Stats manual

for Stats version 1.13

A. van Lomwel

Date : 2003/06/1817 : 31 : 58

Contents

Chapter 1

Introduction

Stats is a program that was born out of the desire to store historical data for a game of vgaplanets by extracting some critical data from the game's data files. When you have access to the data why not generate some nice reports from it, etc. Today stats combines the building of this historical data, extensive reporting facilities, unpack of rst files, and what randmax can do in one program.

Stats has two major modes of operation; the command line described in chapter 3 , and the configuration files described in chapter 4. We will start off with some details concerning the inner working of stats which might help to understand some surprising output from the program.

1.1 Inner workings

Stats goes through the following steps in the program:

1. Set defaults.
2. Analyse command line options.
3. Parse master configuration file. This implies that the command line settings are overruled by the master configuration file.
4. Load game data (truehull, engspec, xyplan.dat, etc.).
5. For all players do:
   (a) Unpack rst file.
   (b) Load player data.

(c) Print command line reports.

(d) Load payer command file. While parsing the lines in the configuration file lines the following is done:

i. Parameter settings are applied immediately.

ii. Reports are generated immediately.

iii. Planetary commands are added to the command list.

(e) Prioritise commands.

(f) For all objects (planets, ships, etc.) the commands are executed. More specific commands at the top of the list overwrite the more generic ones at the end.

i. Turn specific commands.

ii. Object specific commands.

iii. Object class commands.

iv. General commands.

v. Universal commands.

(g) When necessary write player data to disk

The position of the commands in the order of execution implies it is not possible to generate reports to reflect the situation *after* commands have been executed. The items in the reports are printed in a fixed order. It is not possible to change this order. Summarised: the order in which everything is executed is built into the program, the order in which command line options are passed is irrelevant.

1.2 Calculation accuracy

All formulas used in this application are based on phost, and match to the dot.

1.3 Compatibility

This sections deals with interoperatability of stats with the many tools that exist around planets. The developers have most experiance with the following set of tools; phost, planets.exe, vpa, tkf, exploremap. All results and predictions are tuned to match exactly those of phost.

host.exe Only as far as it is compatable with phost. No problems expected.

winplan Files are accepted but unpacked as with unpack.exe. (The dos way.) The change option does not take the addittional checksums of winplan into account. In short Winplan cannot be used together with stats, although stats can work with winplan.rst files, and registration.

exploremap Is fully supported. In case of missing xyplan.dat stats looks for xyplan<playerid>.dat files in the player data directory. In case of multiple player files these files are merged. When a xyplan.dat is present in the Base directory xyplan¡playerid¿.dat are not read.

util.dat Stats does not do anything with this file.

phost Switches settable in the master commandfile are fully supported in all functions they are relevant. All other switches are assumed to be at their default. Bigtargets is supported on top of this.

alternative ship lists Should be supported without a problem. The short ship type might result in double naming. See **??** for further details. (Some alt list have a config file for cloak and gravitonic, but this config file is not used).

Alternative universe maps Are fully supported. Explore map games should stay within 1 - 5000 range with the planet coordinates, since planets outside this range are not merged into the planposition structures.

mission.ini –sanity assumes the mission.ini file coming with phost for mission numbers. Same goes for the -cbB options. -m accepts all types of files.

vpa Fully compatable, just don't run a datafile changing option while having the client accessing the datafiles.

dos planets Same as vpa above.

unregistered players Stats assumes you are registered. Stay away from starbase techlevel raising build commands when you are not. Stats will not complain, but your host will.

Chapter 2

Examples

In the below examples the prompt is displayed in the normal manner:

```
$>
```

Unpack game files:

```
$> stats -ut <gamedir> -B<basedir>
```

randmax:

```
$> stats -j<MasterConf> -i<raceid>,<conffile>
```

For the master configuration file with host parameters, see section 4.2
Player-specific command file:

```
# First build 15 factories, then 20 mines and 15 defence, then continue
# with maxing factories and 250 mines, then build a starbase:
planet 12: f15 m20 d15 f999 m250 b
# Max base with 60 fighters & 200 defense posts:
planet Wolf 359 : F40+20 p200
# Tax all planets:
tax
```

See also script directory for some further examples.
Show planet names and mineral output for player 6:

```
$>stats <gamedir> -B<basedir> -p6 -dnod
```

