We support several options for timestamps, time zones, time ranges, and dates. When collecting log data, the timestamp attached to messages is vital, both for the integrity of the data in your account, and for accurate query results. Because of the importance of timestamps, Sumo Logic indexes the timestamp of each message, making sure that data relevant to a query’s time range is returned properly in search results, which allows you to reconstruct a correct event timeline.

**Timestamps**

The timestamp is the part of a log message that marks the time that an event occurred. During ingestion, we can detect the message timestamp, convert it to Unix epoch time (the number of milliseconds since midnight, January 1, 1970 UTC), and index it. The timestamp is parsed either using the default timestamp parsing settings, or a custom format that you specify, including the time zone.

When configuring a Source you can choose to use the default timestamp parsing settings, or you can specify a custom format for us to parse timestamps in your log messages. The Enable Timestamp Parsing option is selected by default. If it's deselected, no timestamp information is parsed at all. Instead, we stamp logs with the time at which the messages are processed.

**Timestamp considerations**

By default, we can automatically detect timestamps in your log messages. Automatic detection identifies timestamps in common formats and prefers timestamps that appear early in the message.

If your log messages from a Source contain multiple timestamps, timestamps in unusual formats, or a mix of distinct timestamp formats, you have two options:

- Configure a Source for each log format
- Configure a custom timestamp format for your Source

The Collector assumes that all log messages coming from a particular Source will have timestamps that are close together. If a message comes through that appears to be more than one day earlier or later than recent messages from that Source it will be auto-corrected to match the current time. You can stop this auto-correction by explicitly configuring a custom timestamp format on your Source.

The Collector also assumes that all log messages coming from a particular Source will have timestamps that are within a window of -1 year through +2 days compared to the current time. Any log message with a parsed timestamp outside of
that window is automatically re-stamped with the current time. You must contact Sumo Logic Support to adjust this auto-correction behavior. See how to ingest old or historical data for further details.

---

**Automated Timestamp Parsing**

Sumo can automatically parse any of the following timestamp formats. If more than one valid timestamp is detected in a log message, Sumo will generally select the timestamp that appears “furthest left” in the message. Sumo uses the Java SimpleDateFormat library for timestamp parsing. For more information see https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html.

<table>
<thead>
<tr>
<th>Timestamp Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>yyyy-MM-dd'T'HH:mm:ss*SSSZZZZ</td>
<td>2018-08-20T13:20:10*633+0000</td>
</tr>
<tr>
<td>yyyy MMM dd HH:mm:ss.SSS zzz</td>
<td>2017 Mar 03 05:12:41.211 PDT</td>
</tr>
<tr>
<td>MMM dd HH:mm:ss ZZZZ yyyy</td>
<td>Jan 21 18:20:11 +0000 2017</td>
</tr>
<tr>
<td>dd/MMM/yyyy:HH:mm:ss ZZZZ</td>
<td>19/Apr/2017:06:36:15 -0700</td>
</tr>
<tr>
<td>MMM dd, yyyy hh:mm:ss a</td>
<td>Dec 2, 2017 2:39:58 AM</td>
</tr>
<tr>
<td>MMM dd yyyy HH:mm:ss</td>
<td>Jun 09 2018 15:28:14</td>
</tr>
<tr>
<td>MMM dd HH:mm:ss yyyy</td>
<td>Apr 20 00:00:35 2010</td>
</tr>
<tr>
<td>MMM dd HH:mm:ss ZZZZ</td>
<td>Sep 28 19:00:00 +0000</td>
</tr>
<tr>
<td>MMM dd HH:mm:ss</td>
<td>Mar 16 08:12:04</td>
</tr>
<tr>
<td>yyyy-MM-dd'T'HH:mm:ssZZZZ</td>
<td>2017-10-14T22:11:20+0000</td>
</tr>
<tr>
<td>yyyy-MM-dd HH:mm:ss ZZZZ</td>
<td>2017-08-19 12:17:55 -0400</td>
</tr>
<tr>
<td>yyyy-MM-dd HH:mm:ssZZZZ</td>
<td>2017-08-19 12:17:55-0400</td>
</tr>
<tr>
<td>yyyy-MM-dd HH:mm:ss,SSSZZZ</td>
<td>2017-06-26 02:31:29,573</td>
</tr>
<tr>
<td>yyyy/MM/dd*HH:mm:ss</td>
<td>2017/04/12*19:37:50</td>
</tr>
<tr>
<td>yyyy-MM-dd HH:mm:ss,SSS*zzz</td>
<td>2018 Apr 13 22:08:13.211*PDT</td>
</tr>
<tr>
<td>yyyy MMM dd HH:mm:ss.ZSS</td>
<td>2017 Mar 10 01:44:20.392</td>
</tr>
<tr>
<td>yyyy-MM-dd HH:mm:ss,SSSZZZ</td>
<td>2017-03-10 14:30:12,655+0000</td>
</tr>
</tbody>
</table>
dd/MM/yyyy HH:mm:ss 23/Apr/2017 11:42:35
dd MMM yyyy HH:mm:ss 23 Apr 2017 11:42:35
dd MMM yyyy HH:mm:ss*SSS 23 Apr 2017 10:32:35*311
MMdd_HH:mm:ss 0423_11:42:35
MMdd_HH:mm:ss.SSS 0423_11:42:35.883

