

TankMover

```
{  
    DECIMAL mySpeed  
    - Top Speed  
}
```

TankNonMover

- No Parameters

DropshipMover

```
{  
    DECIMAL mySpeed  
    - Top speed(in metres per second)  
  
    DECIMAL myFlyingHeight  
    - Height above ground the Dropship will try to maintain when moving.  
}
```

HoverMover

```
{  
    DECIMAL mySpeed  
    - Top speed(in metres per second)  
}
```

HoverNonMover

- No Parameters

VehicleMover

```
{  
    DECIMAL mySpeed  
    - Top speed(in metres per second)  
}
```

VehicleNonMover

- No Parameters

WalkerMover

```
{  
    DECIMAL mySpeed  
    - Top speed(in metres per second)  
}
```

InfantryMover

```
{  
    DECIMAL mySpeed  
    - Top speed(in metres per second)  
}
```

InfantryNonMover

- No Parameters

CopterMover

```
{  
    DECIMAL mySpeed  
    - Top speed(in metres per second)  
  
    DECIMAL myFlyingHeightWhenMoving  
    - Height above ground the Unit will try to maintain when moving
```

DECIMAL myFlyingHeightWhenIdle
- Height above ground the Unit will try to maintain when **not** moving
}

NonShooter
- No Parameters

StraightShooter
{

DECIMAL myMinRange
- Minimum firing range. Unit can not attack anything closer to itself than this

DECIMAL myMaxRange
- Maximum firing range. Unit can not attack anything farther away than this

TEXT myFireSound1
- Name of sound file to play when firing

TEXT myFireSound2
- Name of sound file to play when firing

TEXT myFireSound3
- Name of sound file to play when firing

DECIMAL myTargetSpreadRadius
- Determines the radius of a circular area around the target within which the unit's shots are randomly scattered when firing. A smaller radius means shots are more likely to hit their target

FILEList myAcquiredTargetFeedback
- Name of sound file to play when unit acquires a target

FILEList myKilledEnemyFeedback
- Name of sound file to play when unit kills an enemy

DECIMAL myFireRate
- The weapons rate of fire. The time interval in seconds between shots fired(from the same magazine)

NUMBER myMagazineCapacity
- The number of bullets held in each magazine.

DECIMAL myReloadTime
- The time in seconds it takes to load a new magazine

DECIMAL myShooterOffsetX
DECIMAL myShooterOffsetY
DECIMAL myShooterOffsetZ
- Distance to Turret rotation from main object pivotpoint in x,y,z

DECIMAL myBarrelOffsetX
DECIMAL myBarrelOffsetY
DECIMAL myBarrelOffsetZ
- Distance to Barrel rotation from turret pivotpoint in x,y,z

DECIMAL myFireOffsetX
DECIMAL myFireOffsetY
DECIMAL myFireOffsetZ
- Distance to Bullet spawnpoint from Barrel pivotpoint in x,y,z

NUMBER myUseMovementSpecificFireAnimationFlag
- Set this to 1 if the unit has a special animation for moving and firing at the same time(ex. The F1_Marine in Ground Control 2)
TEXT myHeadingAimingChild
- Name of objects Heading (Turret) child from showbox.
TEXT myPitchAimingChild
- Name of objects Pitch (Barrel) child from showbox.
DECIMAL myPitchOffset
- initial pitch offset of the Shooter, in degrees

ProjectileType myProjectile
{
 FILE myModelFile
 - Filename of bullet object mrb.
 DECIMAL myPostHitTimeToLive
 - How long bullet will live after hit in second.
 UnitType::EffectType mySolidHitEffect
 - See **“Definition of special classes used by the parasites”** below
 UnitType::EffectType myWaterHitEffect
 - Filename of bullet hit mrb on water.
 SoundSource mySoundSource
 - The sound of the projectile while moving
}

DECIMAL myProjectileSpeed
- The speed of the bullets in metres per second.

NUMBER myCanTargetAirFlag
- Determines if the unit can attack air units

NUMBER myCanTargetGroundFlag
- Determines if the unit can attack ground units

NUMBER myCanTargetInfantryFlag
- Determines if the unit can attack infantry units

ProjectileParasiteList myParasites
- See **“Definition of special classes used by the parasites”** below.

