Chapter 25: Generic Command Option Descriptions

This chapter provides an alphabetical listing of **UPS/UPD** options with generic descriptions. More detailed information on a few selected options can be found at the end of the chapter.

In the command reference chapters, Chapter 23: *UPS Command Reference* and Chapter 24: *UPD/UPP Command Reference*, the options supported by each command are listed with command-specific descriptions.

25.1 Alphabetical Option Listing

Table 25.1.0-a:

`-?`	Prints command description and option usage information to screen Not valid with other options. Note for C shell users: enclose **`-?`** in double quotes (e.g., **`ups declare "-?"`**); **`-?`** is interpreted by **sh**.
`-a`	Operates on all instances that match the other options given on command line
`-A <nodeList>`	Specifies authorized nodes. When declaring an instance, sets the AUTHORIZED_NODES keyword. By default, products can be used from any node which has access to the database; use this option to restrict usage to a limited set of nodes. Note: can also be used inside **`-G ""`**
`-b <compileFile>`	Specifies name of the output file for the **`ups compile`** command (COMPILE_FILE keyword) associated with the COMPILE action (see Chapter 38: *Use of Compile Scripts in Table Files*); `.sh` or `.csh` gets added automatically (the file's path is specified with **`-u`**)

Table 25.1.0-a:

-B <depProdName>= "<options>"	Specifies options to prepend to the `setupRequired` line (in table file) for the dependent product **<depProdName>**
-c	Specifies "current" chain Note: can also be used inside **-G ""**
-C	For **ups declare** when initially declaring a product: Prevents execution of the CONFIGURE action For **ups declare** when declaring a chain: Prevents execution of the corresponding chain action For **ups undeclare** when removing a product: Prevents execution of the UNCONFIGURE action For **ups undeclare** when removing a chain: Prevents execution of the corresponding "unchain" action
-d	Specifies "development" chain Note: can also be used inside **-G ""**
-D "<origin>"	Specifies the product's master source file (This becomes the value of the keyword ORIGIN after any spaces are removed.) Note: can also be used inside **-G ""**
-e	Sets $UPS_EXTENDED (to the value **1**). See section 25.2.1 *-e* for more information.
-f <flavorList>	Specifies flavor (operating system[+release]). Multiple values can usually be specified; several **UPD** commands operate on each listed flavor, however most **UPS** commands ignore all except first in list. See also **-H** and the number options **-0**, **-1**, **-2**, and **-3**. Notes: can also be used inside **-G ""**; not valid with options **-0**, **-1**, **-2**, or **-3**
-F	Used only with **ups get**; prints to screen a list of files that are associated with the product but which are maintained external to the products area (excluding table file)
-g <chainName>	Specifies chain using the chain name (either a standard or user-defined chain name can be used here). Multiple values can usually be specified. Note: can also be used inside **-G ""**

Table 25.1.0-a:

-G "<options>"	Used by commands that copy or install a product instance (namely **ups copy**, **upd cloneproduct**, and **upd install**), the **-G** option provides the means to specify target instance identifiers to pass to the internally-run **ups declare** command. The elements valid for use with **-G** include **<product>**, **<version>** and the following subset of the **ups declare** options: **-A <nodeList>**, **-c**, **-d**, **-D <origin>**, **-f <flavor>**, **-g <chainName>**, **-n**, **-o**, **-O "<flagList>"**, **-p "<description>"**, **-q <qualifierList>**, **-t**, **-z <databaseList>**, **-0**, **-1**, **-2**, **-3**
-h <host>	Specifies distribution host; the default is *fnkits.fnal.gov*. Usually just the plain host name is required, however the **-h** option can always specify any of the following: • a plain host name (e.g., *fred.sub.domain*, or just *fred* if *sub.domain* is *fnal.gov*) • a host and Webserver port number (e.g., *fred.sub.domain:8080*) • a full URL to the `ups.cgi` cgi script (e.g., `http://fred.sub.domain:8080/cgi-bin/some/dir/ups.cgi`)
-H <flavorList>	Specifies flavor and builds a flavor list for that family starting at the level specified. Multiple values accepted; several **UPD** commands operate on each listed flavor, however most **UPS** commands ignore all except first in list. In many commands **-H** can be used alone (without an accompanying number option). In this case, **UPS/UPD** finds the best match instance for the specified flavor family. If used with any of **-0**, **-1**, **-2**, **-3**, **UPS/UPD** finds the product instance of specified level of that flavor; e.g., **-2H IRIX+6.2** is equivalent to **-f IRIX+6**. Note: can also be used inside **-G ""**
-i	Ignores errors; allows operation to continue to the next product instance (e.g., if one product dependency can't be downloaded during **upd install**, it proceeds to the next one)
-I	Unwinds product tar file on the local node after transmission (default is to unwind during transmission); this option allows **ftp** to "reget" a tar file if network connection is lost during the transfer
-j	Ignores dependencies, operates just on top level product

