gatling;
    begin = usti hlavne;
    end = konec hlavne;
};
```

## rotL/R

defined within a tank.Wheels class to describe the series of small wheels on each side

```
rotR[] = {koll1, koll2, koll3, koll4, koll5, koll6, koll7, koll8};
rotL[] = {kolP1, kolP2, kolP3, kolP4, kolP5, kolP6, kolP7, kolP8};
upDownL[] = {koloP1, podkoloP1, koloP2, podkoloP2, koloP3, podkoloP3, koloP4, podkoloP4, koloL1, podkoloL1, koloL2, podkoloL2, koloL3, podkoloL3, koloL4, podkoloL4, koloL5, podkoloL5, koloL6, podkoloL6, koloL7, podkoloL7, koloL8, podkoloL8};
upDownR[] = {koloL1, podkoloL1, koloL2, podkoloL2, koloL3, podkoloL3, koloL4, podkoloL4, koloL5, podkoloL5, koloL6, podkoloL6, koloL7, podkoloL7, koloL8, podkoloL8};
```

## reversed

Boolean: Default true

Normally, models ARE reversed with respect to how they present on the screen after editing with tools like oxygen (eg) This overrides the default. See #model

```
reversed = false;           // class thing
```

## rotor...

String: Helicopters

```
rotorBig = "vrtule_velka";      // cobra
rotorBigBlend = "vrtule_velka_bl";
rotorSmall = "vrtule_mala";
rotorSmallBlend = "vrtule_mala_bl";
```

## S

## ...Sensitivity

Float: Default 1.0 Used by Plane Class

```
aileronSensitivity = 0.33;          // Cessna
elevatorSensitivity = 0.1;          // Cessna
wheelSteeringSensitivity = 1.0; // default
```

## ...Simul/Plan

float: in meters

```
steerAheadSimul = 0.5; // default
steerAheadPlan = 0.35; // default
steerAheadSimul = 0.2; // man
steerAheadPlan = 0.2;

predictTurnSimul = 1.2; // default
predictTurnPlan = 1.2; // default
predictTurnSimul = 3; // ship
predictTurnPlan = 3;
```

## scope

Integer: Default value = private.

```
scope = public;
```

Scope in concept is the same as the C++ reserved words of public, protected and private.

Defines normally exist at the top of a well written config.cpp to make meanings clearer.

```
#define private 0
#define protected 1
#define public 2
```

The meaning of each is as follows

### **private:**

Only other classes inherit this class. It is not createVehicle'able, nor can the class be accessed via the Mission Editor.

private is a common method of grouping base characteristics together and inheriting all those common characteristics into a class that can be viewed or accessed.

### **public:**

Any classes declared public are CamCreateable, **and** selectable via the Editor.

### **protected:**

Identical to public, except the class will not be listed in the mission editor. A very common form of use for this is

```
class vegetables {
    scope = protected;
    VehicleClass = "Fruit and Onions";      // a generic group in mission editor
    displayName = "Generic Vegetable";
    namesound = whatever;
    icon = "vegetableIcon.paa";
    // put any other common characteristics in here
};
class GreenOnions : vegetables {
    scope = public;
    displayName = "Green Onion";
}
class PurpleOnion : vegetables {
    ...
}
```

The effect here is to reduce (considerably) not only the amount of typing, but memory storage too.

It is only the public classes that are listed in the Mission Editor.

So in above example while each one of these has a separate *displayName*=, they are ALL 'grouped' in the Editor's "Fruit and Onions", and **all** of them will be referred to *in the mission* as "Generic Vegetable". If you follow along, you should fully expect to see another protected class called "Generic Fruit"

In the above example, it is just as useful to declare the base class '*private*'. But, see below, and see #accuracy

Protected classes are CamCreatable in the mission.sqm.

Protected classes are immensely useful to 'hide' obsolete models that are still required to maintain compatibility with older missions. Ie older missions will still be playable, but newly created ones will only 'get at' the newer improvements specified in a public class. This

```
class OriginalThing {
    scope=protected;
    ...
    // Lots of original things
    ...
};
class ImprovedThing : OriginalThing { // <<< inherits it all
    scope=public;
    ...
    // Lots of new improved things
    ...
};
```

Related TokenNames #scope, #accuracy, #displayName, #vehicleClass, #nameSound, #camouflage

## secondaryExplosion

Integer: Default disabled (-1)