Unix epoch timestamps

Unix epoch timestamps are supported in the following formats:

- 10 digit epoch time format surrounded by brackets (or followed by a comma). The digits must be at the very start of the message. For example, [1234567890] or [1234567890, other] followed by the rest of the message.
- 13 digit epoch time. The 13 digits must be at the very start of the message. For example, 1234567890123... followed by the rest of the message.
- 16 digit epoch time. The 16 digits must be at the very start of the message. For example, 1234567890123... followed by the rest of the message.
- 19-digit epoch time. The 19 digits must be at the very start of the message. For example, 1496756806.655123456…. followed by the rest of the message.
- We also recognize the time format for the Akamai log delivery service. The format is 13 digits with a period before the last three (ms) digits: 1234567890.123
- Comma separated values where the 5th value from the start of the message is a 10 digit epoch time. For example, field1, field2, field3, field4, 1234567890
- JSON formatted property called "timestamp" followed by a 13 digit epoch time. For example, "timestamp":"123456789013".
- Format of Cisco Fortigate/Meraki log message:

  <134>1 1439277406.903768018 Store_020026 flows src=<redact> dst=72.245.34.184 protocol=udp sport=62118 dport=53 pattern: 1 all

- Format of Linux audit message:

  type=PATH msg=audit(1439992022.365:83931889): item=0 name="/usr/sbin/ss"
  inode=91193416 dev=08:02 mode=0100755 ouid=0 ogid=0 rdev=00:00
Specifying a custom timestamp format

Sumo automatically parses most timestamps without any issues, but if you're seeing timestamp parsing issues you can manually specify the parse format. The steps are the same if you're configuring a new Source or if you're editing the timestamp information for an existing Source.

When you specify a custom format, you provide us with the timestamp format and optionally a regex to help locate the desired timestamp in your log line format. If you don't provide a locator, we'll scan the entire log message for a timestamp matching the given format by default. You can also test some sample log lines and see if we can parse the new format.

Your custom timestamp format must follow our supported timestamp conventions.

When providing multiple custom formats, specify the most common format first. Sumo will process each custom format in the order provided. Once a timestamp is located no further timestamp checking is done.

If no timestamps are located that match your custom formats Sumo will still attempt to automatically locate the log's timestamp.