Table 25.1.0-a:

-J <fileName>	Specifies individual file to fetch (for archive database, can only be `INSTALL_NOTE`, `README`, or a version or table file)
-k	Prevents execution of unsetup files prior to (subsequent) setup
-K <keywordList>	Returns values of specified keywords only; see section 25.2.3 *-K* for more information
-l	Produces a long listing
-L	Adds the STATISTICS keyword to the version file, thereby instructing **UPS** to keep statistics on this product instance. A record of the form: `"tcl" "v7_3q" "IRIX" "" "" "berman" "1998-03-13 17.56.54 GMT" "list"` will get added to the file `$PRODUCTS/.upsfiles/statistics/<product>` each time a **UPS** command is run on this instance. See section 28.6.3 *STATISTICS* for more information.
-m <tableFileName>	Specifies table file name
-M <tableFileDir>	Specifies table file directory
-n	Specifies "new" chain Note: can also be used inside **-G ""**
-N <fileName>	Specifies file to be checked and edited by **ups modify**
-o	Specifies "old" chain Note: can also be used inside **-G ""**
-O "<flagList>"	Sets the value of $UPS_OPTIONS to **<flagList>**. This is a means of passing information to actions listed in the table file. Note: can also be used inside **-G ""**
-p "<description>"	Specifies product description Note: can also be used inside **-G ""**
-P	Requires **UPS** to rely only on information supplied on the command line to locate the product instance (prevents **UPS** from searching in a database)
-q <qualifierList>	Specifies required or optional qualifiers; see section 25.2.4 *-q* for more information Note: can also be used inside **-G ""**

Table 25.1.0-a:

-r <prodRootDir>	Specifies the product root directory (when declaring a product, sets the value of the keyword PROD_DIR) If **<prodRootDir>** specifies a relative path, **UPS** appends it to PROD_DIR_PREFIX in order to construct the entire path. (PROD_DIR_PREFIX is set in the `dbconfig` file; see Chapter 31: *The UPS Configuration File*)
-R	Retrieves/operates on only the required dependencies (those listed in table file using **setupRequired**); ignores the optional ones.
-s	Lists what command would do; but does not execute the command (creates the temp file, but does not source it)
-t	Specifies "test" chain Note: can also be used inside **-G ""**
-T	Specifies archive file directory or URL (when declaring a product, sets the value of the keyword ARCHIVE_FILE) Note: can also be used inside **-G ""**
-u <compileDir>	Specifies the directory for the output file (which is named via the **-b** option) for the **ups compile** command associated with the COMPILE action (see Chapter 38: *Use of Compile Scripts in Table Files*)
-U <upsDir>	Specifies location of `ups` directory (when declaring a product, sets the value of the keyword UPS_DIR); default value is **ups** If **<upsDir>** specifies a relative path, **UPS** appends it to the product root directory in order to construct the entire path. Note: can also be used inside **-G ""**
-v	Prints out extra debugging information. To get progressively more information, use multiple **v**'s, e.g., **-vv**, **-vvv** (up to four).
-V	Does not delete the temporary script files or partially installed products when command finishes; instead lists them on the screen See section 25.2.5 *-V* for more information on the temporary scripts. Note: can also be used inside **-G ""**
-w	For **ups start**, stops the product first, then restarts it
-W	For **ups copy**, uses environment variables (e.g., $SETUP_<PRODUCT>) to identify dependent product instances for target product (that is, it uses instances that are already setup in preference to what is listed in table file)

Table 25.1.0-a:

