

NAME

sasl_decode - Cyrus SASL documentation

SYNOPSIS

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

```
int sasl_decode(sasl_conn_t *conn,  
               const char * input,  
               unsigned inputlen,  
               const char ** output,  
               unsigned * outputlen);
```

DESCRIPTION

```
int sasl_decode(sasl_conn_t *conn,  
  
const char * input,  
  
unsigned inputlen,  
  
const char ** output,  
  
unsigned * outputlen);
```

sasl_decode decodes `data` received. After successful authentication this function should be called on all `data` received. It decodes the data from encrypted or signed form to plain data. If there was no security layer negotiated the *output* is identical to the *input*.

Parameters

- ⊗ **conn** - is the SASL connection context
- ⊗ **output** - contains the decoded data and is allocated/freed by the library.
- ⊗ **outputlen** - length of *output*.

One should not give `sasl_decode` more `data` than the negotiated *maxbufsize* (see `sasl_getprop(3)`).

Note that `sasl_decode` can succeed and `outputlen` can be zero. If this is the case simply wait for more `data` and call `sasl_decode` again.

RETURN VALUE

SASL callback functions should return SASL return codes. See `sasl.h` for a complete list. **SASL_OK** indicates success.

Other return codes indicate errors and should be handled.

SEE ALSO

RFC 4422, `saslman:sasl(3)`, `sasl_encode(3)`, `sasl_errors(3)`

AUTHOR

The Cyrus Team

COPYRIGHT

1993-2016, The Cyrus Team