Automatic Ammunition Configuration

Overview

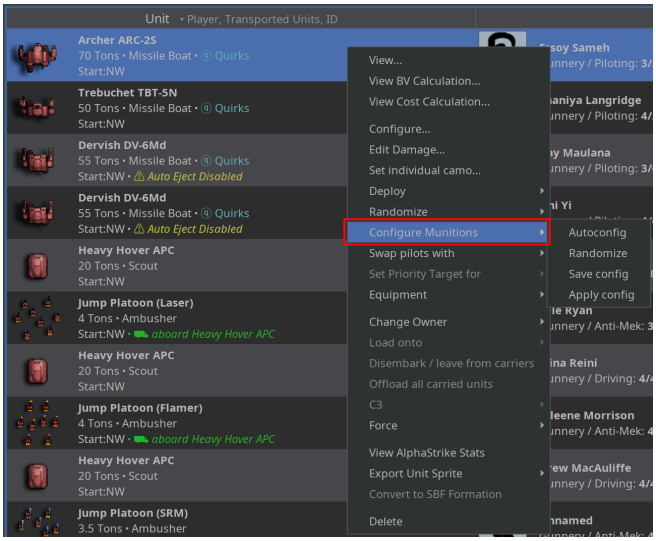
We've added a section in the Player Configuration / Player Settings dialog to adjust ammunition bins and ASF bombs semi-intelligently.
A subset of the same options have also been added as a context menu entry in the Lobby Unit List.

These controls allow players to kit out ammo-using units a few different ways:

1. Autoconfig: select appropriate munitions for each unit using information about the current game settings, opposition units, time of day, and friendly unit capabilities.
2. Randomize: randomly select valid munitions from available options (only damage-dealing ammo by default).
3. Save/Load Loadouts: save the current configuration, or load a previously-created configuration.

Players can autoconfigure selected units of the same Team in the Lobby Unit List (via the right-click menu), or via the Configure Player… dialog.

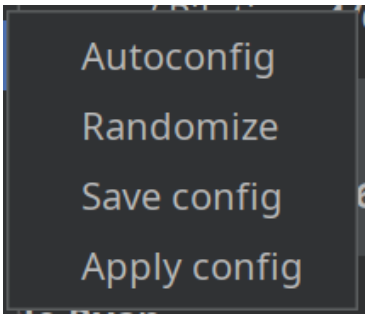
Lobby Unit List + Context Menu



A player can select any number of units (on the same team) and use the autoconfigure options to refit the selected units with any applicable alternative munitions, depending on various factors such as the year, tech level, etc.

Note: currently the Context Menu options use “Inner Sphere - General” as the faction when auto-generating munition loadouts. To use a specific Faction for auto-configuration, use the Configure Player dialog instead.

Options

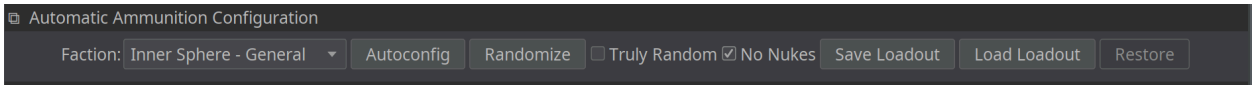


- **Autoconfig**: applies semi-intelligent processing that generates a set of desired munition types for each weapon type (based on the selected units and opposing units) and installs them. Currently uses “IS - General” as the faction for availability checks.
- **Randomize**: replaces all munitions with randomized valid munition choices. Operates the same as the Config Player “Randomize” button _without_ “Truly Random” checked.
- **Save config**: creates a new .adf file containing entries for each of the selected units, with imperatives matching the currently-installed munition types for each bin. Applying this config to the same set of units (or any superset including them) will result in the units’ munitions being set to they munition types in the .adf file *if* other parameters (game year, faction [if from the Configure Player Dialog]) remain the same.
  Note: these entries are specific to the unit type, variant, and pilot; these configs can be generalized via manual editing.
- **Apply config**: load an .adf file and attempt to apply its imperatives (see ADF Format for more details) to the selected units.
  Note: Depending on the source of the .adf file, the specificity of the imperatives, and the year and tech level of the current game, some imperatives may not apply completely or at all.

