

**NAME**

Heimdal Kerberos 5 deprecated functions -

**Functions**

KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL **krb5\_change\_password** (krb5\_context context, krb5\_creds \*creds, const char \*newpw, int \*result\_code, krb5\_data \*result\_code\_string, krb5\_data \*result\_string) KRB5\_DEPRECATED\_FUNCTION('Use X instead')

KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL **krb5\_config\_parse\_string\_multi** (krb5\_context context, const char \*string, krb5\_config\_section \*\*res) KRB5\_DEPRECATED\_FUNCTION('Use X instead')

KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL **krb5\_keytype\_to\_etypes** (krb5\_context context, krb5\_keytype keytype, unsigned \*len, krb5\_etype \*\*val) KRB5\_DEPRECATED\_FUNCTION('Use X instead')

KRB5\_LIB\_FUNCTION krb5\_boolean KRB5\_LIB\_CALL **krb5\_etypes\_compatible\_keys** (krb5\_context context, krb5\_etype etype1, krb5\_etype etype2) KRB5\_DEPRECATED\_FUNCTION('Use X instead')

KRB5\_LIB\_FUNCTION void KRB5\_LIB\_CALL **krb5\_free\_data\_contents** (krb5\_context context, krb5\_data \*data) KRB5\_DEPRECATED\_FUNCTION('Use X instead')

KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL **krb5\_keytype\_to\_etypes\_default** (krb5\_context context, krb5\_keytype keytype, unsigned \*len, krb5\_etype \*\*val) KRB5\_DEPRECATED\_FUNCTION('Use X instead')

KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL **krb5\_keytype\_to\_string** (krb5\_context context, krb5\_keytype keytype, char \*\*string) KRB5\_DEPRECATED\_FUNCTION('Use X instead')

KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL **krb5\_string\_to\_keytype** (krb5\_context context, const char \*string, krb5\_keytype \*keytype) KRB5\_DEPRECATED\_FUNCTION('Use X instead')

KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_CALLCONV **krb5\_password\_key\_proc** (krb5\_context context, krb5\_etype type, krb5\_salt salt, krb5\_const\_pointer keyseed, krb5\_keyblock \*\*key) KRB5\_DEPRECATED\_FUNCTION('Use X instead')

KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL **krb5\_get\_in\_tkt\_with\_password** (krb5\_context context, krb5\_flags options, krb5\_addresses \*addrs, const krb5\_etype \*etypes, const krb5\_preauthtype \*pre\_auth\_types, const char \*password, krb5\_ccache ccache, krb5\_creds \*creds, krb5\_kdc\_rep \*ret\_as\_reply) KRB5\_DEPRECATED\_FUNCTION('Use X instead')

KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL **krb5\_get\_in\_tkt\_with\_skey** (krb5\_context context, krb5\_flags options, krb5\_addresses \*addrs, const krb5\_etype \*etypes, const krb5\_preauthtype \*pre\_auth\_types, const krb5\_keyblock \*key, krb5\_ccache ccache, krb5\_creds \*creds, krb5\_kdc\_rep \*ret\_as\_reply) KRB5\_DEPRECATED\_FUNCTION('Use X instead')

KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_CALLCONV **krb5\_keytab\_key\_proc** (krb5\_context context, krb5\_etype enctype, krb5\_salt salt, krb5\_const\_pointer keyseed, krb5\_keyblock \*\*key) KRB5\_DEPRECATED\_FUNCTION('Use X instead')

KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL **krb5\_get\_in\_tkt\_with\_keytab** (krb5\_context

```

context, krb5_flags options, krb5_addresses *addrs, const krb5_etype *etypes, const krb5_preauthtype
*pre_auth_types, krb5_keytab keytab, krb5_ccache ccache, krb5_creds *creds, krb5_kdc_rep
*ret_as_reply) KRB5_DEPRECATED_FUNCTION('Use X instead')
KRB5_LIB_FUNCTION krb5_realm *KRB5_LIB_CALL krb5_princ_realm (krb5_context context,
krb5_principal principal) KRB5_DEPRECATED_FUNCTION('Use X instead')
KRB5_LIB_FUNCTION void KRB5_LIB_CALL krb5_princ_set_realm (krb5_context context,
krb5_principal principal, krb5_realm *realm) KRB5_DEPRECATED_FUNCTION('Use X instead')
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_free_creds_contents (krb5_context
context, krb5_creds *c) KRB5_DEPRECATED_FUNCTION('Use X instead')
KRB5_LIB_FUNCTION void KRB5_LIB_CALL krb5_free_error_string (krb5_context context, char *str)
KRB5_DEPRECATED_FUNCTION('Use X instead')
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_set_error_string (krb5_context
context, const char *fmt,...) __attribute__((format(printf
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_vset_error_string (krb5_context
context, const char *fmt, va_list args) __attribute__((format(printf
KRB5_LIB_FUNCTION void KRB5_LIB_CALL krb5_clear_error_string (krb5_context context)
KRB5_DEPRECATED_FUNCTION('Use X instead')
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_get_cred_from_kdc_opt (krb5_context
context, krb5_ccache ccache, krb5_creds *in_creds, krb5_creds **out_creds, krb5_creds ***ret_tgt,
krb5_flags flags) KRB5_DEPRECATED_FUNCTION('Use X instead')
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_get_cred_from_kdc (krb5_context
context, krb5_ccache ccache, krb5_creds *in_creds, krb5_creds **out_creds, krb5_creds ***ret_tgt)
KRB5_DEPRECATED_FUNCTION('Use X instead')
KRB5_LIB_FUNCTION void KRB5_LIB_CALL krb5_free_unparsed_name (krb5_context context, char
*str) KRB5_DEPRECATED_FUNCTION('Use X instead')
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_generate_subkey (krb5_context
context, const krb5_keyblock *key, krb5_keyblock **subkey) KRB5_DEPRECATED_FUNCTION('Use
X instead')
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_auth_getremoteseqnumber
(krb5_context context, krb5_auth_context auth_context, int32_t *seqnumber)
KRB5_DEPRECATED_FUNCTION('Use X instead')
KRB5_LIB_FUNCTION void KRB5_LIB_CALL krb5_get_init_creds_opt_init (krb5_get_init_creds_opt
*opt) KRB5_DEPRECATED_FUNCTION('Use X instead')
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_get_init_creds_opt_get_error
(krb5_context context, krb5_get_init_creds_opt *opt, KRB_ERROR **error)
KRB5_DEPRECATED_FUNCTION('Use X instead')
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_c_etype_compare (krb5_context
context, krb5_etype e1, krb5_etype e2, krb5_boolean *similar)
KRB5_DEPRECATED_FUNCTION('Use X instead')

```

**Detailed Description****Function Documentation**

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL krb5\_auth\_getremoteseqnumber**

(krb5\_context context, krb5\_auth\_context auth\_context, int32\_t \* seqnumber)

Deprecated: use krb5\_auth\_con\_getremoteseqnumber()

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL krb5\_c\_enctype\_compare** (krb5\_context context, krb5\_enctype e1, krb5\_enctype e2, krb5\_boolean \* similar)

Deprecated: keytypes doesn't exist, they are really encetypes.

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL krb5\_change\_password** (krb5\_context context, krb5\_creds \* creds, const char \* newpw, int \* result\_code, krb5\_data \* result\_code\_string, krb5\_data \* result\_string)

Deprecated: krb5\_change\_password() is deprecated, use krb5\_set\_password().

**Parameters:**

*context* a Kerberos context

*creds*

*newpw*

*result\_code*

*result\_code\_string*

*result\_string*

**Returns:**

On success password is changed.

@

**KRB5\_LIB\_FUNCTION void KRB5\_LIB\_CALL krb5\_clear\_error\_string** (krb5\_context context)

Clear the error message returned by krb5\_get\_error\_string().

