

# Guide for trident v0.29.0

## Contents

<b>1</b>	<b>Poseidon package repositories</b>	<b>1</b>
<b>2</b>	<b>Analysing your own dataset outside of the main repository</b>	<b>2</b>
<b>3</b>	<b>Package creation and manipulation commands</b>	<b>3</b>
3.1	Init command . . . . .	3
3.2	Fetch command . . . . .	4
3.3	Forge command . . . . .	5
3.3.1	The forge selection language . . . . .	8
3.3.2	Other options . . . . .	8
3.4	Genoconvert command . . . . .	9
3.5	Update command . . . . .	10
<b>4</b>	<b>Inspection commands</b>	<b>12</b>
4.1	List command . . . . .	12
4.2	Summarise command . . . . .	13
4.3	Survey command . . . . .	14
4.4	Validate command . . . . .	14

## 1 Poseidon package repositories

Trident generally requires Poseidon “packages” to work with (since version 0.28.0 it also supports direct interaction with “unpackaged” genotype data – see `-p` below). Most trident subcommands therefore have a central parameter, called `--baseDir` or simply `-d` to specify one or more base directories to look for packages. For example, if all Poseidon packages live inside a repository at `/path/to/poseidon/packages` you would simply say `trident <subcommand> -d /path/to/poseidon/dirs/` and trident would automatically search all subdirectories inside of the repository for valid poseidon packages (as identified by valid `POSEIDON.yml` files).

You can arrange a poseidon repository in a hierarchical way. For example:

```
/path/to/poseidon/packages
  /modern
    /2019_poseidon_package1
    /2019_poseidon_package2
  /ancient
  /...
```

```

33     /...
34 /Reference_Genomes
35     /...
36     /...
37 /Archaic_Humans
38     /...
39     /...

```

40 You can use this structure to select only the level of packages you're interested in, and you can make use of the  
 41 fact that `-d` can be given multiple times.

42 Let's use the `list` command to list all packages in the `modern` and `Reference_Genomes`:

```

43 trident list -d /path/to/poseidon/packages/modern \
44 -d /path/to/poseidon/packages/ReferenceGenomes --packages

```

## 45 2 Analysing your own dataset outside of the main repository

46 Being able to specify one or multiple repositories is often not enough, as you may have your own data to  
 47 co-analyse with the main repository. This is easy to do, as you simply need to provide your own genotype data  
 48 as yet another poseidon package to be added to your `trident list` command. For example, let's say you have  
 49 genotype data in `EIGENSTRAT` format (`trident` supports `EIGENSTRAT` and `PLINK` as formats.):

```

50 ~/my_project/my_project.geno
51 ~/my_project/my_project.snp
52 ~/my_project/my_project.ind

```

53 then you can make that to a skeleton Poseidon package with the `init` command. You can also do it manually by  
 54 simply adding a `POSEIDON.yml` file, with for example the following content:

```

55 poseidonVersion: 2.5.0
56 title: My_awesome_project
57 description: Unpublished genetic data from my awesome project
58 contributor:
59   - name: Stephan Schiffels
60     email: schiffels@institute.org
61 packageVersion: 0.1.0
62 lastModified: 2020-10-07
63 genotypeData:
64   format: EIGENSTRAT
65   genoFile: my_project.geno
66   snpFile: my_project.snp
67   indFile: my_project.ind
68 jannoFile: my_project.janno
69 bibFile: sources.bib

```

70 Two remarks: 1) all file paths are considered *relative* to the directory in which `POSEIDON.yml` resides. Here I  
 71 assume that you put this file into the same directory as the three genotype files. 2) Besides the genotype data  
 72 files there are two (technically optional) files referenced by this example `POSEIDON.yml` file: `sources.bib` and

73 my\_project.janno. Of course you can add them manually - `init` automatically creates empty dummy versions.

74 Once you have set up your own “Poseidon” package (which is really only a skeleton so far), you can add it to

75 your `trident` analysis, by simply adding your project directory to the command using `-d`:

```
76 trident list -d /path/to/poseidon/packages/modern \
77     -d /path/to/poseidon/packages/ReferenceGenomes
78     -d ~/my_project --packages
```

## 79 3 Package creation and manipulation commands

### 80 3.1 Init command

81 `init` creates a new, valid poseidon package from genotype data files. It adds a valid `POSEIDON.yml` file, a dummy

82 `.janno` file for context information and an empty `.bib` file for literature references.

