LOGZILLA DOCUMENTATION

CLI Overview

Introduction to the logzilla command line tool, covering command structure, subcommands, options, arguments, and the built-in help system

Command Line Tools · Generated April 27, 2026 · logzilla.ai/docs/command-line-tools/cli-overview

© LogZilla Corporation · logzilla.ai

LogZilla Command Line Tools Overview

LogZilla provides a command line interface for system administration, data management, and automation. The CLI tools enable administrators to configure systems, manage data, and automate workflows without requiring web interface access.

Command Structure

All LogZilla commands follow a consistent pattern:

```
logzilla <command> [subcommand] [options] [arguments]
```

Examples:

```
logzilla version                    # Simple command
logzilla settings update TIME_ZONE=UTC  # Command with argument
logzilla query --type Search        # Command with options
logzilla apps install cisco-asa     # Command with subcommand and argument
```

Getting Help

Every command supports detailed help information:

```
# General help
logzilla --help

# Command-specific help
logzilla <command> --help
logzilla query --help
logzilla settings --help
```

Authentication and Access

Root User Requirement

Most LogZilla CLI commands require root privileges:

```
# Switch to root user
sudo su -

# Or run individual commands with sudo
sudo logzilla version
```

API Authentication

Commands that interact with the LogZilla API support multiple authentication methods:

Method 1: API Token (Recommended)

```
# Create an API token
logzilla authtoken create --user admin

# Use token with commands
logzilla query --authtoken <your-token> --type Search

# Set token in environment variable
export LOGZILLA_API_KEY="your-token-here"
logzilla query --type Search  # Uses environment variable
```

Method 2: Username and Password

```
logzilla query --user admin --password mypassword
```

Method 3: Configuration File

Create a configuration file at `$HOME/.lz5query`:

```
[lz5query]
authtoken = your-token-here
base_url = http://front/api
```

Or with username/password:

```
[lz5query]
user = admin
password = mypassword
base_url = http://front/api
```

Command Categories

LogZilla CLI tools are organized into functional categories:

System Administration

Configure LogZilla system settings and manage the service lifecycle.

Command	Purpose
`settings`	Configure system parameters
`license`	Manage licensing
`apps`	Install/manage LogZilla applications
`https`	Configure SSL certificates
`start/stop/restart`	Control LogZilla service
`version`	Display version information
`upgrade`	Upgrade LogZilla

User and Access Management

Manage user accounts, passwords, and authentication systems.

Command	Purpose
`password`	Change user passwords
`authtoken`	Manage API tokens
`ldap`	Configure LDAP/Active Directory

Data Management

Control data ingestion, storage, and lifecycle management.

Command	Purpose
`archives`	Manage data archiving

Command	Purpose
`events`	View event statistics
`drop`	Delete event data
`forwarder`	Configure event forwarding
`sniffer`	Capture network traffic

Event Processing

Configure how LogZilla processes and responds to events.

Command	Purpose
`rules`	Manage parser rules
`triggers`	Configure automated actions

Querying and Analysis

Search and analyze event data programmatically.

Command	Purpose
`query`	Perform structured queries

Troubleshooting

Diagnose system issues and gather diagnostic information.

Command	Purpose
`logs`	View LogZilla system logs
`inspect-dump`	Create diagnostic packages

© LogZilla Corporation · logzilla.ai

Common Patterns

Working with Time Ranges

Many commands accept time range parameters:

```
# Using presets
--ts-from "3 days ago"
--ts-from "24 hours ago"
--ts-from "today"
--ts-from "yesterday"

# Using specific dates
--ts-from "2024-01-01" --ts-to "2024-01-31"
--ts-from "2024-01-01 00:00:00" --ts-to "2024-01-01 23:59:59"

# Using Unix timestamps
--ts-from 1704067200 --ts-to 1706745600
```

Output Formats

Commands that return data often support multiple output formats:

```
# JSON output (default for most commands)
logzilla query --type TopN --output $HOME/results.json

# Excel format
logzilla query --type TopN --output /root/report.xlsx --format xlsx
```

Batch Operations

Many commands support batch operations through configuration files:

```
# Query parameters in JSON file
logzilla query --type Search --params search-config.json

# Import configurations from files
logzilla triggers import --input triggers.yaml
logzilla forwarder import --input-file forwarder-config.yaml
```

© LogZilla Corporation · logzilla.ai

Error Handling

CLI commands use standard exit codes:

- **0**: Success
- **1**: General error
- **2**: Invalid arguments
- **3**: Authentication failure

Check command success in scripts:

```
if logzilla version; then
    echo "LogZilla is accessible"
else
    echo "Error accessing LogZilla"
    exit 1
fi
```

Environment Variables

LogZilla CLI tools recognize several environment variables:

Variable	Purpose	Example
`LOGZILLA_API_KEY`	Default API token	`export LOGZILLA_API_KEY="token123"`
`LOGZILLA_BASE_URL`	API base URL	`export LOGZILLA_BASE_URL="http://logzilla.local/api"`
`LOGZILLA_USER`	Default username	`export LOGZILLA_USER="admin"`

Best Practices

Security

- **Use API tokens** instead of passwords when possible
- **Protect configuration files** with appropriate file permissions

- **Use environment variables** for automation scripts
- **Rotate API tokens** regularly

Automation

- **Check exit codes** in scripts for error handling
- **Use configuration files** for complex parameters
- **Log command output** for audit trails
- **Test commands** in development environments first

Performance

- **Use specific time ranges** to limit data processing
- **Leverage caching** where available
- **Run resource-intensive operations** during off-peak hours
- **Monitor system resources** during large operations

Quick Start Examples

Check System Status

```
# Verify LogZilla is running
logzilla version

# Check license status
logzilla license info

# View recent system logs
tail -f /var/log/logzilla/logzilla.log
```

Basic Data Query

```
# Create API token
TOKEN="$(logzilla authtoken create --user admin | tail -n1 | tr -d '\r')"

# Simple search query
echo '{"time_range": {"preset": "last_1_hours"}, "limit": 10}' > search.json
logzilla query --type Search --params search.json --authtoken $TOKEN
```

System Maintenance

```
# Archive old data
logzilla archives archive --expire-days 90

# Check system performance
logzilla events stats --ts-from "24 hours ago"

# Create diagnostic package
logzilla inspect-dump --output /tmp/diagnostics-$(date +%Y%m%d)
```

Next Steps

- **System Commands**: Learn about system administration commands
- **Data Commands**: Explore data management and lifecycle tools
- **Query Tools**: Master the query interface
- **Automation**: Build scripts and automated workflows

The LogZilla CLI provides tools for every aspect of log management. Each command category builds upon these foundational concepts to provide specialized functionality for different administrative tasks.

© LogZilla Corporation · logzilla.ai