

**NAME**

sasl\_getopt\_t - Cyrus SASL documentation

**SYNOPSIS**

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

```
int sasl_getopt_t(void *context,  
                 const char *plugin_name,  
                 const char *option,  
                 const char ** result,  
                 unsigned * len);
```

**DESCRIPTION**

```
int sasl_getopt_t(void *context,  
  
const char *plugin_name,  
  
const char *option,  
  
const char ** result,  
  
unsigned * len);
```

**sasl\_getopt\_t** is used to retrieve an option, often mechanism specific, from the application. An example of this is requesting what KERBEROS\_V4 srvtab file to use.

**Parameters**

- ⊕ **context** - is the SASL connection context
- ⊕ **plugin\_name** - is the plugin this value is for.
- ⊕ **option** - is a string representing the option. A common option that all server applications should handle is the method for checking plaintext passwords. See the *administrators guide* for a full description of this option.

Memory management of options supplied by the getopt callback should be done by the application, however, any requested option must remain available until the callback is no longer valid. That is, when `sasl_dispose(3)` is called for a the connection it is associated with, or `sasl_done(3)` is called for global callbacks.

**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*, `:/usr/share/doc/saslman/sasl(3)`, `sasl_errors(3)` `sasl_callbacks(3)`

**AUTHOR**

The Cyrus Team

**COPYRIGHT**

1993-2016, The Cyrus Team