83 [Click here for command line details](#)

```
84 Usage: trident init ((-p|--genoOne ARG) | (-r|--inFormat ARG)
85                   (-g|--genoFile ARG) (-s|--snpFile ARG) (-i|--indFile ARG))
86                   [--snpSet ARG] (-o|--outPackagePath ARG)
87                   [-n|--outPackageName ARG] [--minimal]
```

88 Create a new Poseidon package from genotype data

89 Available options:

91	<code>-h,--help</code>	Show this help text
92	<code>-p,--genoOne ARG</code>	one of the input genotype data files. Expects <code>.bed</code> or
93		<code>.bim</code> or <code>.fam</code> for PLINK and <code>.geno</code> or <code>.snp</code> or <code>.ind</code> for
94		EIGENSTRAT. The other files must be in the same
95		directory and must have the same base name
96	<code>-r,--inFormat ARG</code>	the format of the input genotype data: EIGENSTRAT or
97		PLINK
98	<code>-g,--genoFile ARG</code>	the input geno file path
99	<code>-s,--snpFile ARG</code>	the input snp file path
100	<code>-i,--indFile ARG</code>	the input ind file path
101	<code>--snpSet ARG</code>	the snpSet of the new package: 1240K, HumanOrigins or
102		Other. Default: Other
103	<code>-o,--outPackagePath ARG</code>	the output package directory path
104	<code>-n,--outPackageName ARG</code>	the output package name - this is optional: If no
105		name is provided, then the package name defaults to
106		the basename of the (mandatory) <code>--outPackagePath</code>
107		argument
108	<code>--minimal</code>	should only a minimal output package be created?

109 The command

```
110 trident init \
111     -r EIGENSTRAT/PLINK \
112     -g path/to/geno_file \
```

```

113 -s path/to/snp_file \
114 -i path/to/ind_file \
115 --snpSet 1240K|HumanOrigins|Other \
116 -o path/to/new_package_name

```

117 requires the format `-r` (`--inFormat`) of your input data (either EIGENSTRAT or PLINK), the paths to the  
118 respective files in `-g` (`--genoFile`), `-s` (`--snpFile`), and `-i` (`--indFile`), and optionally the “shape” of these  
119 files (`--snpSet`), so if they cover the 1240K, the HumanOrigins or an Other SNP set. A simpler interface added  
120 in trident 0.29.0 is available with `-p` (`+ --snpSet`).

	EIGENSTRAT	PLINK
genoFile	.geno	.bed
snpFile	.snp	.bim
indFile	.ind	.fam

121 The output package of `init` is created as a new directory `-o`, which should not already exist, and gets the  
122 package `title` corresponding to the basename of `-o`. You can also set the title explicitly with `-n`. The `--minimal`  
123 flag causes `init` to create a minimal package with a very basic POSEIDON.yml and no `.bib` and `.janno` files.

## 124 3.2 Fetch command

125 `fetch` allows to download poseidon packages from a remote poseidon server.

126 [Click here for command line details](#)

```

127 Usage: trident fetch (-d|--baseDir DIR) [-f|--fetchString ARG] [--fetchFile ARG]
128                [--remoteURL ARG] [-u|--upgrade] [--downloadAll]

```

129 Download data from a remote Poseidon repository

130 Available options:

132 <code>-h,--help</code>	Show this help text
133 <code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
134 <code>-f,--fetchString ARG</code>	List of packages to be downloaded from the remote server. Package names should be wrapped in asterisks: *package_title*. You can combine multiple values with comma, so for example: "*package_1*, *package_2*, *package_3*". <code>fetchString</code> uses the same parser as <code>forgeString</code> , but does not allow excludes. If groups or individuals are specified, then packages which include these groups or individuals are included in the download.
144 <code>--fetchFile ARG</code>	A file with a list of packages. Works just as <code>-f</code> , but multiple values can also be separated by newline, not just by comma. <code>-f</code> and <code>--fetchFile</code> can be combined.
147 <code>--remoteURL ARG</code>	URL of the remote Poseidon server (default: "https://c107-224.cloud.gwdg.de")

```

149  -u,--upgrade          overwrite outdated local package versions
150  --downloadAll         download all packages the server is offering

