

# Star Trek: Armada ODF Directives

[ODF files](#) of the game *Star Trek: Armada* consist of lines of so-called directives, that influence, how an element of the game behaves, e.g. how much health a specific unit does have. Here is a rather lengthy list of directives, with their different uses and effects.

## Meta Commands

Some commands/directives are actually meta commands, as they don't have any effect on their own, but only import other files referenced, which in turn are causing the actual effects. They work with any kind of file, but don't always make sense, depending on the type of ODF/object they are used by.

Command/Directive	Effect/Usage
<code>#include "filename.odf"</code>	Makes the game include any directives that are not already included in the current file, from another file. Many weapons use this so that they can have a different name and sprite but have the same damage and fire rates etc... Simply put, it behaves as if all the text from the referenced file is placed at this location, instead of the <i>#include</i> command. This works cascading. So if an included file uses <i>#include</i> as well, these files are also de-referenced during loading. Do not create cyclic references!
<code>baseName = "my_file_without.sod"</code>	Specifies which <a href="#">SOD file</a> is used for this unit. This is only useful when the SOD is named differently than the unit's ODF file. When using this directive in an ODF, this ODF inherits all the GUI properties of the ODF designated by the <i>baseName</i> directive as well as the SOD (i.e. button, wireframe).
<code>physicsFile = "my_file.odf"</code>	Specifies which physics file is used for this ship.

## Type Keys

Key	Meaning
C	directive for commands
M	map object directive
O	other miscellaneous directive
P	physics directive
SH	ship directive
ST	station directive
SW	special weapon directive
W	weapons directive

## Ship and Station Directives

Directive	Description	Type
aiName = <i>"CraftProcess"</i>	This is an AI parameter that helps the AI evaluate targets. It tells the AI what type of ship it is so it can prioritize the targets and function correctly when told to seek and destroy etc... Allowed string values are:\\CraftProcess\\SalvageProcess\\StarbaseProcess	SH, ST
ambientSound = <i>"omega.wav"</i>	This defines the file to be used for the sound a map object makes. The volume changes depending on the proximity to the object.	O
animation = 1	Turns the animation in a SOD file on when the unit is constructed.\\1 means play once then stop.\\2 means play looping infinitely.	SH, ST
attackPower = 0.5	This is used by the AI to evaluate how strong the unit is. The higher the number that follows is, the more powerful the AI considers it and the more ships it will send against it when fighting it.	SH, ST
avoidanceClass = 555	How much should we try to avoid this ship? The higher the number the more space we assume is around it that we want to avoid.	SH, ST
boardingPartyStrength = 12.5	The strength of boarding parties when they transport aboard an enemy ship. The higher the number the more powerful the party is and the more likely they are to overcome the target ships crew. The values of both races, attacker and attacked, will influence how much crew the attacked will lose for each crew beamed over to it. Example: Attacker has 3, attacked has 2. Each beam over of 5 crewmen (standard beam-in party size) costs the attacker 5 crewmen on his own ship, but will kill 7.5 on the attacked ship. So in order to take over a 100 crewmen target ship in this constellation, an investment of 70 crewmen will be required, ending up in 5 own crewmen remaining on the target ship, eventually.	SH
buildAnimation = 1	y/n value, are there any build functions at this station?	SH, ST
buildHardpoint = <i>"hp01"</i>	Defines the hardpoint at which building takes place at this station or unit.	SH, ST
buildItem0 = <i>"odf_name"</i>	This tells the game that whatever ODF name follows this statement can be built from this unit or station, the 0 can change into any number, depending on how many items are on the list.	ST
buildTime = 123	The amount of time in seconds it takes to build this unit.	SH, ST
canAttack = 0	This y/n value tells the game whether or not this ship can engage in combat. By default ships like construction ships cannot attack, with a couple exceptions. It determines if the button for an action related to this is present when you click on a unit with this line in it.	SH, ST
crewCost = 50	The amount of crew required for the ship or station, to be built. The amount is removed from the players stock, the moment the construction starts.	SH, ST
crewHitPercent = 2.5f	This determines the chances that the crew will be hit by a weapon, once the shields are completely down. Must add up to 100 (%) along with the following other values: <i>hullHitPercent</i> , <i>lifeSupportHitPercent</i> , <i>sensorsHitPercent</i> , <i>shieldGeneratorHitPercent</i> and <i>weaponsHitPercent</i> .	SH, ST
criticalTargetHardpoints = <i>"hp12"</i>	This designates the hardpoint for critical hits. After this directive a series of hardpoint names is given, that are the firing points for weapons. If the critical hit hardpoint has no hitpoints left, the ship or station is destroyed.	SH, ST

