

**NAME**

**krb5\_get\_creds**, **krb5\_get\_creds\_opt\_add\_options**, **krb5\_get\_creds\_opt\_alloc**, **krb5\_get\_creds\_opt\_free**, **krb5\_get\_creds\_opt\_set\_enctype**, **krb5\_get\_creds\_opt\_set\_impersonate**, **krb5\_get\_creds\_opt\_set\_options**, **krb5\_get\_creds\_opt\_set\_ticket** - get credentials from the KDC

**LIBRARY**

Kerberos 5 Library (libkrb5, -lkrb5)

**SYNOPSIS**

```
#include <krb5.h>
```

*krb5\_error\_code*

```
krb5_get_creds(krb5_context context, krb5_get_creds_opt opt, krb5_ccache ccache,  
    krb5_const_principal inprinc, krb5_creds **out_creds);
```

*void*

```
krb5_get_creds_opt_add_options(krb5_context context, krb5_get_creds_opt opt, krb5_flags options);
```

*krb5\_error\_code*

```
krb5_get_creds_opt_alloc(krb5_context context, krb5_get_creds_opt *opt);
```

*void*

```
krb5_get_creds_opt_free(krb5_context context, krb5_get_creds_opt opt);
```

*void*

```
krb5_get_creds_opt_set_enctype(krb5_context context, krb5_get_creds_opt opt, krb5_enctype enctype);
```

*krb5\_error\_code*

```
krb5_get_creds_opt_set_impersonate(krb5_context context, krb5_get_creds_opt opt,  
    krb5_const_principal self);
```

*void*

```
krb5_get_creds_opt_set_options(krb5_context context, krb5_get_creds_opt opt, krb5_flags options);
```

*krb5\_error\_code*

```
krb5_get_creds_opt_set_ticket(krb5_context context, krb5_get_creds_opt opt, const Ticket *ticket);
```

**DESCRIPTION**

**krb5\_get\_creds()** fetches credentials specified by *opt* by first looking in the *ccache*, and then it doesn't exist, fetch the credential from the KDC using the *krbtgts* in *ccache*. The credential is returned in

*out\_creds* and should be freed using the function **krb5\_free\_creds()**.

The structure **krb5\_get\_creds\_opt** controls the behavior of **krb5\_get\_creds()**. The structure is opaque to consumers that can set the content of the structure with accessor functions. All accessor functions make copies of the data that is passed into accessor functions, so external consumers free the memory before calling **krb5\_get\_creds()**.

The structure **krb5\_get\_creds\_opt** is allocated with **krb5\_get\_creds\_opt\_alloc()** and freed with **krb5\_get\_creds\_opt\_free()**. The free function also frees the content of the structure set by the accessor functions.

**krb5\_get\_creds\_opt\_add\_options()** and **krb5\_get\_creds\_opt\_set\_options()** adds and sets options to the structure . The possible options to set are

**KRB5\_GC\_CACHED** Only check the *ccache*, don't got out on network to fetch credential.

**KRB5\_GC\_USER\_USER** request a user to user ticket. This options doesn't store the resulting user to user credential in the *ccache*.

**KRB5\_GC\_EXPIRED\_OK**

returns the credential even if it is expired, default behavior is trying to refetch the credential from the KDC.

**KRB5\_GC\_NO\_STORE** Do not store the resulting credentials in the *ccache*.

**krb5\_get\_creds\_opt\_set\_enctype()** sets the preferred encryption type of the application. Don't set this unless you have to since if there is no match in the KDC, the function call will fail.

**krb5\_get\_creds\_opt\_set\_impersonate()** sets the principal to impersonate., Returns a ticket that have the impersonation principal as a client and the requestor as the service. Note that the requested principal have to be the same as the client principal in the *krbtgt*.

**krb5\_get\_creds\_opt\_set\_ticket()** sets the extra ticket used in user-to-user or constrained delegation use case.

## SEE ALSO

**krb5(3)**, **krb5\_get\_credentials(3)**, **krb5.conf(5)**