Configure Player Dialog

This can be used to configure a player's own units, or Bot units, and will work in network games.

Controls



There are a few obvious, and not-so-obvious, inputs to these capabilities. The obvious adjustments to be made are:

- "Faction" dropdown *Defaults to "IS - General"*
  Choose your team's faction to adjust which munitions are available. This will only affect a few factions at specific times.
- "Autoconfig": Applies the selected options to the entire list of units owned by the currently-selected player.
- Randomize: randomly select valid munitions from available options
- "Truly Random" checkbox *Defaults to Off*
  - Off: "random" ammo only includes ammo that deals damage;
  - On: ammo is fully randomized and results may be useless / suboptimal.
- "No Nukes" checkbox *Defaults to On*
  - Off: allows equipping nuclear munitions on relevant units (Davey Crockett, Alamo, etc.) if the game options allow.
  - On: ignore game options and never equip nukes.

But other game option settings will also influence how these controls function:

- Allowed Units and Equipment
  - Year: determines which munitions are available in general, and to specific factions.
  - Tech Level: determines which munitions are available in general; most optional munitions are Advanced or Experimental
  - Variable Tech Level: can make some munitions available earlier depending on Tech Level and canon development time
  - Ignore Clan Ammo Limitations: expand the range of munitions available to Clans
- Advanced Rules
  - TacOps Double Blind: this will prevent the autoconfig tool from evaluating enemy forces, so munitions will be selected based on friendly capabilities only (e.g. Flak rounds are less likely)
- Advanced Aerospace Rules
  - Allow Nuclear Weapons: this must be enabled _and_ "No Nukes" must be unchecked in order for nuclear munitions to be generated.

When the "OK" button is clicked, the new ammo configurations are applied!
Hitting "Cancel", or explicitly restoring the original configuration, will prevent the dialog from making changes.

Autoconfig logic

When Autoconfig is clicked the system assesses the battlefield conditions (day/night, ground/space), friendly forces (including TAG/NARC capabilities), enemy force composition and capabilities (if Double Blind is not set), and attempts to choose four of the most effective ammo types for each weapon system type.

- If conditions are not “daylight”, illuminating munitions are weighted higher: Inferno, Tracer, and artillery Illumination, etc.
- If friendly forces include NARC or TAG, guided munitions will be weighted more highly; if none are included, guided munitions are weighted lower.
- Depending on *Blind settings and enemy force composition
    - munitions that increase or reduce rounds/ton may be weighted differently;
    - higher-damage munitions may be selected depending on enemy armor types;
    - Inferno rounds are favored against energy-heavy forces and vehicles;
    - rounds with extra anti-personnel damage will be weighted higher if there are many infantry or BA troops; etc.
- If the enemy forces include a large number of Airborne units, Flak rounds will be weighted higher.
- If the enemy force includes a number of high-MP units, Precision, seeking, and guided munitions may be selected

Caveats

The feature as presented in 0.50.0 is in continued development, and players should note that it still has some limitations:

- Weather / planetary conditions outside of day/night are not considered; e.g. hurricane winds may affect missile accuracy but this is not currently modeled or accounted for..
- "Utility" munitions such as Smoke, Thunder, Illumination, etc. are not selected by default. Princess doesn't know how to use these munitions so they are weighted very low and will only be generated by the Randomize button.
- ASF bomb loadouts are semi-randomized based on the units’ faction and make-up, and do not yet incorporate all of the automatic configuration options of other munitions
- Due to the nature of the .adf file format, saved *manual* loadouts are only applicable to the units from which they derive (but can be later edited to be more generic).
- MHQ limitations, such as players' factions or ammo supplies, do *not* apply. It is possible to equip forces with munitions that they don't own in their Campaign after loading into MM for a mission, which may significantly increase difficulty.

ADF Format

In order to make it possible for players to customize and re-use loadouts, we created a new file format that can be saved and loaded from this dialog. If a player manually sets their munitions via the lobby, they can save the configuration for loading in future games.
(Note that entries will be saved for each specific chassis, variant, and pilot in the player's list, and will only apply to that specific combination.)

By contrast, the auto-generated configurations will be as generic as possible: a single imperative defining the top four munitions for each bin type will be applied to all units.
The file format is designed to be human readable and editable.