Directive	Description	Type
<code>damagedScan = 150f</code>	When the sensors are destroyed/deactivated, how far does the ship scan, or „see“?	
<code>dilithiumCost = 1000</code>	The amount of dilithium required for the ship or station to be built. The amount is removed from the players stock, the moment the construction starts.	SH, ST
<code>dilithiumTransferRate = 15.0</code>	Sets the rate at which the resources are transferred between the freighter and the resource station.	SH, ST
<code>distanceBelowGrid = 30</code>	How far below the grid should we move this station?	ST
<code>enginesCrewLoss = 5.0f</code>	This is the percentage of how much of the crew dies, the moment the engines sub-system gets destroyed.	SH, ST
<code>enginesHitPoints = 12</code>	Number of hitpoints the engine sub-system has. Once damage exceeds this number it becomes disabled.	SH, ST
<code>enginesRepairTime = 1.0f</code>	The amount of time in seconds that it takes to repair one hitpoint of damage to the engines sub-system. This is modified by the crew status. If the crew status is yellow, it usually is slower. In status red it is usually even slower. But it can also be totally different (e.g. Borg always repair at the same rate).	SH, ST
<code>enginesTargetHardpoints = "hp12"</code>	This designates the hardpoint for the engine sub-system. After this directive a series of hardpoint names is given, that are targets for weapons.	SH, ST
<code>fireball = "xfirebal"</code>	This specifies the explosion fireball that the ship uses when it blows up.	SH, ST
<code>footprintBuffer = 20.0f</code>	Sets the footprint that the unit leaves when placing it for construction.	ST
<code>freighterName = "kfreight"</code>	Defines what freighter is used as a resource gatherer for a refinery. This ship is automatically built along with the mining station.	ST
<code>hullHitPercent = 62.0f</code>	This determines the chances that the hull will be hit by a weapon, once the shields are completely down. Must add up to 100 (%) along with the following other values: <i>crewHitPercent</i> , <i>lifeSupportHitPercent</i> , <i>sensorsHitPercent</i> , <i>shieldGeneratorHitPercent</i> and <i>weaponsHitPercent</i> .	SH, ST
<code>hullTargetHardpoints = "hp12"</code>	This designates the hardpoint for the hull sub-system. After this directive a series of hardpoint names is given, that are the firing points for weapons. If the hull hardpoint's hitpoints are depleted, the ship or station gets destroyed.	SH, ST
<code>intrinsicValue = 1.5</code>	This tells the AI how important of a target this unit is. The higher the number the more likely the AI is to attack it.	SH, ST
<code>is_starbase = 0</code>	Is it a starbase? y/n value.	ST
<code>isHero = 1</code>	Specifies that this ship is a special hero unit, y/n value.	SH, ST
<code>isTranswarpGate = 1</code>	Specifies that this station is a Transwarp Gate, y/n value.	ST
<code>lifeSupportCrewLoss = 3.0f</code>	This is the percentage of how much of the crew dies, the moment the life support sub-system gets destroyed. The only exception is life support. Life support goes by a percent of the crew that dies for every second that the life support is disabled. See also <i>lifeSupportLoss</i> .	SH, ST
<code>lifeSupportHitPercent = 2.5f</code>	This determines the chances that the life support sub-system will be hit by a weapon, once the shields are completely down. Must add up to 100 (%) along with the following other values: <i>crewHitPercent</i> , <i>hullHitPercent</i> , <i>sensorsHitPercent</i> , <i>shieldGeneratorHitPercent</i> and <i>weaponsHitPercent</i> .	SH, ST
<code>lifeSupportHitPoints = 12</code>	Number of hitpoints this system has, once damage exceeds this number it becomes disabled.	SH, ST

