NAME
systemd-notify - Notify service manager about start-up completion and other daemon status changes

SYNOPSIS
systemd-notify [OPTIONS...] [VARIABLE=VALUE...]

DESCRIPTION
systemd-notify may be called by daemon scripts to notify the init system about status changes. It can be used to send arbitrary information, encoded in an environment-block-like list of strings. Most importantly, it can be used for start-up completion notification.

This is mostly just a wrapper around **sd_notify()** and makes this functionality available to shell scripts. For details see **sd_notify(3)**.

The command line may carry a list of environment variables to send as part of the status update.

Note that systemd will refuse reception of status updates from this command unless *NotifyAccess=all* is set for the service unit this command is called from.

OPTIONS
The following options are understood:

--ready
Inform the init system about service start-up completion. This is equivalent to **systemd-notify READY=1**. For details about the semantics of this option see **sd_notify(3)**.

--pid=
Inform the init system about the main PID of the daemon. Takes a PID as argument. If the argument is omitted, the PID of the process that invoked **systemd-notify** is used. This is equivalent to **systemd-notify MAINPID=$PID**. For details about the semantics of this option see **sd_notify(3)**.

--status=
Send a free-form status string for the daemon to the init systemd. This option takes the status string as argument. This is equivalent to **systemd-notify STATUS=...**. For details about the semantics of this option see **sd_notify(3)**.

--booted
Returns 0 if the system was booted up with systemd, non-zero otherwise. If this option is passed, no message is sent. This option is hence unrelated to the other options. For details about the semantics of this option, see **sd_booted(3)**. An alternate way to check for this state is to call **systemctl(1)** with the **is-system-running** command. It will return "offline" if the system was not booted with systemd.

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

--version
Print a short version string and exit.

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

EXAMPLE
Example 1. Start-up Notification and Status Updates

A simple shell daemon that sends start-up notifications after having set up its communication channel. During runtime it sends further status updates to the init system:

```
#!/bin/bash

mkfifo /tmp/waldo
systemd-notify --ready --status="Waiting for data..."

while : ; do
read a < /tmp/waldo
systemd-notify --status="Processing $a"
```

Do something with $a ...

systemd-notify --status="Waiting for data..."
done

SEE ALSO
systemd(1), systemctl(1), systemd.unit(5), sd_notify(3), sd_booted(3)