NAME

systemd-cat - Connect a pipeline or program's output with the journal

SYNOPSIS

systemd-cat [OPTIONS...] [COMMAND] [ARGUMENTS...]

systemd-cat [OPTIONS...]

DESCRIPTION

systemd-cat may be used to connect the standard input and output of a process to the journal, or as a filter tool in a shell pipeline to pass the output the previous pipeline element generates to the journal.

If no parameter is passed, **systemd-cat** will write everything it reads from standard input (stdin) to the journal.

If parameters are passed, they are executed as command line with standard output (stdout) and standard error output (stderr) connected to the journal, so that all it writes is stored in the journal.

OPTIONS

The following options are understood:

-h, **--help**
Print a short help text and exit.

--version
Print a short version string and exit.

-t, **--identifier=**
Specify a short string that is used to identify the logging tool. If not specified, no identification string is written to the journal.

-p, **--priority=**
Specify the default priority level for the logged messages. Pass one of "emerg", "alert", "crit", "err", "warning", "notice", "info", "debug", or a value between 0 and 7 (corresponding to the same named levels). These priority values are the same as defined by **syslog(3)**. Defaults to "info". Note that this simply controls the default, individual lines may be logged with different levels if they are prefixed accordingly. For details, see **--level-prefix=** below.

--level-prefix=
Controls whether lines read are parsed for syslog priority level prefixes. If enabled (the default), a line prefixed with a priority prefix such as "<5>" is logged at priority 5 ("notice"), and similar for the other priority levels. Takes a boolean argument.

EXIT STATUS

On success, 0 is returned, a non-zero failure code otherwise.

EXAMPLES

Example 1. Invoke a program

This calls /bin/ls with standard output and error connected to the journal:

```
# systemd-cat ls
```

Example 2. Usage in a shell pipeline

This builds a shell pipeline also invoking /bin/ls and writes the output it generates to the journal:

```
# ls | systemd-cat
```

Even though the two examples have very similar effects the first is preferable since only one process is running at a time, and both stdout and stderr are captured while in the second example, only stdout is captured.

SEE ALSO

systemd(1), **systemctl(1)**, **logger(1)**