Directive	Description	Type
lifeSupportLoss = 2.0f	This is the percentage of how much of the crew dies every second, while the life support sub-system is inoperable. Important: If the percentage x residual crew amount drops below 1, it is always 1 crewman per second. So a ship without life support can totally „bleed out“.	SH, ST
lifeSupportRepairTime = 0.1f	The amount of time in seconds that it takes to repair one hitpoint of damage to the life support sub-system. This is modified by the crew status. If the crew status is yellow, it usually is slower. In status red it is usually even slower. But it can also be totally different (e.g. Borg always repair at the same rate).	SH, ST
lifeSupportTargetHardpoints = "hp12"	This designates the hardpoint for the life support sub-system. After this directive a series of hardpoint names is given, that are the firing points for weapons. If the life support hardpoint's hitpoints are depleted, the ship or station will lose crew over time. This progresses faster, the more crew is actually still aboard.	SH, ST
mapIcon = "map_worm"	This specifies what icon to use for the minimap to display a planet.	O
maxHealth = 100	Max Shield Strength	SH, ST
maximumUpgrades = 6	Defines the max number of officer upgrades the station can build.	ST
maxRoll = 12	This defines the roll of a craft. Roll is side to side tilting when turning, also called „banking“. Low values makes the ship always fly flat and level even when turning.	SH
maxSpecialEnergy = 123	Specifies the total amount of special energy this unit can have when fully charged.	SH, ST
moveBelowCombatArea = 0	Should we move this station below the grid. This is a y/n value.	ST
numberOfWorkerBees = 5	Defines how many bees are used to build it.	SH
officerCost = 10	The amount of officers required for the ship or station to be built. The amount is removed from the players stock, the moment the construction starts. Specialty here: If the ship or station gets destroyed, the used up officers will get freed up, too. So it's more like reserving the cost, than subtracting it from the stock.	SH, ST
officerGain = 20	Tells how many officers are gained when this upgrade is built.	ST
possibleCraftNames = "name_1" "name_2"	Lists all the possible names for the craft or station. and each one is surrounded by quotes and separated with a space	SH, ST
podHardpoints = "hp01"	Defines the hardpoint at which pod building takes place at this station or unit.	SH, ST
race = "borg"	Names the race that can build the station or ship	SH, ST
rangeScan = 123	Defines how far this unit can scan (line of sight).	SH, ST
repairFacility = 0	Specifies whether this station is a repair facility, y/n value.	SH, ST
repairRed = 0.5	A modifier that controls how fast the ship repairs when the crew status is red.	SH, ST
repairYellow = 0.5	A modifier that controls how fast the ship repairs when the crew status is yellow.	SH, ST
ScaleSOD = 1.1	Multiplier, resizes the SOD to a smaller or larger size.	SH, ST

