NAME

logger – enter messages into the system log

SYNOPSIS

logger [options] [*message*]

DESCRIPTION

logger makes entries in the system log.

When the optional *message* argument is present, it is written to the log. If it is not present, and the **–f** option is not given either, then standard input is logged.

OPTIONS

–d, **––udp**

Use datagrams (UDP) only. By default the connection is tried to the syslog port defined in /etc/services, which is often 514 .

See also **––server** and **––socket** to specify where to connect.

–e, **––skip-empty**

Ignore empty lines when processing files. An empty line is defined to be a line without any characters. Thus a line consisting only of whitespace is NOT considered empty. Note that when the **––prio–prefix** option is specified, the priority is not part of the line. Thus an empty line in this mode is a line that does not have any characters after the priority prefix (e.g. **<13>**).

–f, **––file** *file*

Log the contents of the specified *file*. This option cannot be combined with a command-line message.

–i Log the PID of the logger process with each line.

––id[**=***id*]

Log the PID of the logger process with each line. When the optional argument *id* is specified, then it is used instead of the logger command's PID. The use of **––id=$$** (PPID) is recommended in scripts that send several messages.

Note that the system logging infrastructure (for example **systemd** when listening on /dev/log) may follow local socket credentials to overwrite the PID specified in the message. **logger**(1) is able to set those socket credentials to the given *id*, but only if you have root permissions and a process with the specified PID exists, otherwise the socket credentials are not modified and the problem is silently ignored.

––journald[**=***file*]

Write a systemd journal entry. The entry is read from the given *file*, when specified, otherwise from standard input. Each line must begin with a field that is accepted by journald; see **systemd.journal-fields**(7) for details. The use of a MESSAGE_ID field is generally a good idea, as it makes finding entries easy. Examples:

```
logger ––journald <<end
MESSAGE_ID=67feb6ffbaf24c5cbec13c008dd72309
MESSAGE=The dogs bark, but the caravan goes on.
DOGS=bark
CARAVAN=goes on
end

logger ––journald=entry.txt
```

Notice that **––journald** will ignore values of other options, such as priority. If priority is needed it must be within input, and use PRIORITY field. The simple execution of **journalctl** will display MESSAGE field. Use **journalctl ––output json-pretty** to see rest of the fields.

To include newlines in MESSAGE, specify MESSAGE several times. This is handled as a special case, other fields will be stored as an array in the journal if they appear multiple times.

−−msgid *msgid*
Sets the RFC5424 MSGID field. Note that the space character is not permitted inside of *msgid*. This option is only used if **−−rfc5424** is specified as well; otherwise, it is silently ignored.

−n, **−−server** *server*
Write to the specified remote syslog *server* instead of to the system log socket. Unless **−−udp** or **−−tcp** is specified, **logger** will first try to use UDP, but if this fails a TCP connection is attempted.

−−no−act
Causes everything to be done except for writing the log message to the system log, and removing the connection or the journal. This option can be used together with **−−stderr** for testing purposes.

−−octet−count
Use the RFC 6587 octet counting framing method for sending messages. When this option is not used, the default is no framing on UDP, and RFC6587 non-transparent framing (also known as octet stuffing) on TCP.

−P, **−−port** *port*
Use the specified *port*. When this option is not specified, the port defaults to syslog for udp and to syslog-conn for tcp connections.

−p, **−−priority** *priority*
Enter the message into the log with the specified *priority*. The priority may be specified numerically or as a *facility.level* pair. For example, **−p local3.info** logs the message as informational in the local3 facility. The default is **user.notice**.

−−prio−prefix
Look for a syslog prefix on every line read from standard input. This prefix is a decimal number within angle brackets that encodes both the facility and the level. The number is constructed by multiplying the facility by 8 and then adding the level. For example, **local0.info**, meaning facility=16 and level=6, becomes **<134>**.

If the prefix contains no facility, the facility defaults to what is specified by the **−p** option. Similarly, if no prefix is provided, the line is logged using the *priority* given with **−p**.

This option doesn't affect a command-line message.

−−rfc3164
Use the RFC 3164 BSD syslog protocol to submit messages to a remote server.

−−rfc5424[=*without*]
Use the RFC 5424 syslog protocol to submit messages to a remote server. The optional *without* argument can be a comma-separated list of the following values: **notq**, **notime**, **nohost**.

The **notq** value suppresses the time-quality structured data from the submitted message. The time-quality information shows whether the local clock was synchronized plus the maximum number of microseconds the timestamp might be off. The time quality is also automatically suppressed when **−−sd−id timeQuality** is specified.

The **notime** value (which implies **notq**) suppresses the complete sender timestamp that is in ISO-8601 format, including microseconds and timezone.

The **nohost** value suppresses **gethostname**(2) information from the message header.

The RFC 5424 protocol has been the default for **logger** since version 2.26.

−s, **−−stderr**
Output the message to standard error as well as to the system log.