Set tax and randomise planetary friendly codes:

```
$>stats -ctf
```

List of options:

```
$>stats --help
```

Chapter 3

Command line options

The list below gives all available command line options. Following the full list is a further explanation of those options that mere mortals might not fully understand from the brief descriptions in the list below.

```
Usage: stats [-a][-B<path>]
             [-b[A][n][l][F][d][D][f][r][m][t][b]]
             [-c[t][f]]
             [-d[n][l][c][f][g][r][D][o][s][i][m][p][h][t]
             [T<happylow>][d]]
             [-f[path]][-help][-l][-o<filename>][-p<player>]
             [-r[i][w][b][c<t,b,e>][r]][-s][-u[t]][-v]
             [-w][-t<happylow>]
             [-x[n][d][T][t][w][m][o][h][W][M]]
             [path]
-a : Include all planets in report (also the unpopulated).
-b : List starbase related information.
  A: All options.
  b: Building order at starbase.
  d: Defence (<base>+<planet>) / <base>.
  D: Damage.
  f: Fighters stationed here.
  F: Friendly code.
  l: Location of base in galactic coordinates x,y.
  m: Mission.
  n: Name of the planet the starbase is orbiting.
  r: Reserves of torpedos.
  t: Techlevels.
-B<path>: Path pointing to planets base dir ( Hullspec, etc.)
          Default path is player data path.
-c : Change player data files.
  b: Create/Replace beamup commands.
  B: Undo '-cb'.
  t: Set tax (see 'T' command line option).
  f: Randomise planetary friendly code.
-d : Display planet related information.
  A: All options.
  c: Climate.
  C: Consumption of supplies due to overpopulation.
  e: Population expansion.
  d: Display defense position (starbase==*).
  D: Depletion time in turns per mineral.
  f: Friendly code.
  g: Ground deposits per mineral.
  G: Tax levied in one maximum step of growth taxing.
```

```
  h: Happiness values.
  i: Industry (mines & factories).
  l: Location of planet.
  L: Tax levied at maximum sustainable tax rate.
  m: Money (cash & supplies).
  M<turns>: # mines needed to empty planet in [turns] turns.
            Default: 20.
  n: Planet name.
  o: Display planetary output per mineral.
  p: Population.
  P: Maximum population.
  r: Retrieval rate per mineral.
  s: Display surface stores per mineral.
  S: Maximum sustainable tax rate.
  t: Current tax rate.
  T: Maxtax.
-e : Common error detection.
  A: All options.
  d: Ship Fcode depending on mining mission.
  b: bdm.
  M: ships without missions.
  c: cln.
  T: Beam Transfere fcodes.
  P: PBx.
  e: ETA > 1.
  l: lfm.
  B: Bases without build order or mission.
  D: planets with d > 49.
  m: mkt.
  s: special missions.
  p: pop.
  N: NTP.
  f: fuel shortages.
-f[path]: Stores various data in files in directory pointed to
          by path. Default path is current dir.
-h : This list.
-i<1|2|3|4|5|6|7|8|9|a|A|b|B>,<filename> : Parse commandfile.
-j<filename>: Parse configuration file.
-l : Display taxes levied from your subjects.
-m : Display mission list.
```

```
-M : Dump outgoing messages.
-o<filename>  : Append output to file 'filename'.
-p[1|2|3|4|5|6|7|8|9|a|A|b|B]: Select players. Cumulative.
                                Default: all players.
-r : Display all hulltypes.
  A: All options.
  c: Building costs of complete ship as specified with 'l'.
  i: Special indices and ratios.
  l<torp,beam,engine>: Techlevels. Defaults: 9,7,9.
  r: Empty and full maximum ranges.
  t: Short ship type.
  w: War relevant parameters.
  1|2|3|4|5|6|7|8|9|a|b: Select players. Default: all players.
-s : Gives total industrial production of players
-T<Colonists[,Natives]>: Set happiness targets for taxation.
                         Default: 70,70.
-t : Display data on enemy targets.
  A: All options, except T (-tAT == everything).
  c: Cargo in target, including fuel and weaponry.
  h: Hull mass (no weaponry).
  l: Location.
  n: Name.
  r: Race id.
  t: Hull type.
  T: Hull type long.
  w: Weight.
-u : Unpack player file first. Requires '-p' option.
  t: Create aditional plain text file containing the messages.
-v : Give version.
-w : Make distance table.
-x : Display ship data.
  a: Ammunition (torpedoes/fighters) on board.
  A: All options, except T (-xAT == everything).
  b: Neutronium that will be burned this turn.
  c: Cargo space.
  d: Damage.
  e: Race number of primary enemy.
  f: Friendly code.
  h: Heading.
  H: Hull mass.
```

```
  m: Minerals on board.
  M: Mission.
  n: Name of ship.
  o: Non-mineral ship contents.
  t: Short ship type name.
  T: Long ship type name.
  w: Weapons.
  W: Weight of ship after beam transfer.

path : Directory with player data. (including trailing slash)

--turn       : Display turn number from player files.
--help       : Same as -h.
--defense    : Display overview of interesting number
               of defense posts.
--sanity     : Display report of potentially unwanted
               common errors like running out of fuel.
--mines      : Same as -dpcsgM20rPn -a.
--population : Same as -dpPceLid.
--myhulls    : Same as -rtwc .
```

