Armada 1 Damage Rate Calculation

In different places of this Wiki you will find values given for the damage rate of ships and stations. Depending on the situation, these can be given as a range or as an absolute value. The game itself never tells the player any damage rates, but of course it basis its game mechanics on some sort of values. Here is a description about how to make the determination of damage rate of a particular station or ship.

Source of the Information

As was said above, the player has not easy access to the values behind a station's or ship's damage rate. The actual source is the definition in the ODF files of that ship or station. It is important to use **all** influencing ODFs for any calculations. Values are often inherited from other ODFs and in the process also overridden, if the inheritance hierarchy has more than one definition for certain aspects of a weapon.

Factors Influencing the Value

There are a number of factors that have an influence of the over-all damage rate of a ship or station. Those are:

Aspect	Relevance
Type of Weapon	Even if certain weapons may look like other weapons (e.g. pulse phasers) it does not necessarily mean they deal the same amount of damage. The weapon type is crucial for any damage rate considerations. This, together with the Number of Weapons Used defines how much damage the unit deals.
Number of Weapons Used	Each unit can have a multitude of weapons, which is actually not a rare thing. E.g. most battle ships have Phasers **and** Torpedoes at their disposal. Some ships also have more than one weapon of the same kind, e.g. scouts and destroyers, have two pulse weapons (except the Interceptor).
Base Damage	Each weapon has a base damage value. That does not necessarily mean, that exactly this value is applied, if the weapon hits. But it is a base for other effects to apply to.
Damage Variance	The Base Damage can be varied, meaning it's value can be increased or decreased randomly with each shot
Firing Rate	Even if the Base Damage and Variance are known, it makes of course a difference with what rate the weapon is actually able to fire. The faster the recharge, the higher the damage rate of course.
Hit Probability	Some weapons hit with 100% certainty if you are within weapons range. Others do not. This has an effect on the average damage rate.
Unit Type Dependency	Some weapons make use of a distinction in terms of Hit Probability for different ships or stations. E.g. scouts are often more likely to be missed than others while battle ships are less likely to be missed by a weapon.

Weapon Types

The game distinguishes between conventional weapons and Special Weapons. For all considerations the special weapons should not be taken into account. The complexity added by them, especially when considering whether a specific weapon was already available/researched or not makes it bad practice to factor them into the calculations.

But even among conventional weapons, there is a distinction between phasers, torpedoes and pulse phasers. While phasers are always defined to hit 100% of time (otherwise an extremely large phaser beam would point across the map, which the game designers tried to avoid) torpedoes and pulses phasers may miss. So the average damage rate may be lower than the nominal damage the weapon deals, when it hits.

But even within the same weapons type (e.g. pulse phasers) the look and sound may be same, but there may still be different actual weapons firing. E.g. the scout pulse phasers are a lot weaker than those of pulse defense stations.

The weapon type attributes are stored in their ODF file resp. those that it imports. The name of the weapon is used by the units ODF file.

Weapon Composition

This is a bit of a no-brainer. If a unit is stacked with weapons, e.g. multiple kinds of phasers and torpedoes, it deals the accumulating damage rate of those weapons. If it only has one weapon, it will deal the damage rate of that particular weapon. So the more weapons or the harder these weapons hit individually, the higher the damage rate. It is however, necessary to look at each weapon individually, if you do not just want to state an average damage rate, but the damage rate range. When looking at the ranges, the lower limit of damage rates have to be summed up, separately of the upper limit. For averages of course the averages have to be summed.

The weapon composition is stored in the ODF file of the unit. For each weapon actually used there will be a directive that ties that particular weapon to hard points. Note, that one weapon can be tied to multiple hard points, but that does not multiply it existence. Only one hard point will be used for the weapon at a time, depending on which one is the closest to the target aimed at.