Deprecated: use krb5\_clear\_error\_message()

**Parameters:**

*context* Kerberos context

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL krb5\_config\_parse\_string\_multi**

(krb5\_context context, const char \* string, krb5\_config\_section \*\* res)

Deprecated: configuration files are not strings

**KRB5\_LIB\_FUNCTION** `krb5_boolean` **KRB5\_LIB\_CALL** `krb5_encypes_compatible_keys`  
 (`krb5_context` context, `krb5_etype` etype1, `krb5_etype` etype2)  
 Deprecated: keytypes doesn't exist, they are really encypes.

**KRB5\_LIB\_FUNCTION** `krb5_error_code` **KRB5\_LIB\_CALL** `krb5_free_creds_contents` (`krb5_context`  
 context, `krb5_creds` \* c)  
 Deprecated: use `krb5_free_cred_contents()`

**KRB5\_LIB\_FUNCTION** `void` **KRB5\_LIB\_CALL** `krb5_free_data_contents` (`krb5_context` context,  
`krb5_data` \* data)  
 Same as `krb5_data_free()`. MIT compat.

Deprecated: use `krb5_data_free()`.

**Parameters:**

*context* Kerberos 5 context.

*data* `krb5_data` to free.

**KRB5\_LIB\_FUNCTION** `void` **KRB5\_LIB\_CALL** `krb5_free_error_string` (`krb5_context` context, `char` \*  
 str)  
 Free the error message returned by `krb5_get_error_string()`.

Deprecated: use `krb5_free_error_message()`

**Parameters:**

*context* Kerberos context

*str* error message to free

**KRB5\_LIB\_FUNCTION** `void` **KRB5\_LIB\_CALL** `krb5_free_unparsed_name` (`krb5_context` context, `char` \*  
 str)  
 Deprecated: use `krb5_xfree()`.

**KRB5\_LIB\_FUNCTION** `krb5_error_code` **KRB5\_LIB\_CALL** `krb5_generate_subkey` (`krb5_context`  
 context, `const` `krb5_keyblock` \* key, `krb5_keyblock` \*\* subkey)  
 Deprecated: use `krb5_generate_subkey_extended()`

**KRB5\_LIB\_FUNCTION** `krb5_error_code` **KRB5\_LIB\_CALL** `krb5_get_cred_from_kdc` (`krb5_context`  
 context, `krb5_ccache` ccache, `krb5_creds` \* in\_creds, `krb5_creds` \*\* out\_creds, `krb5_creds` \*\*\* ret\_tgt)  
 Deprecated: use `krb5_get_credentials_with_flags()`.

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL krb5\_get\_cred\_from\_kdc\_opt**  
 (krb5\_context context, krb5\_ccache ccache, krb5\_creds \* in\_creds, krb5\_creds \*\* out\_creds,  
 krb5\_creds \*\*\* ret\_tgts, krb5\_flags flags)

Deprecated: use krb5\_get\_credentials\_with\_flags().

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL krb5\_get\_in\_tkt\_with\_keytab**  
 (krb5\_context context, krb5\_flags options, krb5\_addresses \* addrs, const krb5\_enctype \* etypes, const  
 krb5\_preauthtype \* pre\_auth\_types, krb5\_keytab keytab, krb5\_ccache ccache, krb5\_creds \* creds,  
 krb5\_kdc\_rep \* ret\_as\_reply)

Deprecated: use krb5\_get\_init\_creds() and friends.

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL krb5\_get\_in\_tkt\_with\_password**  
 (krb5\_context context, krb5\_flags options, krb5\_addresses \* addrs, const krb5\_enctype \* etypes, const  
 krb5\_preauthtype \* pre\_auth\_types, const char \* password, krb5\_ccache ccache, krb5\_creds \* creds,  
 krb5\_kdc\_rep \* ret\_as\_reply)

Deprecated: use krb5\_get\_init\_creds() and friends.

