

**NAME**

sasl\_encodev - Cyrus SASL documentation

**SYNOPSIS**

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

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

```
int sasl_encodev(sasl_conn_t *conn,  
                const struct iovec * invec,  
                unsigned numiov,  
                const char ** output,  
                unsigned * outputlen);
```

**DESCRIPTION**

**sasl\_encode** encodes data to be sent to a remote host who we've had a successful authentication session with. If there is a negotiated security the data is signed/encrypted and the output should be sent without modification to the remote host. If there is no security layer the output is identical to the input.

**sasl\_encodev** does the same, but for a *struct iovec* instead of a character buffer.

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

**Parameters**

⊕ **conn** - is the SASL connection context

⊕ **output** - contains the decoded data and is allocated/freed by the library.

⊕ **outputlen** - length of *output*.

```
int sasl_encodev(sasl_conn_t *conn,  
  
const struct iovec *invec,  
  
unsigned numiov,  
  
const char ** output,  
  
unsigned * outputlen);
```

### Parameters

⊕ **conn** - is the SASL connection context

⊕ **output** - contains the decoded data and is allocated/freed by the library.

⊕ **outputlen** - length of *output*.

### 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_decode(3)`, `sasl_errors(3)`

### AUTHOR

The Cyrus Team

### COPYRIGHT

1993-2016, The Cyrus Team