Base Damage and Damage Variance

The base damage of a weapon is the amount of damage it deals, whenever it hits. Together with the Firing Rate that could already determine the damage rate. But some weapons do not simply fire and deal a set amount of damage when hitting, but a variance of their base damage. This is where statistics begin to influence the results. E.g. a phaser may deal 24 damage with a variance of +/- 4. That means, the lowest amount of damage dealt will be 20, the highest amount will be 28.

This already shows why there are differentiations possible, in upper and lower damage rates or average damage rates. As was said above in Weapon Composition, any eventual values will have to take all weapons into consideration, that are not special weapons. It is therefore important to know all the lowest and highest damage values a hit will deal. You can use the lowest and highest values separate. The variance has no skewness, so for the average damage per hit of that weapon the

calculation is simply the base damage.

Firing Rate

Again a no-brainer. If a weapon takes 3 seconds to recharge, even with the same base damage value it will deal double the damage rate than another one with 6 second recharge time.

Hit Probability and Unit Type Dependency

The Damage Variance is not the only aspect that introduces some level or randomness to damage rates. While phasers will always be designed in a way to hit 100% of times, the torpedoes and pulse phasers may miss. Depending on the probability of hitting the target, that can reduce the average damage rate. E.g. when the weapon has an average damage rate of 24/s with 100% hit probability, another weapon with the same damage rate but 20% probability to miss will have 20% less **average** damage rate (19.2/s). While each shot may have a strict 24 damage, fired once a second, one of 5 of them will miss (statistically). Meaning (on average!) the weapon will deal 96 damage with those 5 shots, 4 hits one miss. 96 damage in 5 seconds is said 19.2/s damage.

Always keep in mind that this is a statistical average! If e.g. a particular scout unit is really lucky, it might in reality get fired upon 5 times in a row and still be missed all 5 times. Of course the damage will then also amount to 0, with an actual damage rate of 0/s. (Although a priori that is highly unlikely with 80% hit probability: It equals to a tiny 0.032 % chance, that 5 of 5 shots miss.) But for comparisons that is just alright. 19.2/s is the value to use, even if in a particularly bad situation you may not profit from it quite that much.

So far, all these resulting values could simply be added. You can take the lowest damage value possible (factoring in damage variance), apply the hit probability to it, and know the lower damage rate of the weapon. Same for the upper value and in-between the average.

But the target can also influence the probability to hit or miss. E.g. scouts are often explicitly considered to be hit less likely than 100%, while battle ships are at 100% for many weapons. And this is why even any statistically calculated averages are to be taken with a grain of salt if unit variance enters into the equation. If you take the concept seriously, you'd have to state an average for each ship possibly targeted. This of course would make comparing weapons a bit difficult or cumbersome. So **the better approach is to use upper and lower limits of damage rate**. Even a totally differentiated weapon, that hits differently for basically each kind of ship allows to state a lowest and a highest probability for hitting a target, and derived from that, also a lowest and a highest damage rate. Together with the base damage, the firing rate and the damage variance the value for one particular weapon can be stated as having a highest damage rate and a lowest damage rate.

Real Life Example

To make a real life example, let's take the Negh'Var.

Overriding of Values

The *Negh'Var* is defined by the file *kbattle.odf*. When looking into that file you find the following information on conventional weapons:

```
#include "battle.odf"
[...]
// Phaser
weapon1 = "kbphas"
weaponHardpoints1 = "hp01" "hp02" "hp03" "hp04" "hp06"

// Photon Torpedo
weapon2 = "kbphot"
weaponHardpoints2 = "hp01" "hp02" "hp05" "hp06"
[...]
```

The *#include „battle.odf“* should be taken into consideration. When looking into *battle.odf*, you will find the following weapons-relevant information:

```
#include "craft.odf"
[...]
physicsFile = "battphys.odf"
```