```

151 It works with

```

152 trident fetch -d ... -d ... \
153   -f "*package_title_1*,*package_title_2*,*package_title_3*,group_name,<Individual1>" \
154   --fetchFile path/to/forgeFile

```

155 and the entities you want to download must be listed either in a simple string with comma-separated values  
 156 (-f/--fetchString) or in a text file (--fetchFile). Entities are specified using a special syntax: Package titles  
 157 are wrapped in asterisks: *package\_title* (see also the documentation of **forge** below), group names are spelled  
 158 as is, and individual names are wrapped in angular brackets, like <Individual1>. Fetch will figure out which  
 159 packages need to be downloaded to include all specified entities. --downloadAll causes fetch to ignore -f and  
 160 download all packages from the server. The downloaded packages are added in the first (!) -d directory, but  
 161 downloads are only performed if the respective packages are not already present in an up-to-date version in any  
 162 of the -d dirs.

163 Note that **trident fetch** makes most sense in combination with **trident list --remote**: First one can inspect  
 164 what is available on the server, then one can create a custom fetch command.

165 **fetch** also has the optional arguments --remote https://..." to name an alternative poseidon server. The  
 166 default points to the [DAG server](#).

167 To overwrite outdated package versions with **fetch**, the -u/--upgrade flag has to be set. Note that many file  
 168 systems do not offer a way to recover overwritten files. So be careful with this switch.

### 169 3.3 Forge command

170 **forge** creates new poseidon packages by extracting and merging packages, populations and individuals from  
 171 your poseidon repositories.

172 Click here for command line details

```

173 Usage: trident forge [-d|--baseDir DIR]
174          [
175          ((-p|--genoOne ARG) | (-r|--inFormat ARG)
176          (-g|--genoFile ARG) (-s|--snpFile ARG)
177          (-i|--indFile ARG)) [--snpSet ARG]]
178          [--forgeFile ARG | (-f|--forgeString ARG)]
179          [--selectSnps ARG] [--intersect] [--outFormat ARG]
180          [--minimal] [--onlyGeno] (-o|--outPackagePath ARG)
181          [-n|--outPackageName ARG] [-w|--warnings] [--no-extract]
182  Select packages, groups or individuals and create a new Poseidon package from
183  them

```

184  
 185 Available options:

```

186  -h,--help          Show this help text
187  -d,--baseDir DIR   a base directory to search for Poseidon Packages
188                     (could be a Poseidon repository)
189  -p,--genoOne ARG   one of the input genotype data files. Expects .bed or

```

190 .bim or .fam for PLINK and .geno or .snp or .ind for  
191 EIGENSTRAT. The other files must be in the same  
192 directory and must have the same base name  
193 -r,--inFormat ARG the format of the input genotype data: EIGENSTRAT or  
194 PLINK  
195 -g,--genoFile ARG the input geno file path  
196 -s,--snpFile ARG the input snp file path  
197 -i,--indFile ARG the input ind file path  
198 --snpSet ARG the snpSet of the new package: 1240K, HumanOrigins or  
199 Other. Default: Other  
200 --forgeFile ARG A file with a list of packages, groups or individual  
201 samples. Works just as -f, but multiple values can  
202 also be separated by newline, not just by comma.  
203 Empty lines are ignored and comments start with "#",  
204 so everything after "#" is ignored in one line.  
205 -f,--forgeString ARG List of packages, groups or individual samples to be  
206 combined in the output package. Packages follow the  
207 syntax \*package\_title\*, populations/groups are simply  
208 group\_id and individuals <individual\_id>. You can  
209 combine multiple values with comma, so for example:  
210 "\*package\_1\*, <individual\_1>, <individual\_2>,"  
211 group\_1". Duplicates are treated as one entry.  
212 Negative selection is possible by prepending "-" to  
213 the entity you want to exclude (e.g. "\*package\_1\*,"  
214 -<individual\_1>, -group\_1"). forge will apply  
215 excludes and includes in order. If the first entity  
216 is negative, then forge will assume you want to merge  
217 all individuals in the packages found in the baseDirs  
218 (except the ones explicitly excluded) before the  
219 exclude entities are applied. An empty forgeString  
220 will therefore merge all available individuals.  
221 --selectSnps ARG To extract specific SNPs during this forge operation,  
222 provide a Snp file. Can be either Eigenstrat (file  
223 ending must be '.snp') or Plink (file ending must be  
224 '.bim'). When this option is set, the output package  
225 will have exactly the SNPs listed in this file. Any  
226 SNP not listed in the file will be excluded. If  
227 option '--intersect' is also set, only the SNPs  
228 overlapping between the SNP file and the forged  
229 packages are output.  
230 --intersect Whether to output the intersection of the genotype  
231 files to be forged. The default (if this option is  
232 not set) is to output the union of all SNPs, with  
233 genotypes defined as missing in those packages which  
234 do not have a SNP that is present in another package.