```
secondaryExplosion = -1;
```

## sensitivity

Float: Default 1.0

```
sensitivity = 0.6;           // cars
sensitivity = 2;             // sniper
```

The higher the sensitivity value for an unit, the better it can see.

The effect is not linear, but logarithmic. So for an unit with value 1 vs one with value 3 it must be 1.7 times closer (square root of 3). This example does not take into account other factors like zoom, binocular use, optics, camouflage and accuracy (knowsAbout value needed to detect the type and side).

1 equals roughly to 100 meters, 2.5 to 210m, 3 to 250m.

OA 1.62 state:

```
class All
    sensitivity = 2.5;
    sensitivityEar = 0.0075;
class AllVehicles: All
class Land: AllVehicles
class LandVehicle: Land
class Car: LandVehicle
    sensitivity = 3;
class Motorcycle: LandVehicle
    sensitivity = 3;
class Bicycle: Motorcycle
class Tank: LandVehicle
    sensitivityEar = "0.0075 /3";
class APC: Tank
class Man: Land
    sensitivity = 3;
    sensitivityEar = 0.5;
class Animal: Man
class Air: AllVehicles
class Helicopter: Air
class Plane: Air
class Ship: AllVehicles
class AH6X_EP1: AH6_Base_EP1
    sensitivity = 2;
class Ka137_Base_PMC: Helicopter
    sensitivity = 2;
```

## sensitivityEar

Float: Default 0.0075

Sets how well can the given unit hear others. The bigger the value, the better the hearing.

```
sensitivityEar = 0.13; // man
```

## selection

String:

```
selection = L svtlo;
```

See #Reflectors Class, #ReloadAnimations Class (Arma Only), #Hatch... Class, #Indicator Class

## selection...

String:

```
selectionBackLights = "light_back";
selectionBrakeLights = "light_brake";
selectionFireAnim = "muzzleflash";
selectionHRotorStill= "mainRotorStatic";
selectionHRotorMove = "mainRotorBlurred";
selectionVRotorStill = "tailRotorStatic";
selectionVRotorMove = "tailRotorBlurred";
selectionFabric = "latka";
```

The selections

```
selectionLeftOffset = "PasOffsetL";
selectionRightOffset = "PasOffsetP";
```

are used to animate tank track textures. You must have the appropriate Named Selections in the model and the selection must have an rvmat in order to work.

## shape

Float: This TokenName is used inside the #Light Class

```
shape = "koulesvetlo";
```

See #Light Class, #Reflectors Class

## showgunneroptics

Boolean:

```
showgunneroptics = 0;
```

## showWeaponCargo

Boolean:Default ?

```
showWeaponCargo = true;
```

Found inside weapon holders (ammo boxes)

OFP: If it's set to true player won't be able to put weapons/magazines to container.

## sound

String: Used by animated objects (campfire eg) to give sound effect. Or, simply by ambients(wolves)

```
sound = "Fire";
sound = Fountain;
sound = OwlSfx;
```

## SoundEnvironExt Class

```
class SoundEnvironExt {
```

This is a Class used within the man class. Some/ None, or all sections of it can (of course) be over-ridden by inheritance.

The external class **CfgManActions** accesses the **SoundEnvironExt** class defined (or inherited) for any 'man' (soldier eg).

External class **CfgManActions** contains many clauses

```
soundOverride = fallbody; // eg
```

for instances of when it wants sounds for that 'action' to take place.

thus, you can define none some or all arrays (within class with your SoundEnvironExt{}) for the following

```
normalExt[] =
normal[] =
road[] =
rock[] =
water[] =
gravel[] =
sand[] =
drygrass[] =
grass[] =
forest[] =
mud[] =
wood[] =
metal[] =
snow[] =
hallway[] =
fallbody[] =
laydown[] =
standup[] =
crawl[] =
```

### Example

```
snow[] = {
    { People\snow_L, 0.000032, 1 },
    { People\snow_R, 0.000032, 1 }
};
```

snow is a Variable length Sound Array

see sounds[]

## Sounds Various

Complex Array:

All the following use

```
nameOfSound[] = {"\AddonName\AnySound(.wss)", 0.00000, 1, 1};
```

- 1st parameter: Sound file (path). \*.wss is default, \*.ogg can be specified.
- 2nd parameter: Change of volume compared to the strongest sound that is audible at the camera's (player's) position. It means that the strongest sound is played on 100% and this sound is played weaker (calculated from dB value, distance, occlude, ...). If this sound is alone in sound scene then it is played on 100% every time.
- 3rd parameter: Speed of playing, 1 = normal speed/time of playing, 2 = 2 times slower, with half lower pitch.

## ... Sound

```
flySound[] = {"\AddonName\AnySound.wss", 0.00000, 1, 1};
singSound[] = {"\AddonName\SoundFile.ogg", 0.031623, 1, 1};
scudSound[] = {weapons\rocketflying, 316.227783, 0.20000};
scudSoundElevate[] = {vehicles\gun_elevate, 0.01000, 1};
```

## sound...