That already combines to:

```
#include "craft.odf"
[...]
physicsFile = "battphys.odf"
[...]
// Phaser
weapon1 = "kbphas"
weaponHardpoints1 = "hp01" "hp02" "hp03" "hp04" "hp06"

// Photon Torpedo
weapon2 = "kbphot"
weaponHardpoints2 = "hp01" "hp02" "hp05" "hp06"
[...]
```

Looking up *craft.odf* and *battphys.odf* you will find that neither contains any relevant information on weapons. So the result shrinks to this:

```
// Phaser
weapon1 = "kbphas"
weaponHardpoints1 = "hp01" "hp02" "hp03" "hp04" "hp06"

// Photon Torpedo
weapon2 = "kbphot"
weaponHardpoints2 = "hp01" "hp02" "hp05" "hp06"
[...]
```

So the ship only has two conventional weapons, *kbphas* and *kbphot*. Those translate to files *kbphas.odf* and *kbphot.odf*.

Condensing kbphas.odf

This file contains the following relevant information:

```
#include "phaser.odf"

//Name of the ODF file for the ordinance for this weapon
ordName = "kbphaso"
[...]
//Time Delay between shots
shotDelay = 2.3f
[...]
```

Applying *#include „phaser.odf“*:

```
[...]
//Time Delay between shots
shotDelay = 2.0f
[...]
hitChance = 1.0
[...]
//Name of the ODF file for the ordinance for this weapon
ordName = "kbphaso"
[...]
//Time Delay between shots
shotDelay = 2.3f
[...]
```

Now let's include the information from the ordinance file *kbphaso.odf*:

```
#include "phasero.odf"
[...]
```

result:

```
[...]
//Time Delay between shots
shotDelay = 2.0f
[...]
hitChance = 1.0
[...]
//Name of the ODF file for the ordinance for this weapon
#include "phasero.odf"
[...]
//Time Delay between shots
shotDelay = 2.3f
```

```
[...]
```

The content of the referenced *phasero.odf* is still missing:

```
[...]
//The base amount of damage which the ordinance does with each hit
//Can be applied to specific ships in format default "shipname.odf" 0.9
(this is a number)
damageBase = 24
    "zmama.odf" 2

//The amount of variance applied to the damage (i.e. 10 + or - 2)
//Can be applied to specific ships in format default "shipname.odf" 0.9
(this is a number)
damageVariance = 2
[...]
```

Final result for this one weapon:

```
[...]
//Time Delay between shots
shotDelay = 2.0f
[...]
hitChance = 1.0
[...]
//Name of the ODF file for the ordinance for this weapon
damageBase = 24
    "zmama.odf" 2

//The amount of variance applied to the damage (i.e. 10 + or - 2)
//Can be applied to specific ships in format default "shipname.odf" 0.9
(this is a number)
damageVariance = 2
[...]
//Time Delay between shots
shotDelay = 2.3f
[...]
```

ODF Directives that appear multiple times have to be removed, leaving only the last instance of the same directive:

```
[...]
hitChance = 1.0
[...]
//Name of the ODF file for the ordinance for this weapon
damageBase = 24
    "zmama.odf" 2

//The amount of variance applied to the damage (i.e. 10 + or - 2)
//Can be applied to specific ships in format default "shipname.odf" 0.9
```

```
(this is a number)
damageVariance = 2
[...]
//Time Delay between shots
shotDelay = 2.3f
[...]
```

From that we can gather:

- hit chance = 100% (will never miss, no surprise, it's a phaser)
- Base damage = 24 (2 if you count the Mother Entity as well, but for multi-player considerations this is basically irrelevant).
- Damage Variance = 2
- shot Delay = 2.3, or firing rate is 0,43/s

We chose to ignore the *Mother Entity* aspect, so we can even use a proper average here.