Directive	Description	Type
<code>sensorsCrewLoss = 2.0f</code>	This is the percentage of how much of the crew dies, the moment the sensors sub-system gets destroyed.	SH, ST
<code>sensorsHitPercent = 13.0f</code>	This determines the chances that the sensors sub-system will be hit by a weapon, once the shields are completely down. Must add up to 100 (%) along with the following other values: <i>crewHitPercent</i> , <i>hullHitPercent</i> , <i>lifeSupportHitPercent</i> , <i>shieldGeneratorHitPercent</i> and <i>weaponsHitPercent</i> .	SH, ST
<code>sensorsHitPoints = 12</code>	Number of hitpoints this system has, once damage exceeds this number it becomes disabled.	SH, ST
<code>sensorsRepairTime = 3.0f</code>	The amount of time in seconds that it takes to repair one hitpoint of damage to the sensors sub-system. This is modified by the crew status. If the crew status is yellow, it usually is slower. In status red it is usually even slower. But it can also be totally different (e.g. Borg always repair at the same rate).	SH, ST
<code>sensorsTargetHardpoints = "hp12"</code>	This designates the hardpoint for the sensor sub-system. After this directive a series of hardpoint names is given, that are the firing points for weapons. If the sensor hardpoint's hitpoints are depleted, the ship or station loses it's longer range scanning ability (line of sight), so only the very-short range ling of sight remains.	SH, ST
<code>shieldDelay = 3.0f</code>	Delay before shields begin recharging once it has reached zero in seconds	SH, ST
<code>shieldGeneratorCrewLoss = 2.0f</code>	This is the percentage of how much of the crew dies, the moment the shield generator sub-system gets destroyed.	SH, ST
<code>shieldGeneratorHitPercent = 8.0f</code>	This determines the chances that the shield generator sub-system will be hit by a weapon, once the shields are completely down. Must add up to 100 (%) along with the following other values: <i>crewHitPercent</i> , <i>hullHitPercent</i> , <i>lifeSupportHitPercent</i> , <i>sensorsHitPercent</i> , <i>shieldGeneratorHitPercent</i> and <i>weaponsHitPercent</i> .	SH, ST
<code>shieldGeneratorHitPoints = 12</code>	Number of hitpoints this system has, once damage exceeds this number it becomes disabled.	SH, ST
<code>shieldGeneratorRepairTime = 2.0f</code>	The amount of time in seconds that it takes to repair one hitpoint of damage to the shield generator sub-system. This is modified by the crew status. If the crew status is yellow, it usually is slower. In status red it is usually even slower. But it can also be totally different (e.g. Borg always repair at the same rate).	SH, ST
<code>shieldGeneratorTargetHardpoints = "hp12"</code>	This designates the hardpoint for the shield generator sub-system. After this directive a series of hardpoint names is given, that are the firing points for weapons. If the shield generator hardpoint's hitpoints are depleted, the ship or station loses it's ability to replenish its shield energy. This does not mean, it cannot have any shields any more (some special weapons only replenish shield energy, while the generator may very well be still offline).	SH, ST
<code>shieldRate = 3.5</code>	Number of points regained by the shields roughly, per second.	SH, ST
<code>specialEnergyRate = 12</code>	Specifies the recharge rate of the special energy in units per second.	SH, ST
<code>tooltip = "You text goes here"</code>	Specifies the short tooltip that appears when the mouse is passed over the unit.	SH, ST, SW, W
<code>unitName = "myunit"</code>	Specifies the unit name.	SH, ST

Directive	Description	Type
<code>verboseTooltip = "Some longer text"</code>	The extended tooltip that shows up when the unit is selected or the mouse is held over the unit for an extended time.	SH, ST, W, SW
<code>weapon1 = "name_without_odf"</code>	Specifies the name of the weapon ODF file the number 1 shows that it is the first weapon equipped on this ship, that number changes with each additional weapon added to the ship.	SH, ST
<code>weaponHardpoints1 = "hp1" "hp3"</code>	Designates the hardpoints that the weapon uses number 1 shows that it is the first weapon equipped on this ship, that number changes with each additional weapon added to the ship. After this directive is a series of hardpoints that are the firing points for the weapon. They are noted in format hp $xx$ where $xx$ is a number. Each one is surrounded by "" quotes and separated with a space.	SH, ST
<code>weaponRed = 0.5</code>	A multiplier that controls how fast the ship fires when the crew status is red.	SH, ST
<code>weaponsCrewLoss = 2.0f</code>	This is the percentage of how much of the crew dies, the moment the weapons sub-system gets destroyed.	SH, ST
<code>weaponsHitPercent = 2.0f</code>	This determines the chances that the weapons sub-system will be hit by a weapon, once the shields are completely down. Must add up to 100 (%) along with the following other values: <i>crewHitPercent</i> , <i>hullHitPercent</i> , <i>lifeSupportHitPercent</i> , <i>sensorsHitPercent</i> and <i>shieldGeneratorHitPercent</i> .	SH, ST
<code>weaponsHitPoints = 12</code>	Number of hitpoints this system has, once damage exceeds this number it becomes disabled.	SH, ST
<code>weaponsRepairTime = 2.0f</code>	The amount of time in seconds that it takes to repair one hitpoint of damage to the weapons sub-system. This is modified by the crew status. If the crew status is yellow, it usually is slower. In status red it is usually even slower. But it can also be totally different (e.g. Borg always repair at the same rate).	SH, ST
<code>weaponsTargetHardpoints = "hp12"</code>	This designates the hardpoints for the weapons sub-system. After this directive a series of hardpoint names is given, that are the firing points for weapons. If the weapon hardpoint's hitpoints are depleted, the ship or station loses it's ability to fire any weapon, including special weapons.	SH, ST
<code>weaponYellow = 0.5</code>	A modifier that controls how fast the ship fires when the crew status is yellow.	SH, ST
<code>weldingBeamSprite = "weldbeam"</code>	The sprite used for the workers when this ship is built.	SH, ST
<code>weldingBeamWidth = 12</code>	Sets the width of the beam that the worker bees use when constructing this vessel	SH, ST
<code>workerBeeHardpoints = "hp01"</code>	Defines the hardpoint at which worker bee launching takes place at this station or unit.	SH, ST
<code>workerBeeName = "my_name"</code>	Specifies the name of the small worker bee SOD that is used during construction of this unit.	SH, ST