3.1 Additional information on the command line options

This section contains some additional information on some command line options, but lets first start of with some general remark concerning the parsing of command line options. Options are parsed from left to right. Later setting override earlier ones. The toggles activating information items on the reports only *enable* items, they do not disable. Adding more switches adds more informatio. Consecutive report commands do not generate additional reports. So '-dnpP' is equal to '-dn -dpP'.

-a This is a switch to -d and -w. Without it -d only shows planets with colonists, with -a it also displays the info on not-yet colonised planets. The same with -w, there -a also includes the not-yet colonised planets.

-B should be specified including the trailing '/'. This is also the location where the mission.ini file is expected.

-c When this option is specified all player data files are written at the end of the execution phase.

-cf Randomises all non-special friendly codes to another non-special friendly code. The resulting friendly code is human readable as required by host.

-cb This options check for ships with a beamup special mission (the special mission, not the 'gather molybdenum'-type). When such a ship is found a matching beamup line in the outgoing messages is looked for. When this beamup line exist and specifies all possible parameters, or does not exist the ship us unloaded and a beamup line is created loading the cargo that was beamed down back unto the ship.

Restated, this only applies to ships with the beamup special mission set. In principle all cargo is unloaded to the planet and the beamup line is calculated to beam everything back to the ship again. If an existing beamup line is present, it's contents are taken into account.

In practice this boils down to privateer protection. Set the beamup special mission and stats rains everything down and beams it up again, untouchable by those nasty stealing basterds. That 'll teach them!

To allow easy changes to the cargo content the -cB option has been created. This loads all beamup commands into the ships so changes can be made, getting the right amounts of cargo into the right ships.

-ct Taxation uses the growth method. The lower limits can be set using the -T options. The turn before the 100% happiness would be exceeded the natives are taxed. The tax level used is the lowest tax rate that generates the same amount of MC as the tax rate that would set the population to the happiness threshold next turn. Basically, this means the natives are taxed as much as possible, but keeping in mind their happiness and the amount of available tax collectors (colonists).

When the player is Borg some exceptions exists[1]:

- Natives are taxed towards the native happiness threshold as fast as possible with collection of all coins.
- Colonist are taxed using the growth method only after the natives have disappeared. Before that time colonist are taxed to keep them at 100%.

-dC Phost formulas are used.

-dD Using the current number of mines. Trans uranium decay is accounted for, meteors not.

-dG Outputs the amount of tax levied from the native population when giving them the maximum 100 to 70 happiness kick using the growth method. This is the same as the number of colonists you need to get this maximum efficiency out of the growth tax method.

-dM the optional argument sets the amount of time you allow for stripping the planet clean of it's minerals. When this deadline cannot be met by 999 or less mines 'In' is displayed (for 'infinite').

-dP The maximum planetary population is calculated based on the race owning the planet. In case of an unowned planet the player's id is used. Since different races allow different populations to live on a planet, this might not have the effect you expected.

-dT for more details on tax see the -l report.

-f when a directory is specified this should end with a trailing slash, otherwise it is seen as a file prefix. The files listed below are created in that (or otherwise the current) directory. If the files already exist, the

[1] Guess which race Alexander was playing when he added this functionality – Reinout

information is appended to them. If you do this four times in one turn, the same information gets added four times, so you're warned.