235                               With this option set, the forged dataset will  
 236                               typically have fewer SNPs, but less missingness.  
 237   --outFormat ARG           the format of the output genotype data: EIGENSTRAT or  
 238                               PLINK. Default: PLINK  
 239   --minimal                 should only a minimal output package be created?  
 240   --onlyGeno                should only the resulting genotype data be returned?  
 241                               This means the output will not be a Poseidon package  
 242   -o,--outPackagePath ARG   the output package directory path  
 243   -n,--outPackageName ARG   the output package name - this is optional: If no  
 244                               name is provided, then the package name defaults to  
 245                               the basename of the (mandatory) --outPackagePath  
 246                               argument  
 247   -w,--warnings             Show all warnings for merging genotype data  
 248   --no-extract              Skip the selection step in forge. This will result in  
 249                               outputting all individuals in the relevant packages,  
 250                               and hence a superset of the requested  
 251                               individuals/groups. It may result in better  
 252                               performance in cases where one wants to forge entire  
 253                               packages or almost entire packages. Note that this  
 254                               will also ignore any ordering in the output  
 255                               groups/individuals. With this option active,  
 256                               individuals from the relevant packages will just be  
 257                               written in the order that they appear in the original  
 258                               packages.

259   forge can be used with

```
260 trident forge -d ... -d ... \  
261   -f "*package_name*, group_id, <individual_id>" \  
262   --forgeFile path/to/forgeFile \  
263   -o path/to/new_package_name
```

264   where the entities (packages, groups/populations, individuals/samples) you want in the output package can  
 265   be denoted either as as simple string with comma-separated values (-f/--forgeString) or in a text file  
 266   (--forgeFile).

267   Including one or multiple Poseidon packages with -d is not the only way to include data for a forge operation. It  
 268   is also possible to include unpackaged genotype data directly with -r + -g + -s + -i (+ --snpSet) or -p  
 269   (+ --snpSet). This makes the following example possible, where we merge data from one Poseidon package  
 270   and two genotype datasets.

```
271 trident forge \  
272   -d 2017_GonzalesFortesCurrentBiology \  
273   -r PLINK \  
274   -g 2017_HaberAJHG/2017_HaberAJHG.bed \  
275   -s 2017_HaberAJHG/2017_HaberAJHG.bim \  
276   -i 2017_HaberAJHG/2017_HaberAJHG.fam \  
277   -r PLINK \  
278
```

```

278 -g 2018_VeeramahPNAS/2018_VeeramahPNAS.bed \
279 -i 2018_VeeramahPNAS/2018_VeeramahPNAS.fam \
280 -s 2018_VeeramahPNAS/2018_VeeramahPNAS.bim \
281 -f "<STR241.SG>,<ERS1790729.SG>,Iberia_HG.SG" \
282 -o testpackage \
283 --onlyGeno

```

### 284 3.3.1 The forge selection language

285 Entities in the `--forgeString` or the `--forgeFile` have to be marked in a certain way:

- 286 • Each package is surrounded by `*`, so if you want all individuals of `2019_Jeong_InnerEurasia` in the
- 287 output package you would add `*2019_Jeong_InnerEurasia*` to the list.
- 288 • Groups/populations are not specially marked. So to get all individuals of the group `Swiss_Roman_period`,
- 289 you would simply add `Swiss_Roman_period`.
- 290 • Individuals/samples are surrounded by `<` and `>`, so `ALA026` becomes `<ALA026>`.