Format

Each line of an .adf file consists of two main parts:
The chassis, variant, and pilot to which it applies, separated by single colons
A set of imperatives, separated by double colons,

The imperative set is open-ended and can contain unlimited entries.
Each imperative itself consists of two parts: ammo type, and munitions in order of importance.
e.g.:

```
# Below is an example entry that will apply to any Crusader chassis units
      |-- Chassis
      |   |-- Variant (here, "any" will match any variant)
      |   |   |-- Pilot (here, "any" will match any pilot)
      |   |   |    |-- First imperative starts with Ammo bin type
      |   |   |    |   (if size is specified, applies only to that size; otherwise, applies to all of that type)
      |   |   |    |        |-- Primary, and default, munition to load on matching units.
      |   |   |    |        |   The default ammo will fill any excess bins after all imperatives are fulfilled.
      |   |   |    |        |               |-- Secondary munition.  Exactly one bin will be filled with this
      |   |   |    |        |               |   ammo if more than one matching bin exists.
      |   |   |    |        |               |         |-- Tertiary entry.  Exactly one bin will be filled with this ammo
      |   |   |    |        |               |         |   if more than two matching bins exist.
      |   |   |    |        |               |         |           |-- Second imperative (if desired)
      |   |   |    |        |               |         |           |                                       |-- Third imperative
      |   |   |    |        |               |         |           |                                       |   (if desired, etc.)
Crusader:any:any::LRM 15:LRM 15 Dead-Fire:Standard:Narc-capable[::SRM 6:SRM 6 Dead-Fire:Standard:Inferno[::...]]
```

("[" and "]" indicate optional sections and should not be included in imperatives)

To set all matching bin types to the same ammo, simply provide the desired munition type as the only imperative, e.g.:

```
::AC:Precision
```

Any bins that match an imperative but have no ammo type explicitly stated will receive the first, or default, ammo type.
Any bins that do *not* match an imperative will not be reconfigured.

Any unit that does not match *any* entry will not be reconfigured.
Entries are applied in the following order:

1. Explicit chassis-variant-pilot match (e.g, "`Shadow Hawk:SHD-2H:Alexander Romanov`"
2. Explicit chassis-variant + "any" pilot match
3. Explicit chassis + "any" variant match, "any" or specific match for the pilot field
4. "any" chassis match + explicit/"any" variant + explicit/"any"

Again, any unit that doesn't match an entry at all will not be reconfigured.

Auto-generated Configuration

The current autoconfigure code produces a single, maximally-applicable, entry that covers all weapon types:

```
any:any:any::Narc:Standard:Narc Explosive::Mek
Mortar:Standard:Airburst:Anti-personnel:Semi-Guided::AC:Standard:Flak:Caseless:Precision::
LRM:Dead-Fire:Standard:Fragmentation:Heat-Seeking::Artillery
Cannon:Standard:Fuel-Air::ATM:HE:ER:Standard::Arrow
IV:Standard:Inferno-IV:Cluster:ADA::Artillery:Standard:Flechette:Cluster:Fuel-Air::Bomb:St
andard:ASMissile Ammo:ClusterBomb:RL 10 Ammo
(Bomb)::SRM:Dead-Fire:Standard:Tandem-Charge:Acid
```

Although the default configuration system selects 4 imperatives for each weapon type, fewer or more are allowed.
(Note that the ::Bomb directive is generated but not currently used)

Randomized Configuration

The randomized configuration produces an even simpler single entry:

```
any:any:any::Mek Mortar:Random::Narc:Random::AC:Random::LRM:Random::Artillery
Cannon:Random::ATM:Random::Arrow IV:Random::Artillery:Random::Bomb:Random::SRM:Random
```

This matches any unit, and generates a different random ammo type for each bin of each weapon system. Note that variation may be limited, depending on year, faction, and game settings.