}

BallisticShooter

{

DECIMAL myMinRange
- Minimum firing range. Unit can not attack anything closer to itself than this

DECIMAL myMaxRange
- Maximum firing range. Unit can not attack anything farther away than this

TEXT myFireSound1
- Name of sound file to play when firing

TEXT myFireSound2
- Name of sound file to play when firing

TEXT myFireSound3
- Name of sound file to play when firing

DECIMAL myTargetSpreadRadius

- *Determines the radius of a circular area around the target within which the unit's shots are randomly scattered when firing. A smaller radius means shots are more likely to hit their target*

FILEList myAcquiredTargetFeedback

- *Name of sound file to play when unit acquires a target*

FILEList myKilledEnemyFeedback

- *Name of sound file to play when unit kills an enemy*

DECIMAL myFireRate

- *The weapons rate of fire. The time interval in seconds between shots fired(from the same magazine)*

NUMBER myMagazineCapacity

- *The number of bullets held in each magazine.*

DECIMAL myReloadTime

- *The time in seconds it takes to load a new magazine*

DECIMAL myShooterOffsetX

DECIMAL myShooterOffsetY

DECIMAL myShooterOffsetZ

- *Distance to Turret rotation from main object pivotpoint in x,y,z*

DECIMAL myBarrelOffsetX

DECIMAL myBarrelOffsetY

DECIMAL myBarrelOffsetZ

- *Distance to Barrel rotation from turret pivotpoint in x,y,z*

DECIMAL myFireOffsetX

DECIMAL myFireOffsetY

DECIMAL myFireOffsetZ

- *Distance to Bullet spawnpoint from Barrel pivotpoint in x,y,z*

NUMBER myUseMovementSpecificFireAnimationFlag

- *Set this to 1 if the unit has a special animation for moving and firing at the same time(ex. The F1_Marine in Ground Control 2)*

TEXT myHeadingAimingChild

- *Name of objects Heading (Turret) child from showbox.*

TEXT myPitchAimingChild

- *Name of objects Pitch (Barrel) child from showbox.*

DECIMAL myPitchOffset

- *initial pitch offset of the Shooter, in degrees*

ProjectileType myProjectile

{

FILE myModelFile

- *Filename of bullet object mrb.*

DECIMAL myPostHitTimeToLive

- *How long bullet will live after hit in second.*

UnitType::EffectType mySolidHitEffect

- *See "Definition of special classes used by the parasites" below*

UnitType::EffectType myWaterHitEffect

- *Filename of bullet hit mrb on water.*

SoundSource mySoundSource

- *The sound of the projectile while moving*

}

DECIMAL myProjectileSpeed

- *The speed of the bullets in metres per second.*

NUMBER myCanTargetAirFlag

- *Determines if the unit can attack air units*

NUMBER myCanTargetGroundFlag

- *Determines if the unit can attack ground units*

NUMBER myCanTargetInfantryFlag

- *Determines if the unit can attack infantry units*

ProjectileParasiteList myParasites

- See **“Definition of special classes used by the parasites”** below.

}

HomingShooter

{

DECIMAL myMinRange

- *Minimum firing range. Unit can not attack anything closer to itself than this*

DECIMAL myMaxRange

- *Maximum firing range. Unit can not attack anything farther away than this*

TEXT myFireSound1

- *Name of sound file to play when firing*

TEXT myFireSound2

- *Name of sound file to play when firing*

TEXT myFireSound3

- *Name of sound file to play when firing*

DECIMAL myTargetSpreadRadius

- *Determines the radius of a circular area around the target within which the unit's shots are randomly scattered when firing. A smaller radius means shots are more likely to hit their target*

FILEList myAcquiredTargetFeedback

- *Name of sound file to play when unit acquires a target*

FILEList myKilledEnemyFeedback

- *Name of sound file to play when unit kills an enemy*

DECIMAL myFireRate

- *The weapons rate of fire. The time interval in seconds between shots fired (from the same magazine)*

NUMBER myMagazineCapacity

- *The number of bullets held in each magazine.*

DECIMAL myReloadTime

- *The time in seconds it takes to load a new magazine*

DECIMAL myShooterOffsetX

DECIMAL myShooterOffsetY

DECIMAL myShooterOffsetZ

- *Distance to Turret rotation from main object pivotpoint in x,y,z*

DECIMAL myBarrelOffsetX

DECIMAL myBarrelOffsetY

DECIMAL myBarrelOffsetZ
- Distance to Barrel rotation from turret pivotpoint in x,y,z

DECIMAL myFireOffsetX
DECIMAL myFireOffsetY
DECIMAL myFireOffsetZ
- Distance to Bullet spawnpoint from Barrel pivotpoint in x,y,z

NUMBER myUseMovementSpecificFireAnimationFlag
- Set this to 1 if the unit has a special animation for moving and firing at the same time(ex. The F1_Marine in Ground Control 2)

TEXT myHeadingAimingChild
- Name of objects Heading (Turret) child from showbox.

TEXT myPitchAimingChild
- Name of objects Pitch (Barrel) child from showbox.