291 Do not forget to wrap the `forgeString` in quotes.

292 You can either use `-f/--forgeString` or `--forgeFile`. In the file each line is treated as a separate `forgeString`,  
 293 empty lines are ignored and `#`s start comments. So this is a valid `forgeFile`:

```

294 # Packages
295 *package1*, *package2*
296
297 # Groups and individuals from other packages beyond package1 and package2
298 group1, <individual1>, group2, <individual2>, <individual3>
299
300 # group2 has two outlier individuals that should be ignored
301 -<bad_individual1> # This one has very low coverage
302 -<bad_individual2> # This one is from a different time period

```

303 By prepending `-` to the bad individuals, we can exclude them from the forged package. `forge` fig-  
 304 ures out the final list of samples to include by executing all `forge`-entities in order. So an entity list  
 305 `*PackageA*, -<Individual1>, GroupA` may result in a different outcome than `*PackageA*, GroupA, -<Individual1>`,  
 306 depending on whether `<Individual1>` belongs to `GroupA` or not. If the `forge` entity list starts with a negative  
 307 entity, or if the entity list is empty, `forge` will implicitly assume you want to include all individuals in all  
 308 packages found in the `baseDirs` (except the ones explicitly excluded, of course). An empty `forgeString` will  
 309 therefore merge all available individuals.

### 310 3.3.2 Other options

311 Just as for `init` the output package of `forge` is created as a new directory `-o`. The title can also be explicitly  
 312 defined with `-n`.

313 `--minimal` allows for the creation of a minimal output package without `.bib` and `.janno`. This might be  
 314 especially useful for data analysis pipelines, where only the genotype data is required. Even more basic output  
 315 comes with `--onlyGeno`, which means that only the genotype data is returned without any Poseidon package.

316 `forge` has a an optional flag `--intersect`, that defines, if the genotype data from different packages should  
 317 be merged with an **union** or an **intersect** operation. The default (if this option is not set) is to output the



union of all SNPs, with genotypes defined as missing in samples from packages which do not have a SNP that is present in another package. With this option set, on the other hand, the forged dataset will typically have fewer SNPs, but less missingness.

**--intersect** also influences the automatic determination of the **snpSet** field in the POSEIDON.yml file for the resulting package. If the **snpSets** of all input packages are identical, then the resulting package will just inherit this configuration. Otherwise **forge** applies the following pairwise merging logic:

Input snpSet A	Input snpSet B	--intersect	Ouput snpSet
Other	*	*	Other
1240K	HumanOrigins	True	HumanOrigins
1240K	HumanOrigins	False	1240K

**--selectSnps** allows to provide **forge** with a SNP file in EIGENSTRAT (.snp) or PLINK (.bim) format to create a package with a specific selection. When this option is set, the output package will have exactly the SNPs listed in this file. Any SNP not listed in the file will be excluded. If **--intersect** is also set, only the SNPs overlapping between the SNP file and the forged packages are output.

Merging genotype data across different data sources and file formats is tricky. **forge** is more verbose about potential issues, if the **-w/--warnings** flag is set.

### 3.4 Genoconvert command

**genoconvert** converts the genotype data in a Poseidon package to a different file format. The respective entries in the POSEIDON.yml file are changed accordingly.