```
soundCrash[] = {"\AnyAddon\AnySound(.wss)", 0.01000, 1};
```

## default values

```
soundCrash[] = {Vehicles\crash, 0.316228, 1};
soundDammage[] = {"", 1, 1};
soundEngine[] = {"", 1, 1};
soundEnviron[] = {"", 1, 1};
soundLandCrash[] = {Explosions\intoground, 0.316228, 1};
soundWaterCrash[] = {Explosions\intowater, 0.316228, 1};
soundGetIn[] = {Vehicles\get_in, 0.000316, 1};
soundGetOut[] = {Vehicles\get_out, 0.000316, 1};
soundServo[] = {Vehicles\gun_elevate, 0.01000, 0.500000};
soundGear[] = {"\AnyAddon\AnySound(.wss)", 0.316228, 1}; // no default
additionalSound[] = {\AnyAddon\AnySound(.wss), 0.00000, 1}; // man only
```

## sound[]

### used by animations

```
sound[] = {"\anyPbo\AnySound(.wss)", 10.00000, 1};
```

## side

Integer: Default Value: NEUTRAL.

The side, when declared, sets the 'vehicle' to east, west, resistance, civilian. The effect differs according to the 'vehicle' itself. Eg soldiers, versus hospitals or repair trucks that can only be used by same side, if specifically sided.

Well written missions (and configs) use defines at top of file to make this more legible.

```
#define NO_SIDE -1
#define EAST          // (Russian)
```

```
#define WEST 1           // (NATO)
#define RESISTANCE 2     // Guerilla
#define CIVILIAN 3
#define NEUTRAL 4
#define ENEMY 5
#define FRIENDLY 6
#define LOGIC 7

side = EAST;
```

NEUTRAL is the general case for all objects.

NO\_SIDE is used for ambient seagulls and wolves (see #sound). It differs from NEUTRAL in that the engine spends no time looking for interaction with other objects.

## simulation

String: Default value: invisible.

The engine behaviour with this #model.

```
simulation = "SeaGull";
simulation = "thing";
simulation = "fire";
simulation = "flag";
simulation = "house";
simulation = airplane, helicopter, tank // eg
```

Related TokenName: #model

## size

Float: This TokenName is used inside the Smoke class and Reflector class

```
size = 0.1;
```

part of Reflectors class

## source

String: found in animationSources

```
source = "reload";
source = "time";
```

## sourceAddress

String: found in animationSources

```
sourceAddress = "loop";
```

## sourceSize

Float: This TokenName is used inside the #Light Class

```
sourceSize = 0.015;
```

## ... Speed

Float: In Kph.

Setting maxSpeed for infantry units too low might cause problems when working with waypoints (units won't be able to fully complete them or won't move towards them).

```
minSpeed = -0.5;           // range 0->1
maxSpeed = 80;             // default
maxSpeed = 30;             // tractor
maxSpeed = 60;             // boat
```

## scud...

String: Default ""

```
scudLaunch = scudlunch.rtm;
scudStart = scudstart.rtm;
scudModel = scud_strela_proxy;
scudModelFire = scud_strela_ohen;
```

## straightDistance

INTEGER:

```
straightDistance = 50;
```

## submerged..

Float: default 0.0

Almost all 'objects' are **not** submerged. This parameter is used to hide objects such as effects

```
submerged = -0.5;           // thing effect
submergeSpeed = 0.25;       // positive value makes it go deeper (sound goes quieter)
```

## T

## ... Turn

Integers Degrees

```
initTurn = 90;
minTurn = -70;
maxTurn = 70;
```

part of any Turret class (tanks eg)

see#gun

## TokenNames Embedded

see #Embedded ClassNames

### turning

Boolean:

```
turning = 1;
```

### turnCoef

Float: Default 2.0

```
turnCoef = 6.0; // truck
turnCoef = 9.0; // tractor
```

### terrainCoef

Float: Default 3.0

```
terrainCoef = 6.0; // skoda
```

### threat

float Array: Default Value {0.700000, 0.500000, 0.300000};

How threatening you are to unit types {Soft, Armor, Air}, respectively.

The ai for this model selects targets of opportunity, based on these values.

