

sd_bus_negotiate_fds



- [Description](#)
 - [Return Value](#)
 - [Errors](#)
 - [History](#)
 - [See Also](#)
-

systemd

sd_bus_negotiate_fds

3

sd_bus_negotiate_fds

sd_bus_negotiate_timestamp

sd_bus_negotiate_creds

sd_bus_get_creds_mask

Control feature negotiation on bus connections

```
#include <systemd/sd-bus.h>
```

```
int
```

```
sd_bus_negotiate_fds
```

```
sd_bus *
```

```
bus
```

```
int
```

```
b
```

```
int
```

sd_bus_negotiate_timestamp

sd_bus *

bus

int

b

int

sd_bus_negotiate_creds

sd_bus *

bus

int

b

uint64_t

mask

int

sd_bus_get_creds_mask

sd_bus *

bus

uint64_t *

mask

Description

`sd_bus_negotiate_fds()` controls whether file descriptor passing shall be negotiated for the specified bus connection. It takes a bus object and a boolean, which, when true, enables file descriptor passing, and, when false, disables it. Note that not all transports and servers support file descriptor passing. In particular, networked transports generally do not support file descriptor passing. To find out whether file descriptor passing is available after negotiation, use

`sd_bus_can_send(3)` and pass `SD_BUS_TYPE_UNIX_FD`. Note that file descriptor passing is always enabled for both sending and receiving or for neither, but never only in one direction. By default, file descriptor passing is negotiated for all connections.

`sd_bus_negotiate_timestamp()` controls whether implicit sender timestamps shall be attached automatically to all incoming messages. Takes a bus object and a boolean, which, when true, enables timestamping, and, when false, disables it. Use `sd_bus_message_get_monotonic_usec(3)`, `sd_bus_message_get_realtime_usec(3)`, `sd_bus_message_get_seqnum(3)` to query the timestamps of incoming messages. If negotiation is disabled or not supported, these calls will fail with `-ENODATA`. Note that currently no transports support timestamping of messages. By default, message timestamping is not negotiated for connections.

`sd_bus_negotiate_creds()` controls whether and which implicit sender credentials shall be attached automatically to all incoming messages. Takes a bus object and a boolean indicating whether to enable or disable the credential parts encoded in the bit mask value argument. Note that not all transports support attaching sender credentials to messages, or do not support all types of sender credential parameters, or might suppress them under certain circumstances for individual messages. Specifically, dbus1 only supports `SD_BUS_CREDS_UNIQUE_NAME`. The sender credentials are suitable for authorization decisions. By default, only `SD_BUS_CREDS_WELL_KNOWN_NAMES` and `SD_BUS_CREDS_UNIQUE_NAME` are enabled. In fact, these two credential fields are always sent along and cannot be turned off.

`sd_bus_get_creds_mask()` returns the set of sender credentials that was negotiated to be attached to all incoming messages in `mask`. This value is an upper boundary only. Hence, always make sure to explicitly check which credentials are attached to a specific message before using it.

The `sd_bus_negotiate_fds()` function may be called only before the connection has been started with `sd_bus_start(3)`. Both `sd_bus_negotiate_timestamp()` and `sd_bus_negotiate_creds()` may also be called after a connection has been set up. Note that, when operating on a connection that is shared between multiple components of the same program (for example via `sd_bus_default(3)`), it is highly recommended to only enable additional per message metadata fields, but never disable them again, in order not to disable functionality needed by other components.

Return Value

On success, these functions return a non-negative integer. On failure, they return a negative errno-style error code.

Errors

Returned errors may indicate the following problems:

- EPERM

The bus connection has already been started.

- EINVAL

An argument is invalid.

- ENOPKG

The bus cannot be resolved.

- ECHILD

The bus was created in a different process, library or module instance.

History

`sd_bus_negotiate_fds()`, `sd_bus_negotiate_timestamp()`, and `sd_bus_negotiate_creds()` were added in version 212.

`sd_bus_get_creds_mask()` was added in version 246.

See Also

`systemd(1)`, `sd-bus(3)`, `sd_bus_start(3)`, `sd_bus_can_send(3)`, `sd_bus_message_get_monotonic_usec(3)`, `sd_bus_message_get_realtime_usec(3)`, `sd_bus_message_get_seqnum(3)`, `sd_bus_message_get_creds(3)`

<div><div></div><div></div><div></div></div>	#2	
<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	2026-10-31 14:51:55 UTC
<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	2026-10-31 14:51:56 UTC