Results for kbphas.odf

- The minimum damage rate is (24 base damage - variance 2) / shot Delay 2.3 s x 100 % hit change = 22/2.3s = 9.6/s.
- The maximum damage rate is (24 base damage + variance 2) / shot Delay 2.3 s x 100 % hit change = 26/2.3s = 11.3/s.
- Average therefore is 10,4/s

Condensing kbphot.odf

Again we have to condense the information applying overrides and so on. Starting with contents:

```
#include "photon.odf"
[...]
ordName = "kphotono"
[...]
//Time Delay between shots
shotDelay = 1.3f
[...]
```

Applying contents from *photon.odf*:

```
ordName = "photono"
[...]
shotDelay = 1.3f
[...]
hitChance = 0.7
    "bbattle.odf" 1.0
    "fbattle.odf" 1.0
    "kbattle.odf" 1.0
    "rbattle.odf" 1.0
    "fente.odf" 1.0
```

```
    "fentd.odf" 1.0
    "fgalaxy.odf" 1.0
    "fprem.odf" 1.0
    "fpremnew.odf" 1.0
    "blocutus.odf" 1.0
    "ktoral.odf" 1.0
    "kmartok.odf" 1.0
    "rsela.odf" 1.0
    "zjembat.odf" 1.0
    "zsonbat.odf" 1.0
    "zbreen.odf" 1.0
    "zcardbat.odf" 1.0
```

resulting in

```
ordName = "photono"
[...]
shotDelay = 1.3f
[...]
hitChance = 0.7
    "bbattle.odf" 1.0
    "fbattle.odf" 1.0
    "kbattle.odf" 1.0
    "rbattle.odf" 1.0
    "fente.odf" 1.0
    "fentd.odf" 1.0
    "fgalaxy.odf" 1.0
    "fprem.odf" 1.0
    "fpremnew.odf" 1.0
    "blocutus.odf" 1.0
    "ktoral.odf" 1.0
    "kmartok.odf" 1.0
    "rsela.odf" 1.0
    "zjembat.odf" 1.0
    "zsonbat.odf" 1.0
    "zbreen.odf" 1.0
    "zcardbat.odf" 1.0
[...]
ordName = "kphotono"
[...]
//Time Delay between shots
shotDelay = 1.3f
[...]
```

Using ordinance file *photono.odf*:

```
damageBase = 30
    "zmama.odf" 2
[...]
damageVariance = 2
```

we reach:

```
[...]
damageBase = 30
    "zmama.odf" 2
[...]
damageVariance = 2
[...]
shotDelay = 1.3f
[...]
hitChance = 0.7
    "bbattle.odf" 1.0
    "fbattle.odf" 1.0
    "kbattle.odf" 1.0
    "rbattle.odf" 1.0
    "fente.odf" 1.0
    "fentd.odf" 1.0
    "fgalaxy.odf" 1.0
    "fprem.odf" 1.0
    "fpremnew.odf" 1.0
    "blocutus.odf" 1.0
    "ktoral.odf" 1.0
    "kmartok.odf" 1.0
    "rsela.odf" 1.0
    "zjembat.odf" 1.0
    "zsonbat.odf" 1.0
    "zbreen.odf" 1.0
    "zcardbat.odf" 1.0
[...]
ordName = "kphotono"
[...]
//Time Delay between shots
shotDelay = 1.3f
[...]
```

Applying *kphotono.odf*:

```
#include "photono.odf"
[...]
```

Again no real change:

```
[...]
damageBase = 30
    "zmama.odf" 2
[...]
damageVariance = 2
[...]
shotDelay = 1.3f
[...]
hitChance = 0.7
```

```
    "bbattle.odf" 1.0
    "fbattle.odf" 1.0
    "kbattle.odf" 1.0
    "rbattle.odf" 1.0
    "fente.odf" 1.0
    "fentd.odf" 1.0
    "fgalaxy.odf" 1.0
    "fprem.odf" 1.0
    "fpremnew.odf" 1.0
    "blocutus.odf" 1.0
    "ktoral.odf" 1.0
    "kmartok.odf" 1.0
    "rsela.odf" 1.0
    "zjembat.odf" 1.0
    "zsonbat.odf" 1.0
    "zbreen.odf" 1.0
    "zcardbat.odf" 1.0
[...]
#include "photono.odf"
[...]
//Time Delay between shots
shotDelay = 1.3f
[...]
```