Each line contains the turn number and the information item. This is ideal food for gnuplot. A file with sample gnuplot commands is provided in the `scripts/` directory. Files generated all have a name formatted like:

`VGA<playerid><code>{stats|score|dbase}.dat`

`stats` indicates the file contains statistical data, `score` indicates the file contains score information, and `dbase` indicates the file contains a set of multiple information items. What the codes mean is explained below.

- Score:

  P Planets.

  B Bases.

  C Capital ships.

  F Freighters.

  S Ships = Capital ships + Freighters.

  TP Total number of planets owned by all players together.

  TB Total number of bases owned by all players together.

  TC Total number of capital ships owned by all players together.

  TS Total number of ships owned by all players together.

  TF Total number of freighters owned by all players together.

- Dbase:

  T Targets.

- Stats:

  F Total neutronium generated by empire, alchemy excluded.

  T Total tritanium generated by empire, alchemy excluded.

  D Total duranium generated by empire, alchemy excluded.

  M Total molybdenum generated by empire, alchemy excluded.

  C All MC available.

  c Total tax income.

  f All neutronium available.

  t All tritanium available.

d All duranium available.

m All molybdenum available.

p Total number of defence posts.

r Total number of mines.

I Total number of factories.

i All supplies available.

gF Neutronium still in the ground.

gT Tritanium still in the ground.

gD Duranium still in the ground.

gM Molybdenum still in the ground.

-i Note the comma. It is required.

-l This options tells everything you may want to know about taxation. Some headings might not be clear at first sight, they are explained below.

Hap The current happiness of the colonists.

Hp0 How happy your colonist will be next turn when they don't have to pay taxes.

Mt The maximum tax rate you can set while keeping them happy above happy target.

Ut The lowest tax rate that will earn you as much as the Mt rate.

Natives from left to right: native clans, race, government.

Ind What would be levied at the current Ut rate when colonist were not an issue.

Take Same as Ind, but now taking colonists into account.

-o Redirects all non-error output.

-rt the short hull types are generated using the first letter of each word in the long ship type name. This results in some double codes for the standard ship list. Table 3.1 contains the modifications made to make all codes unique. When using non-standard ship lists one should be aware these might introduce new double codes. No check is performed on this in the code generation. Normally this is not an issue, but ships to get traded or nicked. This short code ends up for instance in the history data files, there it is harder to distinguish between ships, hence the remapping.

ANNIHILATION CLASS BATTLESHIP	ACB	⇒ ANNI
AUTOMA CLASS BASESHIP	ACB	
BLOODFANG CLASS CARRIER	BCC	
BIOCIDE CLASS CARRIER	BCC	⇒ BIO
GORBIE CLASS BATTLECARRIER	GCB	⇒ GORB
GOLEM CLASS BASESHIP	GCB	
SAGE CLASS FRIGATE	SCF	⇒ SAGE
SABER CLASS FRIGATE	SCF	
SAURIAN CLASS LIGHT CRUISER	SCLC	⇒ SRN
SCORPIUS CLASS LIGHT CARRIER	SCLC	
SUPER STAR CARRIER	SSC	
SUPER STAR CRUISER	SSC	⇒ SSCR
VICTORIOUS CLASS BATTLESHIP	VCB	⇒ VICY
VIRGO CLASS BATTLESTAR	VCB	

Table 3.1: Remapped short ship types.

-ri Gives some special indices which are useful for evaluating ship types.

P/i Punch per investment. The higher the more bang for the buck.

PMi Punch times mass per investment. Big bang is no fun when one goes kaboom oneselfs.

P/i Same as before, but now the price of the fish[2] is also accounted for.

PMi Same as before, but now the price of the fish is also accounted for.

Mtr Cargo room / hull mass. The higher the more suited the hull is for moving things around.

-rr How many light years could this ship travel starting with a full tank, and assuming it was either completely full, or empty the entire way. Does not take gravitonic capabilities, and a lot of other things into account.

-tt see -rt.

[2] 'Fish' as in 'load forward torpedo bays'.

-u Unpack the `player?.rst` file when present. It does so only for the specificly specified player. It thus will only work together with the -p option. -u can handle both planets and winplan files. However only the planets.exe files are extracted from the winplan *.rst file. This option is bigtargets proof.

-ut This generates the mdata¡pid¿.txt file. which is human readable and input for the scrore extractor script in the vga_scripts directory.

-xH This is including the weight of weaponry.

-xt See -rt.