Click here for command line details

```
Usage: trident genoconvert [-d|--baseDir DIR]
[
    ((-p|--genoOne ARG) | (-r|--inFormat ARG)
    (-g|--genoFile ARG) (-s|--snpFile ARG)
    (-i|--indFile ARG)) [--snpSet ARG]]
--outFormat ARG [--onlyGeno]
[-o|--outPackagePath ARG] [--removeOld]
Convert the genotype data in a Poseidon package to a different file format
```

Available options:

<b>-h,--help</b>	Show this help text
<b>-d,--baseDir DIR</b>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
<b>-p,--genoOne ARG</b>	one of the input genotype data files. Expects .bed or .bim or .fam for PLINK and .geno or .snp or .ind for EIGENSTRAT. The other files must be in the same directory and must have the same base name
<b>-r,--inFormat ARG</b>	the format of the input genotype data: EIGENSTRAT or PLINK
<b>-g,--genoFile ARG</b>	the input geno file path

```

354 -s,--snpFile ARG          the input snp file path
355 -i,--indFile ARG          the input ind file path
356 --snpSet ARG              the snpSet of the new package: 1240K, HumanOrigins or
357                             Other. Default: Other
358 --outFormat ARG           the format of the output genotype data: EIGENSTRAT or
359                             PLINK.
360 --onlyGeno                should only the resulting genotype data be returned?
361                             This means the output will not be a Poseidon package
362 -o,--outPackagePath ARG    the output package directory path - this is optional:
363                             If no path is provided, then the output is written to
364                             the directories where the input genotype data file
365                             (.bed/.geno) is stored
366 --removeOld               Remove the old genotype files when creating the new
367                             ones

```

368 With the default setting

```

369 trident genoconvert -d ... -d ... --outFormat EIGENSTRAT|PLINK

```

370 all packages in -d will be converted to the desired --outFormat (either EIGENSTRAT or PLINK), if the data is  
371 not already in this format. This includes updating the respective POSEIDON.yml files.

372 The “old” data is not deleted, but kept around. That means conversion can result in a package with both PLINK  
373 and EIGENSTRAT data, but only one is linked in the POSEIDON.yml file, and that is what will be used by  
374 trident. To delete the old data in the conversion you can add the --removeOld flag.

375 Instead of -d to change Poseidon packages, the combination -r + -g + -s + -i (+ --snpSet) or -p (+  
376 --snpSet) allows to directly convert genotype data that is not wrapped in a Poseidon package and store it to a  
377 directory given in -o. See this example:

```

378 trident genoconvert \
379     -p 2018_Mittnik_Baltic/Mittnik_Baltic.bed \
380     --outFormat EIGENSTRAT
381     -o my_directory

```

### 382 3.5 Update command

383 update automatically updates POSEIDON.yml files of one or multiple packages if the packages were changed.

384 [Click here for command line details](#)

```

385 Usage: trident update (-d|--baseDir DIR) [--poseidonVersion ARG]
386                             [--ignorePoseidonVersion] [--versionComponent ARG]
387                             [--noChecksumUpdate] [--newContributors ARG]
388                             [--logText ARG] [--force]
389     Update POSEIDON.yml files automatically
390

```

391 Available options:

```

392 -h,--help                Show this help text
393 -d,--baseDir DIR          a base directory to search for Poseidon Packages
394                             (could be a Poseidon repository)

```

```

395 --poseidonVersion ARG    Poseidon version the packages should be updated to:
396                          e.g. "2.5.3" (default: Nothing)
397 --ignorePoseidonVersion  Read packages even if their poseidonVersion is not
398                          compatible with the trident version. The assumption
399                          is, that the package is already structurally adjusted
400                          to the trident version and only the version number is
401                          lagging behind.
402 --versionComponent ARG   Part of the package version number in the
403                          POSEIDON.yml file that should be updated: Major,
404                          Minor or Patch (see
405                          https://semver.org) (default: Patch)
406 --noChecksumUpdate       Should update of checksums in the POSEIDON.yml file
407                          be skipped
408 --ignoreGeno             ignore SNP and GenoFile
409 --newContributors ARG    Contributors to add to the POSEIDON.yml file in the
410                          form "[Firstname Lastname](Email address);..."
411 --logText ARG            Log text for this version jump in the CHANGELOG
412                          file (default: "not specified")
413 --force                  Normally the POSEIDON.yml files are only changed if
414                          the poseidonVersion is adjusted or any of the
415                          checksums change. With --force a package version
416                          update can be triggered even if this is not the case.

