

LOGZILLA DOCUMENTATION

Trigger Import Export

Import and export LogZilla triggers as JSON from the Tools menu or command line, with duplicate detection to prevent overwriting existing rules

Alerts · Generated April 27, 2026 · logzilla.ai/docs/alerts/trigger-import-export

Trigger Import and Export

LogZilla Triggers are stored in standard JSON format and may be imported and exported from both the UI and the command line.

Import/Export From UI

Exporting Triggers

Users may export all triggers or individual triggers by selecting either the **Tools** menu or an individual trigger's **edit** menu dropdown.

In either case, selecting the **export** option will prompt for the filename and location to be saved to.

Trigger Import/Export Menus

The screenshot displays the 'TRIGGERS' management interface. At the top left, there is a search bar and a dropdown menu for 'Actions' set to 'All', and another dropdown for 'Admin User'. A table lists three triggers with columns for NAME, ACTIONS, TOTAL COUNT, LAST FIRED, and CREATED BY. The first trigger is 'Cisco: OSPF Neighbor Change', the second is 'Cisco: Non IPSec-encapsulated Crypto', and the third is 'Cisco: Spanning Tree BPDU'. Each trigger row has a set of action buttons: 'Issue notification', 'Mark as actionable', and 'Mark as known'. To the right of the table, there are two dropdown menus. The top one is labeled 'Tools' and contains 'Import from file' and 'Export all triggers'. The bottom one is labeled 'Individual Trigger Export' and contains 'Export to file' and 'Delete trigger'. Two orange callout boxes with arrows point to these dropdown menus.

ID	NAME	ACTIONS	TOTAL COUNT	LAST FIRED	CREATED BY
1	Cisco: OSPF Neighbor Change	Issue notification, Mark as actionable, Mark as known	0	— Never —	Admin User 2018-02-02 22:30:16
2	Cisco: Non IPSec-encapsulated Crypto	Issue notification, Mark as actionable, Mark as known	0	— Never —	
3	Cisco: Spanning Tree BPDU	Issue notification, Mark as actionable, Mark as known	0	— Never —	Admin User 2018-02-02

Importing Triggers

The **Tools** menu also includes an option to import triggers.

During individual trigger import, a check is made to ensure that the trigger being imported is not a duplicate of an existing trigger. If the import is a duplicate, the option to click the checkbox for that trigger will not be available.

Trigger Import/Export - Unable to select due to existing trigger

Import triggers ✕

Test trigger import/export

Cancel
Import triggers

Trigger Import/Export - Trigger import passes test, select to proceed

Import triggers ✕

Test trigger import/export

Cancel
Import triggers

Command Line

Import

The output below shows the syntax for importing triggers from the command line.

Available options are:

- `-I` or `--input-file` : the name of the file to import
- `--owner` : an optional username to assign as the owner/creator of that trigger

```
# logzilla triggers import -h
usage: triggers import [-h] [-I INPUT_FILE] [--owner OWNER] [name]

positional arguments:
  name                name filter. To use wildcard put word in quotation
                    marks e.g.: "*cisco*"

optional arguments:
  -h, --help          show this help message and exit
  -I INPUT_FILE, --input-file INPUT_FILE
                    import triggers from file
  --owner OWNER       set owner for imported triggers. Default "admin"
```

Export

The output below shows the syntax for exporting triggers from the command line.

Available options are:

- List all available triggers (-l)
- -O or --output-file : the name of the file the triggers will be exported to
- -F yaml or -F json : the format of the export file
- --owner : only export triggers belonging to the specified owner
- --trigger-id : only export the specified (by id) trigger
- --with-built-in : include built-in triggers in the export (by default they are not included)

```
# logzilla triggers export -h
usage: triggers export [-h] [-O OUTPUT_FILE] [-F {yaml,json}] [--owner OWNER]
                    [--trigger-id TRIGGER_ID] [--with-built-in]
                    [name]

positional arguments:
  name                name filter. To use wildcard put word in quotation
                    marks e.g.: "*cisco*"

optional arguments:
  -h, --help          show this help message and exit
  -O OUTPUT_FILE, --output-file OUTPUT_FILE
                    file to write triggers to
  -F {yaml,json}, --format {yaml,json}
                    export format
  --owner OWNER       limit triggers to those owned by given user
  --trigger-id TRIGGER_ID
                    trigger-id filter
  --with-built-in     show built-in triggers. By default built-in triggers
                    are hidden
```