To manually specify a timestamp format for a Source:

1. Do one of the following:
   ◦ If you're configuring a new Source, continue to step 2.
   ◦ To edit the timestamp settings for an existing Source, click Manage Data > Collection > Collection. Then click Edit to the right of the Source name and go to step 2.
2. Click Advanced (if the advanced settings are not already displaying.)
3. For Timestamp Format, select Specify a format.
4. In the Format field, enter the timestamp format that Sumo Logic should use to parse timestamps in your log. If the timestamp format is in epoch time, enter "epoch" in the Format field. Your custom timestamp format must follow our supported timestamp conventions.
5. The **Timestamp locator** field is required for 16-digit epoch or 19-digit epoch timestamps. Otherwise, this field is optional. If none of the provided formats match, Sumo will attempt to automatically determine the message's timestamp.

Note that a timestamp locator must:

- Be a valid Java regular expression. Otherwise, this error message will be displayed:
  Unable to validate timestamp formats. The timestamp locator regex `your-regex` is invalid. The timestamp locator regex `your-regex` uses matching features which are not supported.

- Be an **RE2-compliant** regular expression, for example: `\[time=(.*?)]`. Otherwise, this error message will be displayed:
  Unable to validate timestamp formats. The timestamp locator regex `your-regex` uses matching features which are not supported.

- Contain one unnamed capture group. When we extract timestamps, we only scan the portion of each log message that is captured by this group. If a log message does not match the locator expression, then your timestamp format cannot be applied to that message. If the regex doesn't contain one unnamed capture group, this error message will be displayed:
  Unable to validate timestamp formats. The timestamp locator regex `your-regex` does not contain a single unnamed capture group. The timestamp locator regex `your-regex` uses matching features which are not supported.

If you use quotes in the timestamp locator regular expression, you may see issues in the display after you save. The regular expression is not actually changed and can still be used to locate your timestamp.

6. If you have more than one custom timestamp format that you want to add, click **Add**.
   The ordering of formats is significant. Sumo will test each provided timestamp format, in the order specified, until a matching format is found. The first matching format determines the final message timestamp. If none of the provided formats match a particular message, Sumo will attempt to automatically determine the message's timestamp.

7. Click **Test** once you've entered all of your custom timestamp formats. If you've entered a valid regex in the timestamp locator, you'll see the **Test Timestamp Parsing** page. Enter sample log messages to test your timestamp formats you want Sumo to extract.
8. Click **Test**. The results display with the timestamp parsed and format matches (if any).

You should see one of the following messages:

- **Format matched.** In this example, the format of `yyyy/MM/dd HH:mm:ss` was matched and highlighted in green. This was the first format provided so it returns as 1(*format: yyyy/MM/dd HH:mm:ss locator: \[time=(.*?)\])* The **Effective message time** would be 2017-01-15 02:12.000 +0000.

- **None of the custom timestamp format was matched.** While the custom formats were not found in the log, there's still an auto detected timestamp highlighted in orange, 2017-06-01 02:12:12.259667 that we can use. The **Effective message time** is going to be 2017-06-01 02:12:12.259 +0000

- **Unable to parse any timestamp.** No part of the sample log line "This line shouldn't parse" has a parseable timestamp and so the timestamp will be the current time.

9. Optional. If you want to make changes to your log line, click **Edit** and you can provide other log lines to test.
10. Click **Done** to exit **Test Timestamp Parsing**.
11. Click **Save** to save your custom timestamp formats.

---

**Using _format for troubleshooting**

You can use **_format** to see how the timestamp is parsed from the log file.

The fields in the result of **_format** are:

\[ t:<parse\_type>,o:<offset>,l:<length>,p:<date\_format> \]

where **<parse\_type>** can take the values:

