

Chapter 26: UPS/UPD Command Usage

This chapter describes the syntax for **UPS** and **UPD** commands.

26.1 Syntax

Most **UPS** and **UPD** commands are of the form `ups <command>` or `upd <command>` (the exceptions are `setup` and `unsetup`), and take a variety of command line options and arguments. The standard syntax is:

```
% <command_name> [<options>] [<product> [<version>]] \  
  [, [<options>] [<product> [<version>]]]
```

For example:

```
% ups list -f IRIX+6 xemacs v20_4
```

The allowed arrangements of command line elements are described below.

26.1.1 Order of Command Line Elements

Most commands take two optional positional parameters, the product name and product version, plus any options. The first occurring unflagged element on the line is interpreted as the product name, and the next (if any) is interpreted as the version. With that limitation, the name, version and options can occur anywhere on the command line.

26.1.2 Specifying Version/Chain

When the product version is not specified, and no chain flag is listed, the version associated with the current chain is used.

You cannot specify a chain name (e.g., `new`) in place of the product version or in place of the chain flag (e.g., `-n`) on the command line (this was permissible in the old **UPS**). For example, you can enter:

```
% setup -n foo
```

but you cannot enter:



```
% setup foo new
```

26.1.3 Grouping Option Flags

Option flags can be grouped, for example `-ct` is equivalent to `-c -t`. Only the last option in a grouped list can have an argument, and that argument must follow directly, for example:

```
% ups list -ctf <flavor> <product>
```

26.1.4 Specifying Arguments to Options

Generally, a space is optional between an option flag and its argument. For example, the following two specifications are correct:

```
-f IRIX
```

```
-fIRIX
```

An exception to this rule is the specification of a path starting with tilde (~). There must be a space before the tilde due to `cs` behavior, for example:

```
-r ~user/a/b
```

When an option has multiple arguments, the arguments must be separated by a colon (:)¹, e.g.,

```
% ups list -f IRIX:SunOS:OSF1+V3 proda
```

lists instances of product **proda** for flavors IRIX, SunOS and OSF1+V3.

Arguments to options may be enclosed in double quotes, and there may be spaces between them in addition to the required colons. If there are spaces, quotes must be used. A multiple-argument specification can therefore look like any of the following examples:

```
-f IRIX:SunOS
```

```
-fIRIX:SunOS
```

```
-f"IRIX:SunOS"
```

```
-f "IRIX:SunOS"
```

```
-f "IRIX: SunOS    "
```

This permits strange looking option lists, for example:

```
-ctfIRIX:SunOS
```

1. A special note for specifying multiple databases: For systems with multiple **UPS** databases, the databases in \$PRODUCTS may be specified using either colons or spaces as separators (e.g., as `db1:db2:db3` or `db1 db2 db3`) for compatibility with older versions of **UPS**. But when the `-z` option is used on the command line, only colons are valid as separators between multiple databases (as between multiple arguments for any option); you cannot use spaces.

26.1.5 Embedded Spaces in Option Arguments

If an argument has any spaces embedded in it, you must put quotes around it, e.g.,

```
% ups list -q "my compile option" proda
```

Most options you'll find won't have spaces in them. During processing, any arguments that have embedded spaces get packed, with the exception of arguments to the options `-p` (product description) and `-O` (set \$UPS_OPTIONS).

26.1.6 Invalid Option Arguments

Any argument that can be interpreted as an additional option will be deemed invalid. (This restriction doesn't apply to the `-O` (uppercase `-o`) option.) As an example, the `-1` in the following commands will be interpreted as an invalid flavor:

```
% ups list -f -1 proda
```

```
% ups list -f "-1" proda
```

26.1.7 Specifying Multiple Products in a Single Command

A single **UPS** command can be issued for multiple product instances, and in fact for multiple product names, in many cases. The command line elements identifying each product must be separated with a comma (,). For example, the following set of three commands:

```
% setup proda
```

```
% setup prodb -o
```

```
% setup -f IRIX+6.5 prodc
```

can be condensed into:

```
% setup proda, prodb -o, -f IRIX+6.5 prodc
```

The simplest example of this is a command of the form:

```
% ups list proda, prodb, prodc
```

where the comma (,) is used simply for separating multiple product names, and the spaces between product specifications are optional. Options are reset when a comma is reached, so even if all the product names in a combined command share the same set of options and arguments, the common elements still need to be listed separately for each product, e.g.,

```
% ups list -f SunOS+5 -z $MY_DB proda, -f SunOS+5 -z $MY_DB  
prodb
```



26.1.8 Multiple Occurrences of Same Option Flag

There can be multiple occurrences of the same flag on the command line and where appropriate, the order is significant and preserved. This is rarely used. As an example:

```
% ups list <product> -c -o -c -g "mychain"
```

lists four chains in the order: current, old, current, and mychain. Here is a more complicated example illustrating option flag grouping as well as repeated options:

```
% ups declare -cf SunOS+5 -Zvvz $MY_DB foo -tnd v2_0
```

The `-v` option requests verbose output; use of multiple `-v`'s increases the level of verbosity. This awkward command declares the product `foo v2_0` as flavor `SunOS+5`, with four chains (current, test, new and development) to the database `$MY_DB`. It also times the command (`-Z`) and requests verbose output (`-vv`).

26.1.9 Use of Wildcards

No wildcard functionality is supported.

26.2 Options

When using the **UPS** and/or **UPD** commands, you must be aware that the options you'll need to specify depend upon your environment. In general, these rules apply:

- **UPS** takes the default flavor as that of the machine on which the command is issued (to the highest specification level possible), or if modifying product instance information, it takes the flavor in the product instance's declaration. This is important to know especially if your database is on a multi-flavor cluster (i.e., includes different operating systems or two or more nodes of the same operating system with different releases). If you wish to specify the flavor differently, use the `-f` or the number options (`-0` through `-3`, described in Chapter 25: *Generic Command Option Descriptions*).
- If your `$PRODUCTS` environment variable is set to multiple databases, you may sometimes need to specify the database to use (see section 27.1 *Database Selection Algorithm*) via the `-z` option.

- If there is more than one instance of a particular product installed on your machine, or in the case of a cluster, more than one instance for a single flavor, you can specify the desired instance uniquely by including its version or chain and possibly qualifiers.
- No option flag is associated with product version. The version, if given, must appear after the product name on the command line, although not necessarily immediately after it.
- The number options (**-0**, **-1**, **-2**, and **-3**) cannot be used in combination with each other, or in combination with **-f**. They may be used in combination with **-H** in many commands.