DECIMAL myPitchOffset
- initial pitch offset of the Shooter, in degrees

ProjectileType myProjectile

```
{  
    FILE myModelFile  
    - Filename of bullet object mrb.  
    DECIMAL myPostHitTimeToLive  
    - How long bullet will live after hit in second.  
    UnitType::EffectType mySolidHitEffect  
    - See “Definition of special classes used by the parasites” below  
    UnitType::EffectType myWaterHitEffect  
    - Filename of bullet hit mrb on water.  
    SoundSource mySoundSource  
    - The sound of the projectile while moving  
}
```

DECIMAL myProjectileSpeed
- The speed of the bullets in metres per second.

DECIMAL myTrackFactor
- Determines the tracking capability of the homing projectiles fired. A higher value means better tracking. A value of 10 is used for most units in Ground Control 2

NUMBER myCanTargetAirFlag
- Determines if the unit can attack air units

NUMBER myCanTargetGroundFlag
- Determines if the unit can attack ground units

NUMBER myCanTargetInfantryFlag
- Determines if the unit can attack infantry units

ProjectileParasiteList myParasites
- See **“Definition of special classes used by the parasites”** below.

}

Dropship

{

FILEList myIsMovingToBaseFeedback
- Points to a sound file to play when the Dropship starts to move back to base

FILEList myCantMoveToBaseFeedback
- Points to a sound file to play when the Dropship tries, but can't, move back to base

FILEList myIsMovingToLandingZoneFeedback
- Points to a sound file to play when the Dropship starts moving to your Landing Zone

FILEList myCantMoveToLandingZoneFeedback
- Points to a sound file to play when the Dropship tries, but can't, move to your Landing Zone

}

DropshipSpawner

- No Parameters

Containable

{

NUMBER myContainableType

- An identifier used to match Containers with Containables. A Container (ex. "F1_Light_APC" in Ground Control 2) of container type 1 can only carry Containables(ex. "F1_Marine" in Ground Control 2) of containable type 1

}

Container

{

NUMBER myCapacity

- The number of passenger the Container can carry, max 8

NUMBER myClaimableFlag

- A value of 1 means that the unit is claimable. A claimable unit

NUMBER myContainerType

- Matches Containers with Containables. See "**Containable**" above

NUMBER myHidePassengersFlag

- A value of 1 means that units being contained should not be visible. (ex. The "F1_Transport_Copter" in Ground Control 2 has this value set to 0, so units being airlifted by that unit are still visible in the 3D world)

}

EngineSoundSource

{

SoundSource myConstantSound

- The idle sound of the unit

SoundSource myMovingSound

- The movement sound of the unit

TEXT myStartMovingSound

- The sound played when the unit begins to move

TEXT myStopMovingSound

- The sound played when the unit stops moving

}

Armor

{

Armor::Type myType

- The armor type of the unit. See **DamageModel in Data Format Documentation**

DECIMAL myFrontArmor

DECIMAL mySideArmor

DECIMAL myBackArmor

DECIMAL myTopBottomArmor

- The armor factors (damage reduction) for the the unit. This is multiplied in to the DamageModel when calculating how much damage a unit will take. The factors can be changed to make a unit more vulnerable in the rear than in the front. A value of 1.0 equals the 100% of the damage reduction already specified in the **DamageModel** in Data Format Documentation

}

Seer

{

DECIMAL myViewRange

- Determines how far a unit can see in meters.

DECIMAL myEyeHeight

- The eye height of the unit, in meters above ground

DECIMAL myPerceptionRange

- The perception range of the unit. Anything within this range will be detected by the unit, regardless of view range, enemy unit stealth range etc.

}

TargetAcquirer

- No Parameters

Stealth

{

DECIMAL myRange

- Determines the units stealth range in meters. The unit will not be spotted by enemy units until he has travelled a distance equal to his own stealth range into the enemy unit's view range, except when the unit is within the enemy unit's perception range which overrides all view/stealth values

DECIMAL myForestRangeMultiplier

- When/if the unit is in forest, it's stealth value is multiplied by this factor. A value of 1.0 leaves the stealth unaltered, a value less than 1.0 gives the unit less stealth and makes him easier to detect. A value greater than 1.0 increases the unit's stealth making it harder to detect.

DECIMAL myBuildingRangeMultiplier

- When/if the unit is in a building, it's stealth value is multiplied by this factor. It works in the same way as the forest range multiplier above.

}

Meldable

- No Parameters

Melder

- No Parameters

Unmeldable

- No Parameters

Unmelder

- No Parameters

ZoneCapturer

- No Parameters

Regenerator

{

NUMBER myTimeIntervallnSeconds

- The time interval between each change to the unit's health

NUMBER myHealedDamagePerTimeInterval

- The amount of health to receive at each time interval

}

Blower

{

DamageType myDamageType

- The DamageType of the blast

NUMBER myDamage

- The amount of damage dealt by the blast

DECIMAL myBlastRadius

- The radius of the blast. This is the same kind of blast damage used by the projectiles

}

BlowerTrigger

{

DECIMAL myTrigRadius

- If a unit has both a **Blower** and a **BlowerTrigger**, this radius is used when the unit detonates to automatically detonate any other unit within the radius that also has a **Blower**, creating a chain effect.