```

417 It can be called with a lot of optional arguments

```

418 trident update -d ... -d ... \
419   --poseidonVersion "X.X.X" \
420   --versionComponent Major/Minor/Patch \
421   --noChecksumUpdate
422   --ignoreGeno
423   --newContributors "[Firstname Lastname](Email address);..."
424   --logText "short description of the update"
425   --force

```

426 By default `update` will not edit a package's POSEIDON.yml file, even when arguments like `--versionComponent`,  
427 `--newContributors` or `--logText` are explicitly set. This default exists to run the function on a large set of  
428 packages where only few of them were edited and need an active update. A package will only be modified by  
429 `update` if either

- 430 • any of the files with checksums (e.g. the genotype data) in it were modified,
- 431 • the `--poseidonVersion` argument differs from the `poseidonVersion` in the package's POSEIDON.yml  
432 file
- 433 • or the `--force` flag was set in `update`.

434 If any of these applies to a package in the search directory (`--baseDir/-d`), it will be updated. This includes  
435 the following steps:

- 436 • If `--poseidonVersion` is different from the `poseidonVersion` field in the package, then that will be  
437 updated.

- The `packageVersion` will be incremented. If `--versionComponent` is not set, then it falls back to `Patch`, so a change in the last position of the three digit version number. `Minor` increments the middle, and `Major` the first position (see [semantic versioning](#)).
- The `lastModified` field will be updated to the current day (based on your computer's system time).
- The contributors in `--newContributors` will be added to the `contributor` field if they're not there already.
- If any checksums changed, then they will be updated. If certain checksums are not set yet, then they will be added. The checksum update can be skipped with `--noChecksumUpdate` or partially skipped for the genotype data with `--ignoreGeno`.
- The `CHANGELOG.md` file will be updated with a new row for the new version and the text in `--logText` (default: "not specified"), which will be appended as the first line of the file. If no `CHANGELOG.md` file exists, then it will be created and referenced in the `POSEIDON.yml` file.

`:heavy_exclamation_mark:` As `update` reads and rewrites `POSEIDON.yml` files, it may change their inner order, layout or even content (e.g. if they have fields which are not in the [Poseidon package definition](#)). Create a backup of the `POSEIDON.yml` file before running `update` if you are uncertain.

## 4 Inspection commands

### 4.1 List command

`list` lists packages, groups and individuals of the datasets you use, or of the packages available on the server.

Click here for command line details

```
Usage: trident list ((-d|--baseDir DIR) | --remote [--remoteURL ARG])
                (--packages | --groups | --individuals
                [-j|--jannoColumn JANNO_HEADER]) [--raw]
```

List packages, groups or individuals from local or remote Poseidon repositories

Available options:

<code>-h,--help</code>	Show this help text
<code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
<code>--remote</code>	list packages from a remote server instead the local file system
<code>--remoteURL ARG</code>	URL of the remote Poseidon server (default: "https://c107-224.cloud.gwdg.de")
<code>--packages</code>	list all packages
<code>--groups</code>	list all groups, ignoring any group names after the first as specified in the Janno-file
<code>--individuals</code>	list individuals
<code>-j,--jannoColumn JANNO_HEADER</code>	list additional fields from the janno files, using the Janno column heading name, such as Country, Site, Date_C14_Uncal_BP, Endogenous, ...
<code>--raw</code>	output table as tsv without header. Useful for piping

479                                   into grep or awk  
 480    --ignoreGeno                   ignore SNP and GenoFile  
 481 To list packages from your local repositories, as seen above you can run  
 482 `trident list -d ... -d ... --packages`  
 483 This will yield a table like this

```

484 .-----
485 |                Title                |    Date    | Nr Individuals |
486 :=====
487 | 2015_1000Genomes_1240K_haploid_pull | 2020-08-10 | 2535           |
488 | 2016_Mallick_SGDP1240K_diploid_pull | 2020-08-10 | 280            |
489 | 2018_BostonDatashare_modern_publish | 2020-08-10 | 2772           |
490 | ...                                | ...        |                |
491 '-----'