–sanity One of the most beautiful options of the game. It was created to help prevent all those silly mistakes you can make when playing planets. What it all checks for is described in the following section.

3.2 Sanity

A description of what is and isn't checked by the sanity checking option.

Scanning ships for mine related friendly codes without proper mission

Friendly Codes 'mdh', 'mdq', 'mdN', and 'miN' require a 'Lay Mines' mission. Friendly Code 'msc' requires a 'mine sweep' mission. Ships with such friendly codes are checked to see if the related mission is also set. When this is not the case a warning is displayed.

Scanning bdm on ships without cash, or above our planets

Ships with Friendly Code 'bdm' are checked on two items. First is there any money to beam down? Second are they above a planet? When either of these questions answers "no" a warning is displayed, since you probably forgot to turn of the friendly code.

Scanning for ships without missions

Missions are too valuable to be running a ship without them. All ships without a mission are reported. Missions formulated as commands in your outgoing messages are not taken into account.

Scanning for ships with Friendly Code 'cln' on location, base presence, base tech, and resource availability

Checks to see if ships with a friendly code 'cln' are next to a planet with a base that has the required tech levels to perform the clone, and if the planet has sufficient resources to build the copy. Not taken into account are:

- Beam down resource missions from allied, or own, ships.
- Resources resulting from an 'unload freighter' mission of your star base.
- Tows on the ship to be cloned.
- If the to-be-cloned ship is not planning to move this turn.

Scanning ships for beam transfer codes without enemy ships or transferable items

- Ships with friendly code 'btm' are checked to for credits.
- Ships with friendly code 'btf' are checked for fighters, or fighter bays.
- Ships with friendly code 'btt' are checked for torpedoes, or torpedo bays.

All of the above ships are checked to see if there is at least one know ship of another race at the same location.

Scanning for double PBx codes, and ordering thereof

Checks if a 'PBx' friendly code is used by more than one planet with base, and also reports any skipped PBx codes. When for example two planets have a 'PBx' friendly code set. Say PB1, a warning will be reported PB1 has been used twice. Also a warning is issued if you skipped a number, for instance if you only used PB1 and PB3.

Scanning for ETA $>$ 1 turn

This reports all ships that will not arrive at their destination the next turn. The following items are taken into consideration:

- Ship's damage preventing set warp speed, taking into account an eventual Lizard bonus.

- Gravitonic accelerators for the three ships in the standard ship list.
- 'HYP' friendly code.
- Gravity wells pulling ships to planets.

Ship repairs and towing are not taken into account.

Scanning for lfm beaming up resources

Ships with friendly code 'lfm' are first checked to see if they are owned by the Robot, Rebel, or Colonial. For the rest of the warnings it is assumed you only want to use the 'lfm' code to process resources already on the ship, and want to reserve the resources remaining on the planet for other things. In order to validate that this policy is correctly implemented two checks are performed. First a check is performed whether at least one fighter can be build using the ship's cargo. Secondly a check is performed to see if the ship has space left to load a resources for at least one additional fighter, and resources for at least one are on the planet. Not taken into account are beam transfer missions, or an 'unload freighter' mission from an orbiting base.

Scanning for bases without build order or mission

Bases with a mission or a build order are reported.

Scanning for planets with d > 49 and no ATT or PB? set

All planets with more than 49 defence posts on the planet and no 'ATT' or 'PB?' friendly code are reported.

Scanning ships with friendly code 'mkt'

Ships with friendly code 'mkt' are reported when they have no torpedo bays, or lack the resources to build at least one torpedo.

Scanning special missions for planet/ship dislocation, own object targets, and fish/resource levels

This is based on the phost default mission.ini configuration. The following items are reported:

- Ships with with exploration mission.