Using *photono.odf*:

```
[...]
damageBase = 30
    "zmama.odf" 2
[...]
damageVariance = 2
[...]
```

We reach this:

```
[...]
damageBase = 30
    "zmama.odf" 2
[...]
damageVariance = 2
[...]
shotDelay = 1.3f
[...]
hitChance = 0.7
    "bbattle.odf" 1.0
    "fbattle.odf" 1.0
    "kbattle.odf" 1.0
    "rbattle.odf" 1.0
    "fente.odf" 1.0
    "fentd.odf" 1.0
    "fgalaxy.odf" 1.0
```

```
    "fprem.odf" 1.0
    "fpremnew.odf" 1.0
    "blocutus.odf" 1.0
    "ktoral.odf" 1.0
    "kmartok.odf" 1.0
    "rsela.odf" 1.0
    "zjembat.odf" 1.0
    "zsonbat.odf" 1.0
    "zbreen.odf" 1.0
    "zcardbat.odf" 1.0
[...]
damageBase = 30
    "zmama.odf" 2
damageVariance = 2
[...]
//Time Delay between shots
shotDelay = 1.3f
[...]
```

Removing all the doubled directives:

```
[...]
hitChance = 0.7
    "bbattle.odf" 1.0
    "fbattle.odf" 1.0
    "kbattle.odf" 1.0
    "rbattle.odf" 1.0
    "fente.odf" 1.0
    "fentd.odf" 1.0
    "fgalaxy.odf" 1.0
    "fprem.odf" 1.0
    "fpremnew.odf" 1.0
    "blocutus.odf" 1.0
    "ktoral.odf" 1.0
    "kmartok.odf" 1.0
    "rsela.odf" 1.0
    "zjembat.odf" 1.0
    "zsonbat.odf" 1.0
    "zbreen.odf" 1.0
    "zcardbat.odf" 1.0
[...]
damageBase = 30
    "zmama.odf" 2
damageVariance = 2
[...]
//Time Delay between shots
shotDelay = 1.3f
```

So we can derive:

- hit Chance is between 0.7 and 1.0

- base damage is 30 (again ignoring *Mother Entity*)
- damage variance is 2
- shot delay is 1.3 seconds or a firing rate of 0,77/s

Results for kbphot.odf

- The minimum damage rate is (30 base damage - variance 2) / shot Delay 1.3 s x 70 % hit change = 28/1.3sx0.7 = 15.1/s (e.g. when encountering a Cruiser instead of a battle ship, and being really unlucky with lower limit damage rate).
- The maximum damage rate is (30 base damage + variance 2) / shot Delay 1.3 s x 100 % hit change = 32/1.3s = 24.6/s (when encountering a battle ship being lucky with the upper limit damage rate).
- Average is **not** 19.8/s, because that would require an averaging over the different ships.

Results for Both Weapons

To get the damage rate of the ship they are basically simply added:

- The minimum damage rate is 9.6/s + 15.1/s = 24.7/s.
- The maximum damage rate is 24.6/s + 35.9/s.

Note: This of course assumes, that you are in firing range of both weapons. If the weapons have different firing ranges, then you might encounter the problem, that the target is within reach of the longest reaching weapon, only. Such considerations are not taken into account here. As one is usually only interested in the values resulting from optimal firing range, the *Negh'Var* has a damage rate between 24.7/s and 35.9/s, not counting its Ion Cannon, or (when things are really dire) the Self-Destruct.

[**Star Trek: Armada**] [Game Play] [Technical Support]

[Units] [Hero Ships] [Ship Types] [Stations] [Station Types] [Special Weapons] [Factions]

[Back to the Games Database]