## Weapons Directives

Directive	Description
<code>blastRadius = 150.0f</code>	This is the radius of the explosion that the weapon creates. This is the effect radius not the size of the sprite.
<code>color = 1.0 0.3 1.0</code>	This RGB value modifies the color of the texture definition.

Directive	Description
<code>explosionSprite = "sparkball1"</code>	This specifies the explosion that is used for this ordinance when it impacts the target.
<code>explosionSpriteRadius = 50.f</code>	This is the radius of the sprite for the above explosion. It doesn't have to be the same size as the blast radius.
<code>explSound = "nell.wav"</code>	The name of the sound played when the ordinance explodes.
<code>fadeFactor = 0.9</code>	A percentage fraction that controls how long before the shockwave or other graphical effect starts to fade out. If this is too high the shockwave will be too big and will appear to be much larger than it's actual effect radius.
<code>fireSound = "my_file.wav"</code>	The sound that is played when the weapon fires.
<code>hitChance = 1.0</code> <code>"shipname.odf" 0.75</code>	This percentage fraction is the chance that the weapon will hit the target. This can be applied to specific ships by adding a line under it in this format: <code>"shipname.odf" x</code> where <i>x</i> is the number between 0 and 1
<code>hullCrewModifier = 0.1</code>	This percentage fraction is a modifier that controls what percent of the damage which hits the hull while exceeding the damage threshold is used to calculate the percentage of the crew, that dies with each hit. This only applies to damage that is in excess of the damage threshold. This can be applied to specific ships by adding a line under it in this format: <code>"shipname.odf" x</code> where <i>x</i> is the number between 0 and 1.
<code>Length = 10.0</code>	This is the length of the beam used for this weapon. It is usually used for Pulse Phasers along with the <i>radius</i> directive to designate the width and length of the pulse.
<code>lightFalloffRange = 5.0</code>	This controls the way the color faded in and out by setting the range of the „falloff“.
<code>lightFalloffStart = 50.0</code>	This controls control the way the color faded in and out by setting the start of the „falloff“.
<code>lightColor = 0.0 1.0 0.3</code>	This RGB value controls the way the weapon lighting works. This controls the actual color of a light.
<code>maxShots = 3</code>	How many shots should we save up? This is the number of shots that will be released the first time the weapon fires. <b>Note:</b> This only works on units with <i>saveFire</i> set to <i>yes</i> .
<code>minimumShotInterval = 0.2</code>	This is the least amount of time between the saved up shots. This only works the first time the weapon is fired. After that the weapon fires normally until it has a period of time not firing saving up shots again. This only works with units who have <i>saveFire</i> set to <i>yes</i> .
<code>needTarget = 1</code>	This y/n value defines whether the weapon does need manual targeting. This is mostly used for special weapons but is also used for artillery Photons. For the latter do not use it for Armada 1. This will have strange side-effects in the form of „sticking“ photons on the hull of the target, instead of going off.
<code>objectName = "muball"</code>	This designates an another ODF that has the specifications for another effect that this weapon has. Could be an artillery wave or an object like an ion storm nebula or any of several other effect types.
<code>omegaTurn = 3.0</code>	This controls the turning radius of the ordinance. The lower the number the faster the weapon turns to chase the target.