- Ships with 'Lay Mines', 'Lay Minefield', or 'Lay Web Mine' mission without torpedo bays or sufficient torpedoes. For 'Lay mines' one torpedo is counted as sufficient.
- Ships with a colonise mission set.
- Ships with 'Beam Up Supp', 'Beam Up Fuel', 'Beam Up Dur', 'Beam Up Tri', or 'Beam Up Mol' which are not orbiting a planet.
- Ships with 'Build Torpedoes from Cargo' mission set that cannot build any torpedoes.
- Non-crystal ships with 'Lay Web Mines' mission, mission number 22.
- Ships with mission 'Scoop Torpedoes', but insufficient cargo space to scoop max torpedoes, or no torpedo bays. (The location of the minefield is not checked).
- Ships with mission 'Gather-Build Torpedoes' that are not above a planet, have no torpedo bays, have insufficient cargo space to build the requested torpedoes, or where the planet lacks the resources to build the requested number of torpedoes.
- Ships with mission 'Beam Down Credits' beaming down in deep space, or above one of our own planets, or lacking the credits specified.
- Ships with 'Transfer Torpedoes', or 'Transfer Fighters' mission that lack the specified number of fighter or torpedoes, or have no known ships from other races on the same location.
- Ships with mission 'Transfer Money' that lack the specified credits or where the destination ships is not present.
- Ships with mission 'Beam Up Credits', or 'Beam Up Clans' that are not above a planet from another race.
- Ships with mission 'Lay Mines In' without torpedo bays or sufficient torpedoes. (The location of the minefield is not checked).
- Ships with mission 'Lay Web Mines In' that have no torpedo bays or insufficient torpedoes, or that are not owned by the Crystals. (The location of the minefield is not checked).

Scanning for Friendly Code 'pop'

All ships with a friendly code 'pop' are listed. Popping is good, but just to make sure you don't pop one too many...

Scanning for Friendly Code 'NTP'

All ships with a friendly code 'NTP' are listed.

Scanning for fuel problems

Reports ships that will run out of fuel before they reach their destination at the given warp speed. Ships with a 'Beam Up Fuel', or 'Beam Up Multiple' mission which are above a planet are not reported, the assumption is made that they will beam up sufficient fuel to reach their destination. *Note: this is not fail-safe!* Intercepts are treated as any other destination. This means that only against the current location a check is performed.

Chapter 4

Configuration and command files

Stats uses two different types of file. Those that contain information on the layout of the file system and the host settings: the master configuration file. And command files for the individual players: the player command files.

4.1 Inner workings

Every line in the files can contain one or no start commands. A backslash can be used to mask end of lines to keep the file more readable. Commentary can be inserted in two ways. Firstly everything after a # is ignored until the end of the line. Secondly everything between '/*' and '*/' is ignored, including new lines. Whitespace can be used in any amount desired, except *within* keywords. After multiple character commands it is used as a separator. In that case at least one white space character is required.

None of the above will be much of a surprise. The sections below describe the commands and options supported by both the master configuration and the player command file.

filename The name of a file. '~', '_' and '/' accepted as well as all alpha numerical characters.

planetname The name of the planet as it appears in planets. You only need to use enough charaters to uniquely identify the planet. Ambiguity is reported, also when using alternative planet lists. The following characters are accepted: [:alnum:], space, quote, and hyphen.

exp numerical expression including one of the following operators. Mathematical ordering is applied to the expression evaluation. This means that multiplication takes place before adding up etc. For americans under the readers multiplication has presedence above devission.

$exp + exp$	add
$exp - exp$	subtract
$-exp$	multiply by -1
exp/exp	divide
$exp * exp$	multiply
(exp)	evaluate exp between braces first.
$exp \wedge exp$	raise first exp to the power of second exp
$\_ exp$	root

toggle Can be either of the following **yes, no, true, false, 1, 0**. Matching is case-insensitive.

float Decimals are separated by a dot.

4.2 Master configuration file

The master configuration file is parsed directly after the evaluation of the command line options. This implies that settings in the master command file overrule the command line options given.

The master configuration file contains the following start commands:

BaseDir	<filename>
PlayerDir	<filename>
Miningrate	<float,float,float,float,float,float, float,float,float,float,float>
CrystalsLikeDesert	<toggle>
ClimateLimitsPopulation	<toggle>
CrystalSinTempBehavior	<toggle>
MaxColTempSlope	<float>
NeutronicDecay	<float>
UseAccurateFuelModel	<toggle>
AllowEatingSupplies	<toggle>
AllowAdvancedRefinery	<toggle>
AllowAlchemy	<toggle>
RaceGrowthRate	<float,float,float,float,float,float, float,float,float,float,float>
GravityWellRange	<float>
NativeDeathRate	<float>
ClimateDeathRate	<float>
BorgAssimilationRate	<float>
OutputFile	<filename>
Unpack	

BaseDir Location of the planets directory.

PlayerDir Location of the rst and player data files.

Miningrate See phost documentation for its effect.