−−sd−id *name*[*@digits*]
Specifies a structured data element ID for an RFC 5424 message header. The option has to be used before **−−sd−param** to introduce a new element. The number of structured data elements is unlimited. The ID (*name* plus possibly *@digits*) is case-sensitive and uniquely identifies the type and purpose of the element. The same ID must not exist more than once in a message. The *@digits* part is required for user-defined non-standardized IDs.

logger currently generates the **timeQuality** standardized element only. RFC 5424 also describes the elements **origin** (with parameters ip, enterpriseId, software and swVersion) and **meta** (with parameters sequenceId, sysUpTime and language). These element IDs may be specified without the *@digits* suffix.

−−sd−param *name*="*value*"
Specifies a structured data element parameter, a name and value pair. The option has to be used after **−−sd−id** and may be specified more than once for the same element. Note that the quotation marks around *value* are required and must be escaped on the command line.

```
logger −−rfc5424 −−sd-id zoo@123            \
       −−sd-param tiger=\"hungry\"  \
       −−sd-param zebra=\"running\" \
       −−sd-id manager@123         \
       −−sd-param onMeeting=\"yes\" \
       "this is message"
```

produces:

```
<13>1 2015-10-01T14:07:59.168662+02:00 ws kzak - - [timeQuality tzKnown="1" isSynced="1" syncAccura
```

−S, **−−size** *size*
Sets the maximum permitted message size to *size*. The default is 1KiB characters, which is the limit traditionally used and specified in RFC 3164. With RFC 5424, this limit has become flexible. A good assumption is that RFC 5424 receivers can at least process 4KiB messages.

Most receivers accept messages larger than 1KiB over any type of syslog protocol. As such, the **−−size** option affects logger in all cases (not only when **−−rfc5424** was used).

Note: the message-size limit limits the overall message size, including the syslog header. Header sizes vary depending on the selected options and the hostname length. As a rule of thumb, headers are usually not longer than 50 to 80 characters. When selecting a maximum message size, it is important to ensure that the receiver supports the max size as well, otherwise messages may become truncated. Again, as a rule of thumb two to four KiB message size should generally be OK, whereas anything larger should be verified to work.

−−socket−errors[=*mode*]
Print errors about Unix socket connections. The *mode* can be a value of **off**, **on**, or **auto**. When the mode is auto logger will detect if the init process is systemd, and if so assumption is made /dev/log can be used early at boot. Other init systems lack of /dev/log will not cause errors that is identical with messaging using **openlog**(3) system call. The **logger**(1) before version 2.26 used openlog, and hence was unable to detected loss of messages sent to Unix sockets.

The default mode is **auto**. When errors are not enabled lost messages are not communicated and will result to successful return value of **logger**(1) invocation.

−T, **−−tcp**
Use stream (TCP) only. By default the connection is tried to the *syslog-conn* port defined in /etc/services, which is often *601*.

See also **−−server** and **−−socket** to specify where to connect.

−t, **−−tag** *tag*
Mark every line to be logged with the specified *tag*. The default tag is the name of the user logged in on the terminal (or a user name based on effective user ID).

−u, **−−socket** *socket*
Write to the specified *socket* instead of to the system log socket.

−− End the argument list. This allows the *message* to start with a hyphen (−).

−V, **−−version**
Display version information and exit.

−h, **−−help**
Display help text and exit.

RETURN VALUE

The **logger** utility exits 0 on success, and >0 if an error occurs.

FACILITIES AND LEVELS

Valid facility names are:

auth	
authpriv	for security information of a sensitive nature
cron	
daemon	
ftp	
kern	cannot be generated from userspace process, automatically converted to **user**
lpr	
mail	
news	
syslog	
user	
uucp	
local0 **to** **local7**	
security	deprecated synonym for **auth**

Valid level names are:

emerg	
alert	
crit	
err	
warning	
notice	
info	
debug	
panic	deprecated synonym for **emerg**
error	deprecated synonym for **err**
warn	deprecated synonym for **warning**

For the priority order and intended purposes of these facilities and levels, see **syslog**(3).

EXAMPLES

logger System rebooted
logger −p local0.notice −t HOSTIDM −f /dev/idmc
logger −n loghost.example.com System rebooted

SEE ALSO

journalctl(1), **syslog**(3), **systemd.journal-fields**(7)

STANDARDS

The **logger** command is expected to be IEEE Std 1003.2 ("POSIX.2") compatible.

AUTHORS

The **logger** command was originally written by University of California in 1983-1993 and later rewritten by Karel Zak ⟨kzak@redhat.com⟩, Rainer Gerhards ⟨rgerhards@adiscon.com⟩ and Sami Kerola ⟨kerolasa@iki.fi⟩.

AVAILABILITY

The logger command is part of the util-linux package and is available from Linux Kernel Archive ⟨https://www.kernel.org/pub/linux/utils/util-linux/⟩.