```
threat[] = {1, 0.0500000, 0.050000};           // soldier
threat[] = {1, 0.900000, 0.100000};           // Law soldier
threat[] = {0.900000, 0.700000, 0.300000};    // bmp
```

Related TokenNames: type, #cost, #threat

### timeToLive

Float: Default 10,000,000,000.0; // seconds?

```
timeToLive = 20;           // thing effects (bullets)
timeToLive = 100000002004087730000.0; // campfire
```

See #Smoke Class

### transport...

## transportAmmo

Integer: Used by supply 'vehicles' to determine total amount available for entire mission.  
Once depleted...

A 'vehicle' in this case can be a genuine repair truck, or, a building.

```
transportAmmo = 300000;           // reammo truck
transportAmmo = 10000000;          // bigship

transportFuel = 3000;              // refueltruck
transportRepair = 20000000;         // repairtruck
```

## transportMax...

These parameters appear not to be used in Arma 3 and are only included for backwards compatibility. See maximumLoad to define a vehicle's storage capacity.

```
transportMaxMagazines = 50;        // car
transportMaxWeapons = 10;           // motorcycle
transportMaxMagazines = 5;          // tank
transportMaxWeapons = 0;
transportMaxMagazines = 50;          // air
transportMaxWeapons = 10;
transportMaxMagazines = 20;          // boat
transportMaxWeapons = 3;
transportMaxMagazines = 100;          // ship
transportMaxWeapons = 200;
transportMaxMagazines = 200;          // truck
transportMaxWeapons = 50;
transportMaxMagazines = 100;          // apc
transportMaxWeapons = 20;
transportMaxMagazines = 200;          // helicopter
transportMaxWeapons = 50;
transportMaxMagazines = 500;          // ammo boxes
transportMaxWeapons = 2000;
```

Values vary depending on exact type of 'boat' eg.

These TokenNames ate used to indicate how many units of each type an object can hold.  
Most objects can't hold anything.

## transportSoldier

Integer: Default Value= 0.

Number of 'passengers' this vehicle can carry. The value does **not** include the driver, **nor** any vehicle positions such as gunner (if any) or commander(if any). See the JeepMg below for a 3 person vehicle which can only transport ONE soldier.

```
transportSoldier = 3;               // jeep/car
transportSoldier = 50;              // large ship
transportSoldier = 6to10;            // SmallShip
transportSoldier = 3;               // a10
transportSoldier = 8to12;             // helicopters
transportSoldier = 2or3;              // ambulance
transportSoldier = 8;                // apc/bmp
transportSoldier = 12;               // truck
transportSoldier = 1;                // jeepmg
transportSoldier = 1or2;              // repair type truck
```

to make the game 'interesting', similar vehicles on different sides, can carry non equivalent numbers.

## transportVehiclesCount

Integer : default 0

```
transportVehiclesCount = 15; // a Carrier (big ship)
```

## TransportMagazines Class

## transportVehiclesMass

Integer: Default 0

```
transportVehiclesMass = 0; // not used by any model
```

## Transportweapons Class

## turretAxis

String:

```
turretAxis = OsaVeze;
```

see #Turret Class

## type (threat)

Integer: Default Value: Armored

This indicates the threat type of the 'vehicle'.

In well written configs, the 3 possible values are declared as defines at top of file for legibility

```
#define VSoft 0
#define VArmor 1
#define VAir 2

type = VAir;
```

Vehicles (and buildings) are armoured, humans are 'soft' and aircraft (obviously) are air

[Related TokenNames](#): type, #cost, #threat

## type (animation)

**old** Unknown: Default Value: rotation

This appears in the ReloadAnimations class

```
type = rotation;
```

## typicalCargo

Variable String Array:Preloads vehicle with units (if auto)

This array can have zero or more strings.