CrystalsLikeDesert See phost documentation for its effect.

ClimateLimitsPopulation See phost documentation for its effect.

CrystalSinTempBehavior See phost documentation for its effect.

MaxColTempSlope See phost documentation for its effect.

NeutronicDecay See phost documentation for its effect.

UseAccurateFuelModel See phost documentation for its effect.

AllowEatingSupplies See phost documentation for its effect.

AllowAdvancedRefinery See phost documentation for its effect.

AllowAlchemy See phost documentation for its effect.

RaceGrowthRate See phost documentation for its effect.

gravityWellRange See phost documentation for its effect. Square wells are assumed.

NativeDeathRate See phost documentation for its effect.

climateDeathRate See phost documentation for its effect.

BorgAssimilationRate See phost documentation for its effect.

OutputFile Redirects the non-error output to the file given.

Unpack Unpacks the rst files in the **playerDir** when present. Same as command line option -u.

4.3 Player command file

The player configuration file is parsed after the processing of the command line options for that player. Commands found in the command file fall into 3 execution categories.

Some are executed immidiatly after they are parsed. This applies unless stated otherwise below to all reports, and parameter settings. For example *colhappytarget*, *outputfile*, *report fleet*.

All other commands are executed after parsing the entire command file. They are then prioritized, and executed in the priority order. Commands of the same priority are executed in the order given in the command file. For further details see 1.1.

Something to take into consideration when playing multiple races.

- All report-start commands clear any reporting options given at the command line. This means the commandline reports will only be generated up until the first player with a report start command in its configuration file.

- The hull review report is generated at the end of the processing this means it will not be printed when it is ordered as a command line option and any of the read configuration files contains a report option.

4.3.1 Start commands

planet	<exp || planetname>:	<planet actions>
base	<exp || planetname>:	<turn> <base actions>
resourcemutation	<planetname>:	<turn> <resource actions>
report	<report type>	
tax		
setfc		
mdatatotxt		
colhappytarget	<exp>	
nathappytarget	<exp>	
colhappythreshold	<exp>	
nathappythreshold	<exp>	
wantplanetemptyinturns	<exp>	
outputfile	<filename>	
autofixatbases	<togle>	
autounloadatbases	<togle>	
warnonbuildfailure	<togle>	
autoloadmerlins	<togle>	

planet links actions to planets. The actions are executed in the order given (top to bottom, left to right). Multiple planet commands can be given for the same planet. The ordering of planets is irrelevant for the processing.

base actions to perform on base in future turn, see section 4.3.3 for further details.

report Request a report to be displayed. For possibilities see section 4.3.5.

tax Same as -ct. Set tax for all planets. This command is executed immediatly.

setfc Randomise non-special planetary friendly codes. Same as -cf. This command is executed immediatly.

mdatatotxt convert mdata.dat to a human readable mdata.txt file. Same as the 't' in -ut. This command is executed immediatly.

colhappytarget Set colonist happiness targets for taxation. Same as –T<Colonists>.

nathappytarget Set native happiness targets for taxation. Same as the [Natives] from –T<Colonists[,Natives]>.

colhappythreshold Same as colhappytarget.

nathappythreshold Same as nathappytarget.

wantplanetemptyinturns Modifier for the default behaviour of the -dM option. It sets the target number of turns to turn the planet ground stores into surface stores.

outputfile Redirects the non-error output to the file given.

autofixatbases For all starbases; The most damaged ship, if any, is ordered to be fixed at the starbase. Default is off.

autounloadatbases For all starbases; minerals, supplies, and megacredits are unloaded from all ships. Default is off.

warnonbuildfailure When a ship command fails insert addional (empty) resource mutation warning of the failure. Default is off.

autoloadmerlins Load as many supplies as possible into all merlins removing any other cargo as needed. Default is off.

4.3.2 Planet actions

The following planet commands are supported. These commands are case sensitive. Each letter is interpreted as a qualified command.

f <exp>
m <exp>
d <exp>
b <exp>
p <exp>
F <exp>
t <exp>

f build factories up unto the number specified.

m build mines up unto the number specified.

d build planetary defense posts up unto the number specified.

b build a starbase.

p build defense posts on the base up unto the number specified.

F build fighters in the base up unto the number specified.

t tax this planet. This only makes sence when no general tax command is in place.