Saved Manual Configuration

On the other hand, if a user customizes their loadout in the lobby and then wishes to save it for later re-use, the entries will be much more specific:

```
Catapult:CPLT-K2:Cheyenne Zogu::MG Ammo:Standard
Brutus Assault Tank::Monica Olba::LRM 20:LRM 20 Dead-Fire:Standard:LRM 20 Follow The Leader::SRM 2:SRM 2
        Dead-Fire::SRM 6:SRM 6 Dead-Fire
Brutus Assault Tank::Bente Dybdahl::LRM 20:LRM 20 Dead-Fire:Standard:LRM 20 Follow The Leader::SRM 2:SRM 2
        Dead-Fire::SRM 6:SRM 6 Dead-Fire
Crusader:CRD-3R:Ru-su Toyota::LRM 15:LRM 15 Dead-Fire:Standard::SRM 6:SRM 6 Dead-Fire::MG Ammo:Standard
Thorn:THE-S:Pori Hansson::LRM 5:LRM 5 Dead-Fire
Vedette Medium Tank:(AC2):Zak Crawford::SRM 2:SRM 2 Dead-Fire::AC/2:Flak AC/2::MG Ammo:Standard
Vedette Medium Tank:(AC2):Mikaho Joe::SRM 2:SRM 2 Dead-Fire::AC/2:Flak AC/2::MG Ammo:Standard
Awesome:AWS-8R:Iwane Jarakian::LRM 15:LRM 15 Dead-Fire:Standard
Peregrine Attack VTOL:(Kurita):Yasuo Doyle::SRM 4:SRM 4 Dead-Fire
Peregrine Attack VTOL:(Kurita):Heidi Kichida::SRM 4:SRM 4 Dead-Fire
Warrior Attack Helicopter:H-7:Teruyo Almaa::SRM 4:SRM 4 Dead-Fire::AC/2:Flak AC/2
Warrior Attack Helicopter:H-7A:Tatsui Nomura::AC/5:Flak AC/5::MG Ammo:Standard
```

Each entry will only match one specific chassis-variant-pilot combo, but a user can make these entries more generic and/or dynamic by:

- Replacing variant and pilot fields with "any"
- Removing weapon size from imperatives
- Replacing some or all imperatives for a give weapon system with the "Random" keyword

The accepted Chassis and Variant names, as well as Weapon Type and Munition names, are determined by MegaMek's internal codes.
If in doubt, I recommend saving off a config containing the desired munitions to use as a reference before editing any .adf files.

Loading Saved Configurations

To apply a manually-edited .adf file to a force, simply click "Load Loadout" (or “Apply Config” in the context menu), select the file, and then click "OK"; the Lobby should show that all the player's units have been modified.
If nothing seems to be happening when using Autoconfig, Randomize, or Load Loadout, check the MegaMek log file for errors.

Autoconfig Settings: munitionLoadoutSettings.xml

In order to allow developers (and adventurous users) to quickly tune the autoconfig system, an editable xml file containing the weights and thresholds used by the autoconfig system has been introduced:

```
<MHQ or MegaMek dir>/mmconf/munitionLoadoutSettings.xml
```

If this file exists, the Autoconfig system will use the values found here when calculating whether to increase or decrease the weight of given ammunition categories, which thresholds to apply when determining if enemy or unit counts meet a requirement, and how much to increase and decrease weight settings *by.*

This file is meant for testing purposes and for publishing tweaks to the Autoconfig system.
Users are advised to back the file up prior to making any changes.
However, if any fields (or the file itself) are deleted, the default values will be used.

Not all fields or values will be enumerated in this document; for help, please visit the MegaMek discord.

Field types

The file uses a simplified XML format that conforms with the Java Properties XML format. Every block should have an opening and closing tag, there is only one allowed `<comment>...</comment>` block, and every `<entry>...</entry>` block must have a unique key value string. Key value strings starting with "`comment…`" will be ignored and may be used for adding user comments, but again, each key value string must be unique.

There are <N> major block or entry types:

- `<properties>...</properties>`
  Required by file format. Contains all other blocks.
- `<comment>...</comment>`
  Optional; unused by autoconfig, currently contains a description of the file's purpose.
  Only one such block is allowed in the file.
- `<entry key="">...</entry>`
  Optional; every key= string must be unique within the file.
- `<entry key="comment…">...</entry>`
  Optional; used to delineate subsections or describe the usage of related entries

