NAME

sssd − System Security Services Daemon

SYNOPSIS

sssd [*options*]

DESCRIPTION

SSSD provides a set of daemons to manage access to remote directories and authentication mechanisms. It provides an NSS and PAM interface toward the system and a pluggable backend system to connect to multiple different account sources as well as D−Bus interface. It is also the basis to provide client auditing and policy services for projects like FreeIPA. It provides a more robust database to store local users as well as extended user data.

OPTIONS

−d,**−−debug−level** *LEVEL*

SSSD supports two representations for specifying the debug level. The simplest is to specify a decimal value from 0−9, which represents enabling that level and all lower−level debug messages. The more comprehensive option is to specify a hexadecimal bitmask to enable or disable specific levels (such as if you wish to suppress a level).

Please note that each SSSD service logs into its own log file. Also please note that enabling "debug_level" in the "[sssd]" section only enables debugging just for the sssd process itself, not for the responder or provider processes. The "debug_level" parameter should be added to all sections that you wish to produce debug logs from.

In addition to changing the log level in the config file using the "debug_level" parameter, which is persistent, but requires SSSD restart, it is also possible to change the debug level on the fly using the **sss_debuglevel**(8) tool.

Currently supported debug levels:

0, *0x0010*: Fatal failures. Anything that would prevent SSSD from starting up or causes it to cease running.

1, *0x0020*: Critical failures. An error that doesn't kill SSSD, but one that indicates that at least one major feature is not going to work properly.

2, *0x0040*: Serious failures. An error announcing that a particular request or operation has failed.

3, *0x0080*: Minor failures. These are the errors that would percolate down to cause the operation failure of 2.

4, *0x0100*: Configuration settings.

5, *0x0200*: Function data.

6, *0x0400*: Trace messages for operation functions.

7, *0x1000*: Trace messages for internal control functions.

8, *0x2000*: Contents of function−internal variables that may be interesting.

9, *0x4000*: Extremely low−level tracing information.

9, *0x20000*: Performance and statistical data, please note that due to the way requests are processed

internally the logged execution time of a request might be longer than it actually was.

10, *0x10000*: Even more low−level libldb tracing information. Almost never really required.

To log required bitmask debug levels, simply add their numbers together as shown in following examples:

Example: To log fatal failures, critical failures, serious failures and function data use 0x0270.

Example: To log fatal failures, configuration settings, function data, trace messages for internal control functions use 0x1310.

Note: The bitmask format of debug levels was introduced in 1.7.0.

Default: 0x0070 (i.e. fatal, critical and serious failures; corresponds to setting 2 in decimal notation)

−−debug−timestamps=*mode*
1: Add a timestamp to the debug messages

0: Disable timestamp in the debug messages

Default: 1

−−debug−microseconds=*mode*
1: Add microseconds to the timestamp in debug messages

0: Disable microseconds in timestamp

Default: 0

−−logger=*value*
Location where SSSD will send log messages.

stderr: Redirect debug messages to standard error output.

files: Redirect debug messages to the log files. By default, the log files are stored in /var/log/sssd and there are separate log files for every SSSD service and domain.

journald: Redirect debug messages to systemd−journald

Default: not set (fall back to journald if available, otherwise to stderr)

−D,**−−daemon**
Become a daemon after starting up.

−i,**−−interactive**
Run in the foreground, don't become a daemon.

−c,**−−config**
Specify a non−default config file. The default is /etc/sssd/sssd.conf. For reference on the config file syntax and options, consult the **sssd.conf**(5) manual page.

−?,**−−help**
Display help message and exit.

−−version
Print version number and exit.

SIGNALS

SIGTERM/SIGINT
: Informs the SSSD to gracefully terminate all of its child processes and then shut down the monitor.

SIGHUP
: Tells the SSSD to stop writing to its current debug file descriptors and to close and reopen them. This is meant to facilitate log rolling with programs like logrotate.

SIGUSR1
: Tells the SSSD to simulate offline operation for the duration of the “offline_timeout” parameter. This is useful for testing. The signal can be sent to either the sssd process or any sssd_be process directly.

SIGUSR2
: Tells the SSSD to go online immediately. This is useful for testing. The signal can be sent to either the sssd process or any sssd_be process directly.

SIGRTMIN+1
: Tells the SSSD to reschedule the periodic tasks. The internal watchdog sends this signal to the providers when a clock shift is detected although it can be sent to any sssd_be process directly.

EXIT STATUS

0
: SSSD was shutdown gracefully.

1
: Bad configuration or command line option.

2
: Memory allocation error.

6
: SSSD is already running.

Other codes
: Other codes denote different errors, most probably about missing required access rights. See SSSD and system logs for details.

NOTES

If the environment variable SSS_NSS_USE_MEMCACHE is set to "NO", client applications will not use the fast in−memory cache.

If the environment variable SSS_LOCKFREE is set to "NO", requests from multiple threads of a single application will be serialized.

SEE ALSO

sssd(8), **sssd.conf**(5), **sssd-ldap**(5), **sssd-ldap-attributes**(5), **sssd-krb5**(5), **sssd-simple**(5), **sssd-ipa**(5), **sssd-ad**(5), **sssd-idp**(5), **sssd-sudo**(5), **sssd-session-recording**(5), **sss_cache**(8), **sss_debuglevel**(8), **sss_obfuscate**(8), **sss_seed**(8), **sssd_krb5_locator_plugin**(8), **sss_ssh_authorizedkeys**(1), **sss_ssh_knownhosts**(1), **sssd-ifp**(5), **pam_sss**(8). **sss_rpcidmapd**(5) **sssd-systemtap**(5)

AUTHORS

The SSSD upstream − https://github.com/SSSD/sssd/