

NAME

sasl_client_step - Cyrus SASL documentation

SYNOPSIS

```
#include <sasl/sasl.h>
```

```
int sasl_client_step(sasl_conn_t *conn,  
    const char *serverin,  
    unsigned serverinlen,  
    sasl_interact_t ** prompt_need,  
    const char ** clientout,  
    unsigned * clientoutlen);
```

DESCRIPTION

```
int sasl_client_step(sasl_conn_t *conn,
```

```
const char *serverin,
```

```
unsigned serverinlen,
```

```
sasl_interact_t ** prompt_need,
```

```
const char ** clientout,
```

```
unsigned * clientoutlen);
```

sasl_client_step() performs a step in the authentication negotiation. It returns **SASL_OK** if the whole negotiation is successful and **SASL_CONTINUE** if this step is ok but at least one more step is needed. A client should not assume an authentication negotiation is successful just because the server signaled success via protocol (i.e. if the server said ". OK Authentication succeeded" in IMAP, **sasl_client_step** should still be called one more time with a *serverinlen* of zero.

If **SASL_INTERACT** is returned the library needs some values to be filled in before it can proceed. *The prompt_need* structure will be filled in with requests. The application should fulfill these requests and call **sasl_client_start** again with identical parameters (the *prompt_need* parameter will be the same pointer as before but filled in by the application).

Parameters

- ⊕ **conn** - is the SASL connection context
- ⊕ **serverin** - is the data given by the server (decoded if the protocol encodes requests sent over the wire)
- ⊕ **serverinlen** - is the length of *serverin*
- ⊕ **clientout** - is created. It is the initial client response to send to the server. It is the job of the client to send it over the network to the server. Any protocol specific encoding (such as base64 encoding) necessary needs to be done by the client.
- ⊕ **clientoutlen** - length of *clientout*.

RETURN VALUE

SASL callback functions should return SASL return codes. See `sasl.h` for a complete list.

SASL_CONTINUE indicates success and that there are more steps needed in the authentication.

SASL_OK indicates that the authentication is complete.

Other return codes indicate errors and should either be handled or the authentication session should be quit.

SEE ALSO

RFC 4422, `saslman:sasl(3)`, `sasl_callbacks(3)`, `sasl_client_init(3)`, `sasl_client_new(3)`, `sasl_client_start(3)`, `sasl_errors(3)`

AUTHOR

The Cyrus Team

COPYRIGHT

1993-2016, The Cyrus Team