

# Node Filtering Options

## Table of contents

1 Filtering Options.....	2
1.1 Keywords.....	3
2 Precedence.....	5
2.1 Explicit Precedence using --filter-exclude-precedence.....	6

Both the [ctl](#) and [ctl-exec](#) commands can use the node-dispatch feature to execute commands on remote nodes. The set of nodes that will execute the commands is determined by two things:

1. Which nodes have the specified Module or Object deployed on them (ctl only)
2. Which nodes match the filter options specified on the command line

When executing a ctl command, only those nodes that have the correct Module or Object deployed will be able to execute the specified defined command. The deployment specifications in the [deployments.properties](#) file determine which nodes have which deployments, so the "base set" of nodes is the set of matching entries in this document. For ctl-exec this is not necessary, and all nodes specified in [nodes.properties](#) are used as the base set.

Finally the base set is filtered based on the filtering options specified on the command line, as described below.

For more information about node-dispatch, or the related options in ctl or ctl-exec:

- [Concepts > Node Dispatch](#)
- [Running ctl > Node Dispatching Options](#)
- [Running ctl-exec > Node Dispatching Options](#)

## 1. Filtering Options

Both ctl and ctl-exec can use the commandline options `-I` (include, aka `--nodes`) and `-X` (exclude, aka `--xnodes`) to specify which nodes to include and exclude from the base set of nodes. You can specify a single value, a list of values, or a regular expression as the argument to these options.

When only inclusion filters (`-I`) are specified, then only the nodes from the base set that match those filters are used.

Examples:

```
ctl -I dev01 -m MyModule -c doSomething
ctl -I dev01,dev02 -m MyModule -c doSomething
ctl -I dev.* -m MyModule -c doSomething
```

This executes the `doSomething` command using various selector types. The first matches only the node with hostname `dev01`, the second two nodes with hostnames `dev01` and `dev02`, and the third all nodes with hostnames that match the regular expression `dev.*`.

When only exclusion filters (`-X`) are specified, then only the nodes from the base set that *do not* match those filters are used.

Examples:

```
ctl-exec -X web01 -- ps
ctl-exec -X web01,web02 -- ps
ctl-exec -X "web.*" -- ps
```

This executes the 'ps' command by excluding nodes based on hostname selectors, thus it will execute on all nodes that are not matched by the exclusion filters.

By default, the `-I` and `-X` options are used to filter based on the node `hostname` property, as it is specified in the [nodes.properties](#) file.

However, *keywords* can be used to specify other properties to filter on besides `hostname`:

## 1.1. Keywords

Keywords are specified by using one of the filter options with "key=value" as the argument to the option.

Example:

```
ctl-exec -I os-name=Linux -- ps
```

This executes on all nodes with an operating system named "Linux".

Notice that the argument to `-I` specifies `os-name=` and then the value `Linux`. The keyword used is `os-name`, and so the filter will match the "os-name" property in `nodes.properties`.

The available keywords are:

- `hostname` - hostname of the node [default keyword]
- `name` - entity name of the node, which may be different than hostname
- `type` - type name of the node, typically "Node"
- `tags` - a set of user defined tags
- `os-name` - operating system name, e.g. "Linux", "Macintosh OS X"
- `os-family` - operating system family, e.g. "windows", "unix"
- `os-arch` - operating system CPU architecture, e.g. "x86", "x386"
- `os-version` - operating system version number

All of the keywords can accept a single value, a comma-separated list, or a regular expression.

### 1.1.1. Tags

In addition, the `tags` keyword can use the boolean operator `+` to represent logical AND. The comma used to separate lists serves as logical OR. The following example matches all nodes tagged with both "devel" AND "secure", OR with "server":

```
ctl-exec -I tags=devel+secure,server -- ps
```

### 1.1.2. Combining Options

All keywords can be combined by specifying the `-I` or `-X` options multiple times on the command line.

The following example includes all nodes tagged with both "devel" and "secure", as well as all nodes with hostnames matching `web.*`:

```
ctl-exec -I tags=devel+secure -I web.* -- ps
```

The following includes all nodes tagged with both "devel" and "secure", as well as all nodes with hostnames matching `web.*`, and excludes the node with hostname "web01":

```
ctl-exec -X web01 -I tags=devel+secure -I web.* -- ps
```

Note, however, that you cannot specify the same keyword multiple times for the same type of filter:

**WRONG:**

```
ctl-exec -X web01 -X dev01 -I tags=devel+secure -I web.* -- ps
```

This example will not exclude the node "web01", because there are two `-X` options with the same keyword. Both `-X web01` and `-X dev01` default to the "hostname" keyword, and the second entry will override the first. To exclude both of those nodes, you must combine them for the option `-X web01,dev01`:

**CORRECT:**

```
ctl-exec -X web01,dev01 -I tags=devel+secure -I web.* -- ps
```

The last example brings up the issue of *Precedence*. The command line first specifies `-X web01`, then later specifies `-I web.*`. However, the hostname "web01" would be matched by the regular expression "web.\*". So is the node with hostname "web01" included in the set

of nodes to execute on, or is it excluded?

## 2. Precedence

Precedence is the issue of whether a node should be included in the result set when it matches both an exclusion filter and an inclusion filter.

Take a simplified example:

```
ctl-exec -X web01 -I web.* -- ps
```

The intent is to exclude "web01" while including all other nodes matching the regular expression "web.\*". Depending on which filter takes precedence, the exclusion filter or the inclusion filter, the result may be different.

- When *inclusion has precedence*, nodes that match both filters will be *included*.
- When *exclusion has precedence*, nodes that match both filters will be *excluded*.

So which filter has precedence?

*The first filter specified on a command line takes precedence.*

This means that if you specify any `-X` option before a `-I` option, then *exclusion* will take precedence, and vice versa.

So in the example above, the `-X` takes precedence (it is first), and so the node with hostname "dev01" is excluded from the result set.

If you change the order of the options:

```
ctl-exec -I web.* -X web01 -- ps
```

Then the node with hostname "web01" will be included in the results.

**Note:**

When only one filter is used, either `-I` or `-X`, there is no need to worry about precedence.

In general, a good rule of thumb when trying to determine which precedence you need is to *specify the most restrictive filter first*.

For an inverse example, suppose you want to dispatch to all non-windows nodes, but you want to include any nodes tagged with "development". You might try this at first:

**WRONG:**

```
ctl-exec -X os-family=windows -I tag=development -- ps
```

This will not return the correct result set, because the `-X` takes precedence as it is the first filter on the line. So any nodes that have both `os-family=windows` and `tag=development` will be *excluded*.

**CORRECT:**

```
ctl-exec -I tag=development -X os-family=windows -- ps
```

Here since the `-I` is specified first, the inclusion filter has precedence, and any nodes that have both `os-family=windows` and `tag=development` will be included in the result.

## 2.1. Explicit Precedence using `--filter-exclude-precedence`

Using the `--filter-exclude-precedence` command-line option, the precedence can be set explicitly. The argument is "true" or "false". When the argument is "true" then the exclusion filter takes precedence, regardless of the order of the filter options. When the argument is "false" then the inclusion filter takes precedence.

```
ctl-exec -I web.* -X web01 --filter-exclude-precedence true -- ps
```

This command-line correctly excludes the "web01" node because the `--filter-exclude-precedence` option is set to "true".