- **fail**—Failed to locate timestamp.
- **cache**—Success, cached format.
- **def**—Success, default (user-specified) format.
- **full**—Success, from "full" parsing against library of patterns.
- **none**—Local/receipt time because timestamp parsing is not enabled for this source.
- **ac1**—Auto-corrected by the "window-based" heuristic (what we call "auto-correction" today). Sumo Logic assumes that all log messages coming from a particular Source will have timestamps that are close together. If a message comes through that appears to be more than one day earlier or later than recent messages from that source, it will be auto-corrected to match the current time. You can stop this auto-correction by explicitly configuring a custom timestamp format on your Source.
  - For example, assume that Sumo parses the timestamp "Dec 2, 2017 2:39:58 AM". If the previously received message from that source has a timestamp prior to "Dec 1, 2017 2:39:58 AM" or after "Dec 3, 2017 2:39:58 AM", Sumo will auto-correct the timestamp to the current time.
- **ac2**—Auto-corrected by the -1y, +2d heuristic. Sumo Logic assumes that all log messages coming from a particular Source will have timestamps that are within a window of -1 year through +2 days compared to the current time. Any log message with a parsed timestamp outside of that window is automatically re-stamped with the current time.
  - For example, assume that Sumo parses the timestamp "Dec 2, 2017 2:39:58 AM". If the previously received message from that source is prior to "Dec 1, 2016 2:39:58 AM" or after "Dec 4, 2017 2:39:58 AM", Sumo will auto-correct the timestamp to the current time.

**Example**

When you’re troubleshooting issues related to timestamp, you can run a query similar to this to see how the timestamp is parsed:

```
[sourceCategory=PaloAltoNetworks] | _format as timestampformat
```

The result would look like this:
Large time between message time and receipt time

See troubleshooting discrepancies between message time and receipt time.

Timestamp conventions

The following conventions are supported as tokens and can be used in custom timestamp formats:

<table>
<thead>
<tr>
<th>Token</th>
<th>Date or Time Component</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>yyyy</td>
<td>4-digit year</td>
<td>2012; 2016</td>
</tr>
<tr>
<td>yy</td>
<td>2-digit year</td>
<td>12; 16</td>
</tr>
<tr>
<td>MMM</td>
<td>3-character month</td>
<td>Jan; Mar; Dec</td>
</tr>
<tr>
<td>MM</td>
<td>1- or 2-digit month (in a year)</td>
<td>1; 01; 9; 09; 12</td>
</tr>
<tr>
<td>dd</td>
<td>1- or 2-digit day (in a month)</td>
<td>1; 01; 16; 30</td>
</tr>
<tr>
<td>a</td>
<td>AM/PM (case insensitive)</td>
<td>AM; PM; am; pm</td>
</tr>
<tr>
<td>HH</td>
<td>1- or 2-digit hour (in a day, 0-23)</td>
<td>2; 02; 14; 23</td>
</tr>
<tr>
<td>hh</td>
<td>1- or 2-digit hour (in a day, 1-12 with AM/ PM)</td>
<td>2; 02; 11; 12</td>
</tr>
<tr>
<td>mm</td>
<td>1- or 2-digit minute (in an hour)</td>
<td>8; 08; 55</td>
</tr>
<tr>
<td>ss</td>
<td>1- or 2-digit second (in a minute)</td>
<td>5; 05; 35</td>
</tr>
<tr>
<td>SSS</td>
<td>1-3 digit subsecond or millisecond (in decimal)</td>
<td>4; 58; 944</td>
</tr>
<tr>
<td>zzz</td>
<td>3- letter time zone</td>
<td>UTC; PST; EDT</td>
</tr>
<tr>
<td>ZZZZ</td>
<td>RFC 822 time zone</td>
<td>-0900; +0500</td>
</tr>
<tr>
<td>'Z'</td>
<td>Literal Z character</td>
<td>Z</td>
</tr>
<tr>
<td>'T'</td>
<td>Literal T character</td>
<td>T</td>
</tr>
<tr>
<td>epoch</td>
<td>10, 13, 16, 19 digit timestamp with optional . (dot) after 10 digits.</td>
<td>1496756806.655123456</td>
</tr>
</tbody>
</table>
Do not use epoch with any other tokens.

**Time zones**

When configuring a Source, you can choose either of the following options:

- Use the time zone present in your log files, and then choose an option in case time zone information is missing from a log message.
- Have us completely disregard any time zone information present in logs by forcing a time zone.