```

492 so a nicely formatted table of all packages, their last update and the number of individuals in it.  
 493 To view packages on the remote server, instead of using directories to specify the locations of repositories on  
 494 your system, you can use `--remote` to show packages on the remote server. For example  
 495 `trident list --packages --remote`  
 496 will result in a view of all published packages in our public online repository.  
 497 You can also list groups, as defined in the third column of EIGENSTRAT `.ind` files (or the first column of a  
 498 PLINK `.fam` file), and individuals:

```

499 trident list -d ... -d ... --groups
500 trident list -d ... -d ... --individuals

```

501 The `--individuals` flag also provides a way to immediately access information from the `.janno` files on the  
 502 command line. This works with the `-j/--jannoColumn` option. For example adding `--jannoColumn Country`  
 503 `--jannoColumn Date_C14_Uncal_BP` to the commands above will add the `Country` and the `Date_C14_Uncal_BP`  
 504 columns to the respective output tables.  
 505 Note that if you want a less fancy table, for example because you want to load this into Excel, or pipe into  
 506 another command that cannot deal with the neat table layout, you can use the `--raw` option to output that  
 507 table as a simple tab-delimited stream.

## 508 4.2 Summarise command

509 `summarise` prints some general summary statistics for a given poseidon dataset taken from the `.janno` files.  
 510 [Click here for command line details](#)

```

511 Usage: trident summarise (-d|--baseDir DIR) [--raw]
512   Get an overview over the content of one or multiple Poseidon packages
513
514 Available options:
515   -h,--help           Show this help text
516   -d,--baseDir DIR    a base directory to search for Poseidon Packages

```

517 (could be a Poseidon repository)  
 518 `--raw` output table as tsv without header. Useful for piping  
 519 into grep or awk

520 You can run it with

521 `trident summarise -d ... -d ...`

522 which will show you context information like – among others – the number of individuals in the dataset, their  
 523 sex distribution, the mean age of the samples (for ancient data) or the mean coverage on the 1240K SNP array  
 524 in a table. `summarise` depends on complete .janno files and will silently ignore missing information for some  
 525 statistics.

526 You can use the `--raw` option to output the summary table in a simple, tab-delimited layout.

### 527 4.3 Survey command

528 `survey` tries to indicate package completeness (mostly focused on .janno files) for poseidon datasets.

529 [Click here for command line details](#)

530 Usage: `trident survey (-d|--baseDir DIR) [--raw]`

531 Survey the degree of context information completeness for Poseidon packages

532

533 Available options:

534 `-h,--help` Show this help text

535 `-d,--baseDir DIR` a base directory to search for Poseidon Packages  
 536 (could be a Poseidon repository)

537 `--raw` output table as tsv without header. Useful for piping  
 538 into grep or awk

539 Running

540 `trident survey -d ... -d ...`

541 will yield a table with one row for each package. See `trident survey -h` for a legend which cell of this table  
 542 means what.

543 Again you can use the `--raw` option to output the survey table in a tab-delimited format.

### 544 4.4 Validate command

545 `validate` checks poseidon datasets for structural correctness.

546 [Click here for command line details](#)

547 Usage: `trident validate (-d|--baseDir DIR) [--verbose]`

548 Check one or multiple Poseidon packages for structural correctness

549

550 Available options:

551 `-h,--help` Show this help text

552 `-d,--baseDir DIR` a base directory to search for Poseidon Packages  
 553 (could be a Poseidon repository)

```

554  --verbose          print more output to the command line
555  --ignoreGeno       ignore SNP and GenoFile
556  --noExitCode       do not produce an explicit exit code
557  You can run it with
558  trident validate -d ... -d ...
559  and it will either report a success (Validation passed) or failure with specific error messages to simplify fixing
560  the issues.
561  validate tries to ensure that each package in the dataset adheres to the schema definition. Here is a list of
562  what is checked:
563      • Presence of the necessary files
564      • Full structural correctness of .bib and .janno file
565      • Superficial correctness of genotype data files. A full check would be too computationally expensive
566      • Correspondence of BibTeX keys in .bib and .janno
567      • Correspondence of individual and group IDs in .janno and genotype data files
568  In fact much of this validation already runs as part of the general package reading pipeline invoked for many
569  trident subcommands (e.g. forge). validate is meant to be more thorough, though, and will explicitly fail if
570  even a single package is broken.

```