Star Trek: Armada ODF Files

Object **D**efinition **F**iles, or short ODF files, are script-like text files, that control many aspects of Units in *Star Trek: Armada*. The file name extension is *.odf*. (This is why it is not redundant to speak of ODF files, while the *F* already stands for *file*.) They contain ship, station, weapon and ordinance definitions, which in turn make up how these game elements behave. This does however not imply the actual availability of an element defined by that ODF, as the Tech Tree File controls these aspects of the game.

In general the content of an ODF file consists of comments, introduced with two slashes (`//`), `#includes`, that place other ODF file contents at a certain location, as well as actual directives, that shape the definition the ODF is all about. ODFs are written in ASCII format with Windows Line Breaks (0x0d 0x0a). As such they can be edited by almost any text editor without special preparation.

Example ODF

A typical ODF file content could look like this:

(click to expand)

```
#include "turret.odf"

unitName = "Rejector"
tooltip = "Rejector (^J)"
verboseTooltip = "Powerful automated defense platform."

//Race which can build the station & starting race of station
race = "borg"

//Amount of time required to build station
buildTime = 620.0

//Number of officers required to build this station
officerCost = 50

//Dilithium Cost to build
dilithiumCost = 6200

//Max Shield Strength & initial shield strength
maxHealth = 3000

//Rate at which shield recharges
shieldRate = 12.5

//****************************************************************
//AI system parameters
//These parameters influence how the AI evaluates and compares craft.
```

```
//An abstract number that represents how much extra strength must be
//used to attack this craft due to its weapons.
// (0.0=no weapons, 0.5 = average weapons, 1.0=many good weapons)
attackPower = 1.00f

//The normalized intrinsic value of a target.  High value craft make good
//attack targets. (0.0=lowest value, 1.0=highest value)
intrinsicValue = 0.75f
//****************************************************************



//****************************************************************
// Very robust weapons system. We need that to be sturdy, to keep the
// regeneration available for as long as possible.
weaponsHitPoints = 100
//****************************************************************



//****************************************************************
//ART PARAMETERS & WEAPON NAMES
weapon1 = "bfpulse"
weaponHardpoints1 = "hp01" "hp02" "hp03"

// Long Range Pulse Cannon, 16 dps
weapon2 = "blrpulse"
weaponHardpoints2 = "hp04"

// Free Regeneration, add this explicitly in the tech tree
weapon3 = "freregen"
weaponHardpoints3 = "hp04"

// Free Tachyon Detection Grid, add this explicitly in the tech tree
weapon4 = "freetach"
weaponHardpoints4 = "hp04"

// Scanning range just as far as the Long Range Pulse Cannon and artillery
can
// shoot and T1 sensor stations can see
rangeScan = 1000

maxSpecialEnergy = 1000
specialEnergyRate = 10

// Hardpoints to hit for various systems and other locations.
// We don't have those on an automated defense platform:
// enginesTargetHardpoints = "hp10" "hp11" "hp12" "hp13"
// lifeSupportTargetHardpoints = "hp10" "hp11" "hp12" "hp13"

// These on the other hand, we do need:
weaponsTargetHardpoints =         "hp10" "hp11" "hp12" "hp13"
shieldGeneratorTargetHardpoints = "hp10" "hp11" "hp12" "hp13"
```

```
sensorsTargetHardpoints =          "hp10" "hp11" "hp12" "hp13"
hullTargetHardpoints =             "hp10" "hp11" "hp12" "hp13"
criticalTargetHardpoints =         "hp10" "hp11" "hp12" "hp13"

//********************************************************************
//MESSAGE & SOUND PARAMETERS
//********************************************************************

eventSelect = "PhaserTurretSelect"
eventAcknowledge = "PhaserTurretAcknowledge"
eventAttack = "PhaserTurretAttack"
eventDecommission = "PhaserTurretDecommission"

//********************************************************************
//KEYMAP LABEL
keymapLabel = "phaser_turret"
```

Location of ODF Files

For *Star Trek: Armada* only a handful of ODF files are presented in a plain fashion, as part of the patches released for the game, namely

- *borg.odf*,
- *chainp.odf*,
- *fed.odf*,
- *kling.odf*,
- *rom.odf*.

The rest are actually also provided along with the game, but in a packed form inside the file *assets.zfs*. ODF files placed in the folder *addon* effectively replace the files from inside *assets.zfs* (like the mentioned five files do, altering minor aspects of the factions and the Chain Reaction Pulsar as part of the 1.2 patch).

ODF Directives

The directives used in ODF files are numerous and as such described here: ODF Directives. The most important ones are probably the following:

Directive	Effect
`buildTime = 12.0`	The time in seconds it takes for this unit to finish construction.
`officerCost = 7`	The amount of supply this unit requires for construction.
`crewCost = 80`	The amount of crew this unit requires for construction. **Note**: This may be different from the eventual amount of crew ending up on a Mining Station as the implicitly along built freighter also deduces its amount of crew from the overall value.
`dilithiumCost = 140`	The amount of Dilithium required for this unit to be built.

Directive	Effect
`maxHealth = 100`	The starting and maximum shield strength of the unit.
`shieldRate = 0.75`	Rate at which the shields are replenished (as long as the shield generator sub-system is operational) in shield energy units per second.
`shieldGeneratorHitPoints = 20`	Hit points of the shield generator sub-system, basically the health of this sub-system.
`enginesHitPoints = 20`	Hit points of the engines sub-system, basically the health of this sub-system.
`weaponsHitPoints = 20`	Hit points of the weapons control sub-system, basically the health of this sub-system.
`lifeSupportHitPoints = 20`	Hit points of the life support sub-system, basically the health of this sub-system.
`sensorsHitPoints = 20`	Hit points of the sensor sub-system, basically the health of this sub-system.
`weaponx = "ODF Name"`	A weapons slot assignment, numbered in steps of 1 for *x*, e.g. *weapon1* or *weapon7* and defined with the ODF name without extension, e.g. `weapon1 = "fc1phas"`. The numbering must not have any gaps.
`weaponHardpointsx = "hpyy" "hpyy" "hpyy" "hpyy" "hpyy"`	Hardpoint definition for weapon number *x*. It is assigned a list of node names of the weapon hard points to be used, that are existing in the SOD model file, e.g. `weaponHardpoints1 = "hp01" "hp02" "hp03" "hp04" "hp05"`.
`race = "borg"`	Definition of the faction this unit usually is built by.
`buildItemx = "<ODF name>"`	This defines the item this unit can construct. The *x* variable is an index starting with 0, that defines which button slot is to be used for the construction button of this construction item. The *ODF name* is the file name of the ODF without *.odf* extension.

[**Modding**] [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 / www.mobile-infanterie.de**

Permanent link:
https://mwohlauer.d-n-s.name/wiki/doku.php?id=en:games:star_trek_armada_1:modding:odf_files

Last update: **2023-11-30-21-20**