It's important to have the proper time zone set, no matter which option you choose. If the time zone of logs can't be determined, we stamp them with UTC.

**Time zone considerations**

The following considerations apply to time zones:

- We highly recommend that the time zone be set explicitly on all Sources. Sumo Logic always attempts to determine the time zone for the Source. However, if that isn't possible, the time zone will revert to UTC. In these cases, the time zone will be incorrect, and that could significantly affect forensic analysis and reporting.
- Sumo Logic does not support all available ISO8601 time zones. For example -00 is not supported. So any timezones written in this format are undetectable by the system. For cases of these formats you will need to supply the proper default timezone to use when one is not detected by the service.

See this Wikipedia article for a list of UTC offsets and this Wikipedia article for a list of time zone codes.

**Default time zone**

By default, we use the time zone from your web browser set by the operating system to display hours and minutes everywhere in our user interface. You can change the default time zone that the user interface displays by adjusting the Default Timezone setting on the Preferences page. This option overrides the time zone from your web browser, and changes how hours and minutes are displayed in the UI. But this is a personal setting, and does not change the time zone for anyone else in your organization.

UI elements that are affected by this setting include the Search page Time Range field, the Time column of the Messages pane, Dashboards, and Anomaly Detection.

Changing the Default Timezone setting affects how the UI displays messages, but not the actual timestamp in the log message.

For example, the following screenshot shows the time zone set to PST in the UI, in the Time column. The logs were collected from a system that was also configured to use the PST time zone, which is displayed in the timestamp of
the **Message** column. The timestamps in both columns match as they are set to the same time zone.

<table>
<thead>
<tr>
<th>#</th>
<th>Time</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>06/16/2015 13:36:20.236 -0700</td>
<td>[HostId=rte-cloudcollector-2] [loggers.cloudcollectorITIVE.ElasticSearchIndexGenerator] [the Bucket: cloudcollector.] [bucket: wzen-143279734076. Prefix:]</td>
</tr>
</tbody>
</table>

The next screenshot shows the same search result after changing the Default Timezone setting to UTC. Now the Time column is displayed in UTC, while the Message column retains the original timestamp, in PST.

<table>
<thead>
<tr>
<th>#</th>
<th>Time</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>06/16/2015 13:36:20.236 -0700</td>
<td>[HostId=rte-cloudcollector-2] [loggers.cloudcollectorITIVE.ElasticSearchIndexGenerator] [the Bucket: cloudcollector.] [bucket: wzen-143279734076. Prefix:]</td>
</tr>
</tbody>
</table>

In another example, if your time zone is set to **UTC**, and you share a Dashboard with another user who has their time zone set to **PST**, what will they see?

They will see the same data, just displayed using their custom set time zone. For example, if you have a Panel that uses a time series, the timeline on the X axis of your chart is displayed in your time zone, **UTC**. The other user will see the timeline on the X axis displayed in their time zone, **PST**. But the data displayed in the chart is exactly the same.

---

**Time Ranges**

The **Time Range** field on the **Search** page uses the time zone that is set for the Sumo Logic user interface. This is either the default time zone used in the web browser and set by the operating system, or the **Default Timezone** setting on the **Preferences** page, if you have set this option.

When you create a **Scheduled Search** or a **Real Time Alert**, the time range of the search that you save uses the time zone that is set for the Sumo Logic user interface. If you have changed the time zone using the **Default Timezone** setting, this time zone will be used for your Scheduled Searches and Real Time Alerts.

The **Default Timezone** setting does not automatically update the configurations of existing Scheduled Searches or Real Time Alerts. So it is important to note that if you would like your Scheduled Searches and Real Time Alerts to use the same time zone as your user interface, you will need to edit them to do so, and save them.

For more information on time ranges, see [Set the Time Range of a Search](#).

Search Time Ranges can also search all data with any and all timestamps. For details, see [Use Receipt Time](#).
Date Format

If the browser used to access Sumo Logic is in a location that uses the day/month/year format instead of month/day/year, dates are presented in that format.