```
typicalCargo[] = {"Soldier", "Soldier", "SoldierLAW", "SoldierLAW"};
```

Note that these are *ClassNames* of soldiers.

## U ...UsesPilotview

Bool: : Default false

```
gunnerUsesPilotView = false; // some choppers (mi17)  
commanderUsesPilotView = true;
```

## uavHacker

Bool

Allows unit to hack UAVs. Used on UAV operators in vanilla (Arma 3)

```
uavHacker = 1;
```

## unitInfoType

String

```
unitInfoType = "UnitInfoSoldier";
```

Certain 'vehicles' hide this information.

see #hideUnitInfo

## unloadInCombat

bool: default true

```
unloadInCombat = false;
```

All true vehicles will cause ai to disembark when in combat.

## upDownL/R

wheels class see #rotL/R

## V vehicleClass

String: Default Value= "Objects"

Used for Mission Editor basic category selection

```
vehicleClass = "Air";
vehicleClass = "Support";
vehicleClass = "My Great Addon";
```

Related TokenNames #scope, #accuracy, #displayName, #vehicleClass, #nameSound, #camouflage

## vehicleClass[]

String Array:

```
vehicleClass[] = {Men, Car, Armored, Air, Support, Camera, Objects, Ammo, Sounds, Mines};
```

## view...Shadow

Boolean:

```
viewCargoShadow = true;
viewGunnerShadow = true;
```

## viewGunnerInExternal

Boolean: default false

```
viewGunnerInExternal = true; // for some turrets and M113
```

## W ...WaterEffect

String

Defines the effect that will be spawned at the position of the LandContact points when vehicle is in/on the water.

```
leftWaterEffect = "vbs2_fx_lowWaterEffects";
rightWaterEffect = "vbs2_fx_lowWaterEffects";
```

see #...DustEffect

## weapon

String:

```
weapon = "M197";
```

Selects a weapon from the CfgWeapons class

## weapons

Variable String Array: default {};

Weaopns and Magazines contain individual description of what a 'vehicle' IS carrying.

```
weapons[] = {"FFARLauncher", "TwinM134"};           // an aircraft
weapons[] = {CarHorn}; // truck

weapons[] = {M21, LAWLauncher, Throw, Put}; // soldier
magazines[] = {M21, M21, M21, M21, HandGrenade, HandGrenade, LAWLauncher};
```

The amount of weapons (and magazines) that can be carried by the 'man' is determined by #weaponSlots

The names M21, Carhorn etc are external references to the cfgWeapons / cfgMagazines classes

## weaponsGroup

Integer:

The weapons grouping for a vehicle, determined by enumerated weapon category definitions.

```
weaponsGroup1 = WEAPONGROUP_CANNONS + WEAPONGROUP_MGUNS;           // A
weaponsGroup2 = WEAPONGROUP_ROCKETS; // A
weaponsGroup3 = WEAPONGROUP_AAMISSILES + WEAPONGROUP_ATMISSILES + WEAPONGROUP_MISSILES; // A
weaponsGroup4 = WEAPONGROUP_BOMBS + WEAPONGROUP_SPECIAL; // A
```

#define WEAPONGROUP_CANNONS	1	// Cannons
#define WEAPONGROUP_MGUNS	2	// Machine Guns
#define WEAPONGROUP_ROCKETS	4	// Rockets
#define WEAPONGROUP_AAMISSILES	8	// Anti-Air Missiles
#define WEAPONGROUP_ATMISSILES	16	// Anti-Tank Missiles
#define WEAPONGROUP_MISSILES	32	// All / other Missiles
#define WEAPONGROUP_BOMBS	64	// Bombs
#define WEAPONGROUP_SPECIAL	128	// Laser Designator + Misc

## weaponslots

Integer:

Weaponslots apply to man class (soldier, civilian, etc)

It indicates the 'gear' capacity.

```
Soldier: 1 + 4 + 12*256 + 2*4096 + 2 + 8*16; // all soldiers/civilians
Medic : 1 + 4 + 8*256 + 2*4096 + 2 + 4*16;

weaponSlots = ;
1 = primary weapon
```

```
2 = handgun slot  
4 = secondary weapon (launcher)  
16 = handgun magazines (8x)(or grenades for M203/GP-25)  
256 = magazine slots (12x / 8x for medics)  
4096 = goggle slot (2x)  
131072 = ?
```

## wheelCircumference

Float Default 2.513 meters

```
wheelCircumference = 8; // tractor
```

## wounds

string Array:

```
wounds[] = {xicht_a.paa, xicht_a_zranen, ...};
```

this token is used in the man class to show a series of pac/paa files depending on damaged state of the body.

Retrieved from "[http://community.bistudio.com/wiki?title=CfgVehicles\\_Config\\_Reference&oldid=103421](http://community.bistudio.com/wiki?title=CfgVehicles_Config_Reference&oldid=103421)"

Categories: Operation Flashpoint: Modelling | Operation Flashpoint: Editing | ArmA: Addon Configuration | OFP Reference Lists | Arma 3: Editing

- 
- This page was last modified on 10 June 2017, at 00:55.
  - This page has been accessed 423,356 times.