}

Repairer

{

DECIMAL myRepairRadius

- This radius specifies how close the unit has to be to another unit in order to be able to heal it. If the unit is given a repair target outside the repair radius, the **Repairer** will tell the unit to move in closer.

NUMBER myTimeIntervalInSeconds

- The time interval in seconds between each health change the repairer does to its repair target

NUMBER myRepairedDamagePerTimeInterval

- The amount of health restored to the repair target each interval

}

AntiMissile

{

DECIMAL myDetectRadius

- The radius within which the AMS can detect incoming hostile projectiles (note that support weapons are no longer targeted by AMS, as of v1.0.0.8)

NUMBER myNumberOfShotsPerSecond

- The number of hostile projectiles the AMS is capable to destroy per second

}

AutoCloudSpawner

{

PP_Cloud myCloudType

- The type of cloud to spawn

DECIMAL mySpawnInterval

-The spawn interval. The **AutoCloudSpawner** will constantly re-spawn a cloud of the chosen type at the unit's current position with this interval. Take into consideration the

duration of the cloud. A new cloud can not be spawned inside an existing one. Also note that the clouds will remain at the position they were spawned and NOT follow as the unit moves away.

}

Resident

- No Parameters

CLASS EMPTY_ORDER_SLOT

- No Parameters

Definition of special classes used by the parasites

DamageType

*-The available damage types. The damage inflicted depends on the DamageType of the projectile and the Armor type of the unit being damaged. See the **DamageModel** in **Data Format Documentation** for more info.*

```
{
    ANTI_TANK
    MACHINE_GUN
    ANTI_PERSONNEL
    MISSILES
    FRAGMENTATION
    CHEMICAL
    ENERGY
    ALIEN
}
```

Armor::Type

-The available armor types

```
{
    BODY
    VEHICULAR
    REINFORCED_VEHICULAR
    ARMOR
    EXOTIC
    BUILDING
    IMMUNE
}
```

UnitType::EffectType

```
{
    UnitType::EffectType::ModelInfoList myModels
    - Model to use as effect

    TEXTList mySounds
    - Sound to play as effect

    UnitType::Decal myDecal
    - Decal to place as effect
}
```

ProjectileParasiteList

-This is a list of special parasites which determine what happens when the projectile impacts. Any combination of the following can be used:

```
{
    CLASS PP_DirectDamage
    {
```

```

        NUMBER myDamage
        - The amount of direct damage. Direct damage is dealt only to a unit
        hit directly by the projectile

        DamageType myDamageType
        - The DamageType.
    }

CLASS PP_BlastDamage
{
    NUMBER myDamage
    - The amount of blast damage. Blast Damage is dealt to all units
    within the blast radius.

    DamageType myDamageType
    - The DamageType

    DECIMAL myBlastRadius
    - The radius used to specify a spherical area within which the blast
    damage is dealt.
}

CLASS PP_UnitSpawner
{
    TEXT myUnitType
    - Name (takes the name of a UnitType in UnitTypes.Juice,
    ex."F1_Marine") of a unit to spawn at the point of impact. If the
    Impact point is not accessible to the specific unit, no unit will be
    spawned.
}

CLASS PP_Cloud
{
    DECIMAL myTimeToLive
    - The duration of the cloud in seconds

    NUMBER myBlockLOSFlag
    - Determines whether the cloud should block line of sight or not.

    NUMBER myHealthChange
    - The amount of change in health administered to units affected by
    the cloud. Positive numbers increase health, negative numbers deal
    damage of the type specified by myDamageType

    DECIMAL myHealthChangeInterval
    - The time interval in seconds between each health change. A value
    of 0.5 means that all units affected by the cloud will have it's health
    changed every half second

    DamageType myDamageType
    - The DamageType to use if health change is set to a negative
    number

    DECIMAL myDamageMultiplier
    - A factor by which to multiply the damage given, from any source, to
    a unit affected by the cloud

    NUMBER myAffectEnemyFlag
    - If this value is set to 1 the cloud will affect enemy units

```

NUMBER myAffectFriendlyFlag

- *If this value is set to 1 the cloud will affect friendly units(units of the same team and neutral units)*

DECIMAL myRadius

- *The radius of the cloud*

FILE myModelFile

- *name of the file that holds the visual representation of the cloud.
(remember to enter a radius that matches the visual representation)*

}

}