- `<entry key="*Factor">X.Y</entry>`
  Optional; used when multiplying a force or subset of the friendly or enemy units to determine whether to increase or decrease a munition category.
  E.g. `<entry key="mtReducingAmmoReduceIfUnderFactor">2.0</entry>` will reduce the weight of munitions that have fewer rounds per ton from their default if the factor of friendly units to enemy units is less than 2.0 to 1 (that is, if the force being autoconfigured does not outnumber the enemy by 2.0 to 1, make those munitions less likely to be chosen).
- `<entry key="*FractionDivisor">X.Y</entry>`
  Optional; defines the fraction of a friendly or enemy force that must be exceeded to increase or decrease some munition category's weight.
  E.g.: `<entry key="mtFlakEnemyFliersFractionDivisor">4.0</entry>` will increase the weight of Flak munitions (LB-X Cluster, AC Flak, Artillery Cluster or HE, ADA Missiles) if airborne units (VTOL, WiGE, or Aerospace) make up more than (1.0 / 4.0) of the enemy forces.
- `<entry key="*Threshold">X.Y</entry>`
  Optional; defines the number that must be exceeded to raise or lower a munition category's weight.
  E.g. `<entry key="mtFlakMinBombersExceedThreshold">0.0</entry>` will raise the weight of Flak munitions if the enemy force contains more than 0.0 bombers (that is, 1 or more).

There are a few more commonly-used entry key types but hopefully these will be self-explanatory, or explained by comments in the file.

Default Weights

The simplest section to adjust is the MWC ("MunitionWeightCollection") section where default weights for all munitions, as well as adjusted defaults for some specific munitions, are set. These weights are adjusted via analysis of the planetary environment, friendly units, and enemy units, and then the top N (where N is set by `mtTopMunitionsSubsetCount`) are selected and loaded into each unit. These entries are:

```
<entry key="commentStartMWCArea">The following entries are used by MWC:</entry>
<entry key="defaultWeaponWeight">1.0</entry>
<entry key="defaultStandardMunitionWeight">2.0</entry>
<entry key="defaultMissileStandardMunitionWeight">2.0</entry>
<entry key="defaultDeadFireMunitionWeight">3.0</entry>
<entry key="defaultArtemiscapableMunitionWeight">0.0</entry>
<entry key="defaultATMMunitionWeight">2.0</entry>
<entry key="defaultATMStandardWeight">1.0</entry>
<entry key="increaseWeightFactor">2.0</entry>
<entry key="increaseWeightIncrement">1.0</entry>
<entry key="decreaseWeightFactor">0.5</entry>
<entry key="decreaseWeightDecrement">0.0</entry>
```

In order:

- **defaultWeaponWeight**: the default weight initially assigned to _all_ munitions for every weapon type.
- **defaultStandardMunitionWeight**: the weight applied to Standard (that is, the most basic munition type) for all weapons by default.

- **defaultMissileStandardMunitionWeight**: the default weight for Standard missile munitions (applies to all missile weapon types)
- **defaultDeadFireMunitionWeight**: default weight applied to Dead Fire munitions *specifically*, for both LRM and SRM launchers.
- **defaultArtemiscapableMunitionWeight**: default weight applied to Artemis-capable munitions *as a general munition type*. If greater than 0.0, there is a chance friendly or enemies without Artemis FCS installed may be assigned Artemis munitions. Units with Artemis FCS will have always get Artemis-compatible munitions assigned by the Autoconfig tool.
- **defaultATMMunitionWeight**: default for ATM munitions *other* than Standard. HE and ER are generally considered more useful than Standard ATM munitions.
- **defaultATMStandardWeight**: default for Standard ATM munitions; set lower than HE and ER by default.
- **increaseWeightFactor**: Amount a weight is multiplied by when some threshold, count, factor, or divisor causes the Autoconfig tool to decide an increase is warranted.
- **increaseWeightIncrement**: Amount added to an increased weight after the factor is multiplied.
- **decreaseWeightFactor**: Amount a weight is multiplied by when the Autoconfig tool determines decreasing the weight is required.
- **decreaseWeightDecrement**: Amount subtracted from a weight after it is multiplied by the decrease factor

The default factors and increment/decrement values mean that a munition category that is increased once (say, because of an enemy type meeting a threshold) and then decreased once (say, because of a planetary environmental factor) will still rank higher in the end than a munition category that was not adjusted at all.

Changing these weights and factors should result in rapid changes in the types of munitions selected.