LOGZILLA DOCUMENTATION

Search Syntax

LogZilla search syntax reference: boolean operators, wildcards, the four-character minimum, and Sphinx prefix and infix indexing rules

LogZilla Dashboards · Generated April 27, 2026 · logzilla.ai/docs/using-the-dashboard/search-syntax

LogZilla provides standard boolean-type search syntax much like users would expect when using Google. The only difference is the ability to append a wildcard (`*`).

- All searches are case *insensitive*
- All searches must contain at least 4 characters at a minimum unless otherwise configured by the administrator.
- Wildcard characters (`*`) count toward the minimum character requirement.

Correct Search Syntax

Example 1:

```
hello*
```

Example 2 (prefix/infix wildcard):

```
*hel*
```

Incorrect Search Syntax

Too few characters:

```
hel
```

The 4 character minimum is set in the Sphinx configuration which administrators can adjust. The minimum word length, prefix length, and infix length settings control search indexing behavior. Customers are welcome to contact LogZilla for guidance on modifying these settings.

Boolean Examples

Operator AND

The `AND` is automatically implied when separating search words with a space and **should not** be included in search criteria.

For example, searching on the text `hello world` would return results containing both `hello` and `world`.

Operator NOT

The `!` or `–` operators may be used to find events `NOT` containing the specified text. For example:

```
hello -world
```

Or

```
hello !world
```

Operator OR

A `|` (pipe) operator may be used to find events matching either of the given terms. For example:

```
hello | world
```

Would return all events matching "hello" or "world".

```
hello | other | world
```

Would return all events matching "hello" or "other" or "world".

Boolean Mode Wildcard

Many Network and Systems logs will include names such as `GigabitEthernet1/0/0`, etc. The wildcard feature allows users to specify a search term when they may not know the trailing characters.

For example:

```
gigabitethernet1*
```

Would return results for `GigabitEthernet1/0/0`, `GigabitEthernet1/0/2`, or even `GigabitEthernet100`.

A **prefix/infix wildcard** may also be used:

```
*bitethernet1/*/2
```

Would return results for `GigabitEthernet1/0/0`, `GigabitEthernet1/1/2` but not `GigabitEthernet100`.

Grouping

Note that expression *grouping* can be used. This is surrounding a search expression with parentheses "(" ")" . This must be used in cases in a multi-term search expression is used with an OR operator "|", in order to clarify which terms are handled by the OR. For example, to indicate that you want to find messages that contain the expression "foo bar", OR messages that contain "baz" but *not* "boz", you would do the following:

```
"foo bar" | (baz -boz)
```

Invalid Search Syntax

The following examples show some of the mixed-mode searches which are not supported at this time:

- Searches containing both `OR` and `NOT` operator's combined:

```
hello | -world
```

- Mixed "Phrase" `AND` or `NOT`

```
"hello world" !world2
```

```
"hello world" world
```

- Negative searching without a preceding positive search

```
!hello
```

This would be analogous to searching Google for every word on the internet that does `NOT` contain the word hello. Which, of course, would not be very useful.