-X	For **upd install** and **ups copy**: executes the generated **ups declare** command(s) instead of just printing to screen
-y	For **ups undeclare**: deletes product root directory, provides confirmation prompt
-Y	For **ups undeclare**: deletes product root directory, does not provide confirmation prompt
-z <databaseList>	Specifies the local database(s); see 27.1 *Database Selection Algorithm* for more information Note: can also be used inside **-G ""**
-Z	Times the command (does not include time for sourcing of temp file for **setup/unsetup**)
-0	Specifies NULL flavor string; equivalent to **-f NULL** Not valid with **-f** or other number options. Can usually be used with **-H** to specify a single flavor (in this case, NULL). Note: can also be used inside **-G ""**
-1	Specifies flavor as OS value up to the generic OS (e.g., SunOS); can be used in place of **-f <flavor>** Examples: • Used alone on a SunOS machine, it is equivalent to **-f SunOS**. • If given together with **-H IRIX+6.2** on any machine (e.g., **-2H IRIX+6.2**, it is equivalent to **-f IRIX**. Not valid with **-f** or other number options. Note: can also be used inside **-G ""**
-2	Specifies flavor as OS value up to the version (e.g., SunOS+5); can be used in place of **-f <flavor>** Examples: • Used alone on a SunOS+5 or SunOS+5.6 machine, it is equivalent to **-f SunOS+5**. • If given together with **-H IRIX+6.2** on any machine (e.g., **-2H IRIX+6.2**, it is equivalent to **-f IRIX+6**. Not valid with **-f** or other number options. Note: can also be used inside **-G ""**

Table 25.1.0-a:

-3	Specifies flavor as OS value up to the release of a version (e.g., SunOS+5.6); can be used in place of **-f <flavor>** Examples: • Used alone on a SunOS+5.6 machine, it is equivalent to **-f SunOS+5.6**. • Used alone on a SunOS+5 machine, it is equivalent to **-f SunOS+5** (**-2** and **-3** are equivalent in this case). If given together with **-H IRIX+6.2** on any machine (e.g., **-3H IRIX+6.2**, it is equivalent to **-f IRIX+6.2**. Not valid with **-f** or other number options. Note: can also be used inside **-G ""**
-4	Specifies flavor as OS value up to the patch level (e.g., Linux+2.4.18); can be used in place of **-f <flavor>** Examples: • Used alone on a Linux+2.4.18 machine, it is equivalent to **-f Linux+2.4.18**. • Used alone on a Linux+2.4 machine, it is equivalent to **-f Linux+2.4** (**-3** and **-4** are equivalent in this case). If given together with **-H IRIX+6.2.x** on any machine (e.g., **-4H IRIX+6.2.x**, it is equivalent to **-f IRIX+6.2.x**. Not valid with **-f** or other number options. Note: can also be used inside **-G ""**

25.2 More Information on Selected Options

25.2.1 -e

UPS_EXTENDED is an on/off type variable. Scripts or the product's table file may contain the UPS_EXTENDED variable to execute "extended" functionality. In order to enable the extended functionality, first **-e** must be set at the time that the product is initially declared, and again in the **setup** command.

For dependencies, first **-e** must be set at the time that the parent product is initially declared, and again in the **setup** command for the parent product.

25.2.2 -H

In order to describe the **-H** option, we first need to define *flavor table* and *flavor levels*:

- a *flavor table* is a list including every level of specificity for a particular OS that you could use to find or declare a product instance. For example, on a SunOS machine of release 5.6, the flavor table generated from **-H SunOS+5.6** is:

```
SunOS+5.6
SunOS+5
SunOS
NULL
ANY
```

- a *flavor level* is a level of specificity for a flavor, e.g., any of the items in the above flavor table.

The **-H** option can be used to run a command "as if" the local machine were of the specified flavor. It can be used alone, in which case **UPS/UPD** uses a best match algorithm to find a matching instance. In contrast, when flavor is specified by **-f**, the command fails if no exact match is found.

The **-H** option can also be used with a number option to specify a particular level of the given flavor family. When used like this, it is equivalent to specifying a flavor via **-f**. E.g., **-2H IRIX+6.2** is equivalent to **-f IRIX+6**.