Directive	Description
<code>ordName = "odf_name"</code>	This specifies the file that is used for the ordinance for this weapon. For example, a gun is a weapon. However a bullet is the ordinance that guns may use. Without some ordinance to do the actual damage the weapon is useless.
<code>Radius = 18</code>	This is the radius or „thickness“ of the sprite. It is used for Phasers, Photons and various other weapons to control the size of the sprite.
<code>radiusFactor = 10.0</code>	A multiplier, that is a multiple of the ship's radius and is used to get the radius of the shockwave.
<code>range = 400.0f</code>	The maximum distance at which this weapon fires. This weapon will fire when an enemy comes within this range of the ship with this weapon.
<code>rate = 150</code>	This controls how fast a wave expands. The higher the number the faster the wave moves and grows.
<code>rays = 16</code>	The number of „rays“ or sections on a shockwave. The higher it is the more detailed the wave will look but also the more heavy the load on the game engine will be.
<code>restrictFireArc = 1</code>	This tells the game to restricts the arc of fire of the weapon. This is a yes/no question, 0 = no, 1 = yes.
<code>saveFire = 1</code>	Should we save up our fire so we can fire several rapid bursts once we get a target? y/n value. <b>Note:</b> This only works on units with the classLabel <i>cannon</i> .
<code>seekTime = 0.25</code>	This is the amount of time in seconds the ordinance obeys the omega turn limit before it starts to head for the target. It is mostly used for Photons, since they are fired from a location on the ship and then start to head toward the target after they clear the immediate area of the ship.
<code>shotAccel = 100.0</code>	This is the speed at which the ordinance accelerates once fired. The higher the number the faster it accelerates.
<code>shotColor = 227</code>	This is a useless directive that seems to do nothing at all. I believe it was originally intended to change the color of the Phaser beams so that they could use one sprite for the Phasers and change the color through their ODFs. But it seems to never have been implemented. Do not remove it but also don't bother changing it.
<code>shotDelay = 2</code>	This is the delay between shots in seconds. It controls the firing rate of the weapon. It is a number and can be virtually any number. The higher the number the slower the firing rate.
<code>spriteDuration = 1.0</code>	This is the length of time it takes to play through one cycle of the sprite's animation. Then it loops back to the beginning.
<code>stopToHitModifier = 0.15</code>	This multiplier modifies the hitChance when the firing ship is stationary. This can be a positive or a negative number. You can add a - in front to represent a negative percentage which would decrease the chance of hitting.
<code>targetLocation = 1</code>	This y/n value allows the weapon to target a location on the map instead of having to target a specific unit or station. This is mostly used for special weapons but is can also used for artillery Photons.
<code>texture = "wselfdes"</code>	The name of the texture used for this item, usually used for shockwaves.

Directive	Description
toHitStopModifier = 0.5	This multiplier modifies the <i>hitChance</i> when the target is stationary. This can be a positive or a negative number. You can add a - in front to represent a negative percentage which would decrease the chance of hitting.
wpnName = "Weapon's Name"	This is the weapon's name to be displayed.

[ [Modding](#) ] [ [Tools](#) ] [ [ODF Files](#) ] [ [ODF Directives](#) ] [ [Class Labels](#) ] [ [Tech Tree Files](#) ] [ [SOD Files](#) ] [ [Buttons](#) ] [ [Wire Frames](#) ] [ [Sprites](#) ] [ [AI Scripts](#) ] [ [Model Hierarchy](#) ] [ [Node Names](#) ] [ [Emitter Names](#) ] [ [Texture Animation Names](#) ] [ [Sprite Names](#) ]

[ [Back to Modding](#) ]

From:

<https://mwohlauer.d-n-s.name/wiki/> - [mwohlauer.d-n-s.name](https://mwohlauer.d-n-s.name/) / [www.mobile-infanterie.de](https://www.mobile-infanterie.de)

Permanent link:

[https://mwohlauer.d-n-s.name/wiki/doku.php?id=en:games:star\\_trek\\_armada\\_1:modding:odf\\_directives&rev=1701379339](https://mwohlauer.d-n-s.name/wiki/doku.php?id=en:games:star_trek_armada_1:modding:odf_directives&rev=1701379339)

Last update: **2023-11-30-21-22**

