

title: Logagent Command-line Parameters description: Command Line Parameters for Logagent, light-weight log shipper with out of the box and extensible log parsing, on-disk buffering, secure transport, bulk indexing to Elasticsearch and Sematext logs management platform. Low memory footprint and low CPU overhead make it suitable for deploying on edge nodes and devices, while its ability to parse and structure logs makes it a great Logstash alternative.

```
# Parse all logs and stream them to Sematext Logs 1-Click managed Elasticsearch
$ logagent -i LOGSENE_TOKEN /var/log/*.log
```

```
# stream logs to local Elasticsearch
$ logagent -e http://localhost:9200 -i myindex /var/log/*.log
```

```
# Act as Syslog server on UDP and forward messages to Sematext Logs
$ logagent -u 514 -i LOGSENE_TOKEN
```

```
# Act as Syslog server on UDP and write YAML formatted messages to console
$ logagent -u 514 -y
```

Use a glob pattern to build the file list

```
$ logagent -i LOGSENE_TOKEN -g '/var/log/**/*.log'
# pass multiple glob patterns
$ logagent -i LOGSENE_TOKEN -g '{/var/log/*.log,/opt/myapp/*.log}'
```

Watch selective log output on console by passing logs via stdin and format in YAML

```
$ tail -f /var/log/access.log | logagent -y -n httpd
$ tail -f /var/log/system.log | logagent -f my_own_patterns.yml -y
```

Command Line Parameters

```
$ logagent [options] [file list]
```

Options	Description
Generate config files	
<code>-w, --writeConfig <file></code>	write example config to a file. The arguments <code>-i</code> , <code>-e</code> , <code>-g</code> are applied in the generated config. See also <code>-c</code>
<code>--writePatterns <file></code>	write example patterns.yml to a file. See also <code>-f</code>
General options	
<code>-h, --help</code>	output Logagent help
<code>-V, --version</code>	output Logagent version
<code>-v, --verbose</code>	verbose debug output for all plugins

Options	Description
<code>-c, --config <configFile></code>	path to Logagent config file (see below)
<code>--geoipEnabled <value></code>	true/false to enable/disable geoip lookups in patterns.
<code>--diskBufferDir path</code>	directory to store status and buffered logs (during network outage)
<code>--includeOriginalLine</code>	includes the original message in parsed logs
<code>-f, --file <patternFile></code>	file with pattern definitions, use multiple -f options for multiple files
<code>--skipDefaultPatterns</code>	skips loading of default patterns.yml file
<code>-s, --suppress</code>	silent, print no logs to stdout; print only stats on exit
<code>--printStats</code>	print processing stats in the given interval in seconds, e.g. <code>--print_stats 30</code> to stderr. Useful with -s to see Logagent activity on the console without printing the parsed logs to stdout.
Log input options	
<code>--stdin</code>	read from stdin, default if no other input like files or UDP are set
list of files	Every argument after the options list is interpreted as a file name. All files in the file list (e.g. <code>/var/log/*.log</code>) are watched by tail-forever starting at end of file
<code>-g glob-pattern</code>	use a glob pattern to watch log files e.g. <code>-g "{/var/log/*.log,/Users/stefan/myapp/*.log}"</code> . The complete glob expression must be quoted to avoid interpretation of special characters by the Linux shell.

Options	Description
<code>--tailStartPosition bytes</code>	-1 to tail from end of file, >=0 to start from the given position (in bytes). This setting applies to new files without their read position saved (see <code>--diskBufferDir</code>)
<code>-n name</code>	name for the log source only when stdin is used, important to make multi-line patterns working on stdin because the status is tracked by the log source name
<code>-u <port></code>	starts a syslogd UDP listener on the given port to act as syslogd
<code>--journalld <port></code>	starts http server to receive logs from systemd-journal-upload.service
<code>--docker <docker-socket></code>	collect docker logs e.g. <code>--docker /var/run/docker.sock</code>
<code>--dockerEvents</code>	collects Docker events from <code>/var/run/docker.sock</code>
<code>--k8sEvents</code>	collects Kubernetes events from Kubernetes API. Detects automatically kubectrl or in-cluster config for API access
<code>--k8sContainerd</code>	enable Kubernetes containerd input-filter plugin to parse containerd logs from <code>/var/log/pods</code>
<code>--heroku <port></code>	listens for Heroku logs (http drain / framed syslog over http)
<code>--cfhttp <port></code>	listens for Cloud Foundry logs (syslog over http)
Output options	
standard output stream (default)	combine Logagent with any Unix tool via pipes
<code>-y, --yaml</code>	prints parsed messages in YAML format to stdout
<code>-p, --pretty</code>	prints parsed messages in pretty JSON format to stdout

Options	Description
<code>-j, --ldjson</code>	print parsed messages in line-delimited JSON format to stdout
Elasticsearch or Sematext Cloud	Log storage
<code>-e, --elasticsearchUrl <url></code>	Elasticsearch URL e.g. <code>http://localhost:9200</code> , default <code>https://logsene-receiver.sematext.com</code>
<code>-i, --index <index></code>	Logs App token to ship data to Sematext Cloud Apps or Elasticsearch index (see <code>--elasticsearchUrl</code>)
<code>--httpProxy <url></code>	HTTP proxy url
<code>--httpsProxy <url></code>	HTTPS proxy url

The default output is line-delimited JSON for parsed log lines, as long as no format options like `-y` (YAML format), `-p` (pretty JSON), or `-s` (silent, no output to console) are specified.

Environment variables

Variable	Description
<code>LOGS_TMP_DIR</code>	Directory to store failed bulk requests for later retransmission.
<code>LOGS_INTERVAL</code>	Time to batch logs before a bulk request is done. Default is 10000 ms (10 seconds)
<code>LOGS_BULK_SIZE</code>	Maximum size of a bulk request. Default is 1000.
<code>LOGS_RECEIVER_URL</code>	URL for the Logsene receiver. For a local Elasticsearch server or for Sematext Enterprise version of Logsene. Defaults to Sematext Logsene SaaS receiver <code>https://logsene-receiver.sematext.com/_bulk</code> . Example for Elasticsearch: <code>LOGSENE_URL=http://localhost:9200/_bulk</code>
<code>HTTPS_PROXY</code>	Proxy URL for HTTPS endpoints, like Logsene receiver. <code>export HTTPS_PROXY=http://my-proxy.example</code>
<code>HTTP_PROXY</code>	Proxy URL for HTTP endpoints (e.g. On-Premises or local Elasticsearch). <code>export HTTP_PROXY=http://my-proxy.example</code>

Variable	Description
LOGAGENT_CONFIG	Filename to read Logagent CLI parameters from a file, defaults to <code>/etc/sematext/logagent.conf</code>
PATTERN_MATCHING_ENABLED	Default is 'true'. The value 'false' disables parsing of logs.
SCAN_ALL_PATTERNS	Default is 'false'. For performance reasons, patterns are matched by source name. Setting the value to 'true' enables pattern search regardless of source name