25.2.3 -K

Two output styles are provided for the commands that print information to the screen: a formatted one that is easy for users to read, and a condensed one for parsing by a subsequent command or a script. Use the **-K** option to request output in the condensed format. The **-K** option requires an argument list specifying which fields to include in the output, for example:

```
% ups list -K product:version:flavor xemacs
   "xemacs" "v19_14" "SunOS+5"
```

The plus sign (**+**) argument, e.g., **-K+**, is a shorthand for requesting the default fields **product:version:flavor:qualifiers:chain**, for example:

```
% ups list -K+ xemacs
   "xemacs" "v19_14" "SunOS+5" "" "current"
```

Some common keyword arguments used with the **-K** option are:

PRODUCT	product name
FLAVOR	product instance flavor

VERSION	product version
QUALIFIERS	additional instance specification information often used to indicate compilation options used by developer
CHAIN	product instance chain
+	all of the above
DATABASE (or DB)	the **UPS** database path; useful if more than one on system
DECLARER	logon id of person who declared the instance
DECLARED	date/time that product instance was declared
MODIFIER	logon id of person who modified/updated the instance
MODIFIED	date/time that product instance was modified/updated

The full list of keywords with definitions can be found in section 28.4 *List of Supported Keywords* and in section 23.11 *ups list*.

The condensed format is useful for parsing the output in scripts. A couple of examples are given in section 23.11 *ups list*. It is also convenient for piping **`ups list`** output to the **`grep`** command, e.g.,:

```
% ups list -aK+ xemacs | grep OSF1
   "xemacs" "v19_14" "OSF1+V3" "" ""
   "xemacs" "v19_14" "OSF1+V4" "" ""
   "xemacs" "v20_4" "OSF1+V3" "" "current"
   "xemacs" "v20_4" "OSF1+V4" "" "current"
```

25.2.4 -q

Products can be declared to the database with qualifiers (see section 2.3.3 *Qualifiers*) using the **`-q`** option with **`ups declare`**. To then match any product instance that has been declared with one or more qualifiers (see section 27.2 *Instance Matching within Selected Database*), you must specify the exact set of qualifiers on the command line, again using the **`-q`** option.

Qualifiers are case-sensitive, and they must be specified on the command line exactly as they appear in the product declaration. Use the **`ups list`** command to determine how a qualifier was declared.

A qualifier may be preceded by one of two operators (**`-q [+|?]qualifier`**) which explicitly makes the qualifier *required* (**`+`**) or *optional* (**`?`**). These types are described below. A qualifier not preceded by either operator is *variable*. Variable is the most commonly used form, and means that:

- when used with a command that sets a qualifier, e.g., **`ups declare`**, **`upd addproduct`**, the qualifier is treated as *required*

- when used with a command that must find a product instance and then do something with the product instance, e.g., **`setup`**, **`ups list`**, **`ups depend`**, **`upd install`**, the qualifier is treated as *optional*

Required Qualifiers

For instance matching, qualifiers can be entered as *required*, where "required" means that to match an instance, the qualifier *must* appear in its declaration. Required qualifiers are entered in the format **`-q <[+]qualifier>`**, for example:

```
% ups declare -c -f SunOS+6 -q build gtools v2_1
% setup -q +build gtools
```

For multiple required qualifiers, the format is **`-q <[+]qual_1:[+]qual_2:...>`**, for example:

```
% setup -q +build:+debug gtools v2_1
```

Optional Qualifiers

Alternatively, qualifiers can be requested as *optional*; i.e., they can but don't have to be in the declaration in order to match. If you request a qualifier as optional, **UPS** selects an instance declared with that qualifier in preference to an otherwise matching instance without it. Optional qualifier specifications may be preceded by a question mark (**`?`**). Given the default behavior, the question mark is no longer necessary, but if used, it should be enclosed in quotes, in the format **`-q "?<qualifier>"`**, for example:

```
% ups declare -c -f SunOS+6 -q build gtools v2_1
% setup -q +build gtools
% setup -q "?build" gtools v2_1
```

If quotes are not used when **`?`** is included in the string, the shell may return the error "No match".

Mixing Required and Optional

You can list some qualifiers as required and others as optional, where **`?`** precedes only the optional ones, for example:

```
% setup -q "+build?debug" gtools v2_1
```

Notice that the colon is not needed between qualifiers if the question mark is used.

25.2.5 -V

A **UPS** command generally executes its internal processes and also looks in the table file for corresponding actions (e.g., **`ups declare`** looks for ACTION=DECLARE and ACTION=CONFIGURE, by default). If the command finds one or more corresponding actions, the command first translates the functions listed under the actions into shell commands. These commands get written to a temporary script in $TMPDIR (if $TMPDIR isn't set, the default is `/tmp`), and the script is then invoked to execute the shell commands. By default, this script gets deleted. The **`-V`** option prevents **UPS** from deleting it. This is very useful for debugging table files. Example:

```
% ups declare -V ... myprod v1_0

   INFORMATIONAL: Name of created temp file is /tmp/aaaaakADa
```