# Line protocol

InfluxDB uses line protocol to write data points. It is a text-based format that provides the measurement, tag set, field set, and timestamp of a data point.



```
// Syntax
<measurement>[,<tag_key>=<tag_value>[,<tag_key>=<tag_valu

// Example
myMeasurement,tag1=value1,tag2=value2 fieldKey="fieldValu
```

Lines separated by the newline character `\n` represent a single point in InfluxDB. Line protocol is whitespace sensitive.

> Line protocol does not support the newline character `\n` in tag or field values.

# Elements of line protocol

```
measurementName,tagKey=tagValue fieldKey="fieldValue" 146
--------------- --------------- --------------------- ---
       |               |                   |
  Measurement       Tag set           Field set
```

## Measurement

(*Required*) The measurement name. InfluxDB accepts one measurement per point. *Measurement names are case-sensitive and subject to naming restrictions.*

***Data type:*** *String*

## Tag set

***Optional*** – All tag key-value pairs for the point. Key-value relationships are denoted with the `=` operand. Multiple tag key-value pairs are comma-delimited. *Tag keys and tag values are case-sensitive. Tag keys are subject to naming restrictions.*

***Key data type:*** *String*
***Value data type:*** *String*

## Field set

(*Required*) All field key-value pairs for the point. Points must have at least one field. *Field keys and string values are case-sensitive. Field keys are*

*subject to naming restrictions.*

***Key data type:*** *String*
***Value data type:*** *Float* | *Integer* | *UInteger* | *String* | *Boolean*

> *Always double quote string field values. More on quotes below.*
>
> ```
> measurementName fieldKey="field string value" 155(
> ```

## Timestamp

***Optional*** – The unix timestamp for the data point. InfluxDB accepts one timestamp per point. If no timestamp is provided, InfluxDB uses the system time (UTC) of its host machine.

***Data type:*** *Unix timestamp*

> ### *Important notes about timestamps*
>
> - To ensure a data point includes the time a metric is observed (not received by InfluxDB), include the timestamp.
> - If your timestamps are not in nanoseconds, specify the precision of your timestamps when writing the data to InfluxDB.

## Whitespace

Whitespace in line protocol determines how InfluxDB interprets the data point. The **first unescaped space** delimits the measurement and the tag set from the field set. The **second unescaped space** delimits the field set from the timestamp.

```
measurementName,tagKey=tagValue fieldKey="fieldValue" 146
                               |                      |
                          1st space              2nd spac
```

# Data types and format

## Float

IEEE-754 64-bit floating-point numbers. Default numerical type. *InfluxDB supports scientific notation in float field values.*

### Float field value examples

```
myMeasurement fieldKey=1.0
myMeasurement fieldKey=1
myMeasurement fieldKey=-1.234456e+78
```

## Integer

Signed 64-bit integers. Trailing `i` on the number specifies an integer.

| Minimum integer | Maximum integer |
| --- | --- |
| `-9223372036854775808i` | `9223372036854775807i` |

### Integer field value examples

```
myMeasurement fieldKey=1i
myMeasurement fieldKey=12485903i
myMeasurement fieldKey=-12485903i
```

## UInteger

Unsigned 64-bit integers. Trailing `u` on the number specifies an unsigned integer.

| Minimum uinteger | Maximum uinteger |
| --- | --- |
| `0u` | `18446744073709551615u` |

### UInteger field value examples

```
myMeasurement fieldKey=1u
myMeasurement fieldKey=12485903u
```

## String

Plain text string. Length limit 64KB.

### String example

```
# String measurement name, field key, and field value
myMeasurement fieldKey="this is a string"
```

## Boolean

Stores `true` or `false` values.

| Boolean value | Accepted syntax |
| --- | --- |
| True | `t` , `T` , `true` , `True` , `TRUE` |
| False | `f` , `F` , `false` , `False` , `FALSE` |

### Boolean field value examples

```
myMeasurement fieldKey=true
myMeasurement fieldKey=false
myMeasurement fieldKey=t
myMeasurement fieldKey=f
myMeasurement fieldKey=TRUE
myMeasurement fieldKey=FALSE
```

> Do not quote boolean field values. Quoted field values are interpreted as strings.

## Unix timestamp

Unix timestamp in a specified precision. Default precision is nanoseconds ( `ns` ).

| Minimum timestamp | Maximum timestamp |
| --- | --- |
| `-9223372036854775806` | `9223372036854775806` |

### Unix timestamp example

```
myMeasurementName fieldKey="fieldValue" 1556813561098000
```

# Quotes

Line protocol supports single and double quotes as described in the following table:

| Element | Double quotes | Single quotes |
|---|---|---|
| Measurement | *Limited* * | *Limited* * |
| Tag key | *Limited* * | *Limited* * |
| Tag value | *Limited* * | *Limited* * |
| Field key | *Limited* * | *Limited* * |
| Field value | **Strings only** | Never |
| Timestamp | Never | Never |

* *Line protocol accepts double and single quotes in measurement names, tag keys, tag values, and field keys, but interprets them as part of the name, key, or value.*

# Special Characters

Line protocol supports special characters in string elements. In the following contexts, it requires escaping certain characters with a backslash ( \ ):

| Element | Escape characters |
| --- | --- |
| Measurement | Comma, Space |
| Tag key | Comma, Equals Sign, Space |
| Tag value | Comma, Equals Sign, Space |
| Field key | Comma, Equals Sign, Space |
| Field value | Double quote, Backslash |

You do not need to escape other special characters.

### Examples of special characters in line protocol

```
# Measurement name with spaces
my\ Measurement fieldKey="string value"

# Double quotes in a string field value
myMeasurement fieldKey="\"string\" within a string"

# Tag keys and values with spaces
myMeasurement,tag\ Key1=tag\ Value1,tag\ Key2=tag\ Value2

# Emojis
myMeasurement,tagKey=🔍 fieldKey="Launch 🚀" 15568135610
```

## Escaping backslashes

Line protocol supports both literal backslashes and backslashes as an escape character. With two contiguous backslashes, the first is interpreted as an escape character. For example:

| Backslashes | Interpreted as |
| --- | --- |
| \ | \ |
| \\ | \ |
| \\\ | \\ |
| \\\\ | \\ |
| \\\\\ | \\\ |
| \\\\\\ | \\\ |

# Comments

Line protocol interprets `#` at the beginning of a line as a comment character and ignores all subsequent characters until the next newline `\n`.

```
# This is a comment
myMeasurement fieldKey="string value" 1556813561098000000
```

# Naming restrictions

Measurement names, tag keys, and field keys cannot begin with an underscore `_`. The `_` namespace is reserved for InfluxDB system use.

# Duplicate points

A point is uniquely identified by the measurement name, tag set, and

timestamp. If you submit line protocol with the same measurement, tag set, and timestamp, but with a different field set, the field set becomes the union of the old field set and the new field set, where any conflicts favor the new field set.

---

*Related*

Write data to InfluxDB

*write* *line protocol* *syntax*

---

Was this page helpful? Yes No

---

## *Support and feedback*

Thank you for being part of our community! We welcome and encourage your feedback and bug reports for InfluxDB and this documentation. To find support, use the following resources:

InfluxData Community InfluxDB Community Slack

**InfluxDB Cloud and InfluxDB Enterprise customers** can contact InfluxData Support.

Edit this page