4.3.3 Base actions

The base commands listed below are supported. They may appear in any order, and in any frequency. Actions listed first in the command file have presedence over actions listed later in case of resource contention.

engines	<exp>	<engine type>		
beams	<exp>	<beam type>		
tubes	<exp>	<torp type>		
hulls	<exp>	< hull type>		
ship	< hull type>	<engine type>	[[<beam type>]	<torp type>]
torpedoes	<exp>	< torp type>		
techlevel	<tech type>	<exp>		
fighters	<exp>			
mission	<base mission>			
defense	<exp>			

engines build **exp** number of engines of type **engine type** .

engine type This is either the slot number, or a keyword string. The following keywords, or unique start part thereof, are supported for engine type: stardrive1, stardrive2, stardrive3, superstardrive4, novadrive5, heavynovadrive6, quantamdrive7, hypderdrive8, transwarpdrive, w1, w2, w3, w4, w5, w6, w7, w8, w9, warp1, warp2, warp3, warp4, warp5, warp6, warp7, warp8, warp9.

beams build **exp** number of beams of type **beam type** .

beam type This is either the slot number, or a keyword string. The following keywords, or unique start part thereof, are supported for beam type: "laser", "xraylaser", "plasmabolt", "blaster", "positronbeam", "disruptor", "heavyblaster", "hblaster", "phaser", "heavydisruptor", "hdisruptor", "heavyphaser", "hphaser", "beam1", "beam2", "beam3", "beam4", "beam5", "beam6", "beam7", "beam8", "beam9", "beam10".

tubes build **exp** number of torpedoe tubes of type **torp type** .

torp type This is either the slot number, or a keyword string. The following keywords, or unique start part thereof, are supported for torp

type: The following keywords are supported for torp type: "mark1", "mk1", "protontorpedo", "mark2", "mk2", "gammabomb", "mark3", "mark4", "mark5", "mark6", "mark7", "mark8", "mk3", "mk4", "mk5", "mk6", "mk7", "mk8".

hulls build **exp** number of hulls of type **hull type** .

hull type This is either the slot number from the hullspec.dat file, or a keyword string. The keywords, or unique first part thereof, are formed as follows. Short ship types as described in 3.1 under the rt option, or the long ship type. Of the later the white space and special characters are removed. So T-REX CLASS BATTLESHIP becomes trexclassbattleship: Case is irrelevant in the matching.

ship set a build order for the specified ship at the starbase. Parts are build as needed. Techlevels are raised as needed. In case needed parts are already present these are used. When resources do not permit to build the ship the order is not set at the base, but parts that can be build are build.

torpedoes build **exp** number of torpedoes of type **torp type** .

tech Raise tech **tech type** to level **exp** .

tech type The following keywords are supported for tech type:

fighters Same as planet F options. Builds up to **exp** fighters

mission Set base mission to **base mission** .

base mission The following keywords are supported for base mission:

defense Same as planet p options. Builds up to **exp** starbase defense posts.

4.3.4 Resource actions

The resource actions listed below are supported. The intention is to allow you to specify miracles regarding resources. Like an ally dropping off some goods on your planet.

n <exp>
t <exp>
d <exp>
m <exp>
s <exp>
$ <exp>
c <exp>

exp can evaluate to a negative number. In that case resource are removed from the planet. When insufficient resources are available for removal those available are removed.

All the commands in this subsection are only applied to future turns. Not the current turn. This to avoid file trn file corruption.

n Create neutroneum.

t Create tritanium.

d Create duranium.

m Create molydenium.

s Create supplies.

$ Create credits.

c Create colonists.

4.3.5 Report types

summary	
fleet	Fleet report options
hull	Hull report options
base	Base report options
targets	Target report options
economy	Planet report options
buildqueue	planetname
levy	
nopcommands	

summary Same as -s.

fleet Same as the command line option -x.

fleet report options are letter codes. They are the same as for the command line option.

hull Same as the command line option -r.

hull report options are letter codes. They are the same as for the command line option.

base Same as the command line option -b.

base report options are letter codes. They are the same as for the command line option.

targets Same as the command line option -t.

target report options are letter codes. They are the same as for the command line option.

economy Same as the command line option -d.

economy report options are letter codes. They are the same as for the command line option.

buildqueue Displays the buildqueue for the given planet. This is a universal command.

levy Same as the command line option -l.

nopcommands Reports the planets owned by the player which do not have commands in the command list. This is a universal command.