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL krb5\_get\_in\_tkt\_with\_skey** (krb5\_context  
 context, krb5\_flags options, krb5\_addresses \* addrs, const krb5\_enctype \* etypes, const  
 krb5\_preauthtype \* pre\_auth\_types, const krb5\_keyblock \* key, krb5\_ccache ccache, krb5\_creds \*  
 creds, krb5\_kdc\_rep \* ret\_as\_reply)

Deprecated: use krb5\_get\_init\_creds() and friends.

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL krb5\_get\_init\_creds\_opt\_get\_error**  
 (krb5\_context context, krb5\_get\_init\_creds\_opt \* opt, KRB\_ERROR \*\* error)

Deprecated: use the new krb5\_init\_creds\_init() and krb5\_init\_creds\_get\_error().

**KRB5\_LIB\_FUNCTION void KRB5\_LIB\_CALL krb5\_get\_init\_creds\_opt\_init** (krb5\_get\_init\_creds\_opt  
 \* opt)

Deprecated: use krb5\_get\_init\_creds\_opt\_alloc().

The reason **krb5\_get\_init\_creds\_opt\_init()** is deprecated is that krb5\_get\_init\_creds\_opt is a static structure and for ABI reason it can't grow, ie can't add new functionality.

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_CALLCONV krb5\_keytab\_key\_proc** (krb5\_context  
 context, krb5\_enctype enctype, krb5\_salt salt, krb5\_const\_pointer keyseed, krb5\_keyblock \*\* key)

Deprecated: use krb5\_get\_init\_creds() and friends.

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL krb5\_keytype\_to\_encypes** (krb5\_context  
 context, krb5\_keytype keytype, unsigned \* len, krb5\_enctype \*\* val)

Deprecated: keytypes doesn't exist, they are really enctypees.

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL krb5\_keytype\_to\_encypes\_default**  
(krb5\_context context, krb5\_keytype keytype, unsigned \* len, krb5\_encype \*\* val)

Deprecated: keytypes doesn't exist, they are really enctypees.

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL krb5\_keytype\_to\_string** (krb5\_context context, krb5\_keytype keytype, char \*\* string)

Deprecated: keytypes doesn't exist, they are really enctypees in most cases, use krb5\_encype\_to\_string().

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_CALLCONV krb5\_password\_key\_proc** (krb5\_context context, krb5\_encype type, krb5\_salt salt, krb5\_const\_pointer keyseed, krb5\_keyblock \*\* key)

Deprecated: use krb5\_get\_init\_creds() and friends.

**KRB5\_LIB\_FUNCTION krb5\_realm\* KRB5\_LIB\_CALL krb5 Princ\_realm** (krb5\_context context, krb5\_principal principal)

Deprecated: use krb5\_principal\_get\_realm()

**KRB5\_LIB\_FUNCTION void KRB5\_LIB\_CALL krb5 Princ\_set\_realm** (krb5\_context context, krb5\_principal principal, krb5\_realm \* realm)

Deprecated: use krb5\_principal\_set\_realm()

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL krb5\_set\_error\_string** (krb5\_context context, const char \* fmt, ...)

Set the error message returned by krb5\_get\_error\_string().

Deprecated: use krb5\_get\_error\_message()

#### Parameters:

*context* Kerberos context

*fmt* error message to free

#### Returns:

Return an error code or 0.

**KRB5\_LIB\_FUNCTION krb5\_error\_code KRB5\_LIB\_CALL krb5\_string\_to\_keytype** (krb5\_context context, const char \* string, krb5\_keytype \* keytype)

Deprecated: keytypes doesn't exist, they are really enctypees in most cases, use krb5\_string\_to\_encype().

**KRB5\_LIB\_FUNCTION** `krb5_error_code` **KRB5\_LIB\_CALL** `krb5_vset_error_string` (`krb5_context` `context`, `const char * fmt`, `va_list args`)

Set the error message returned by `krb5_get_error_string()`, deprecated, use `krb5_set_error_message()`.

Deprecated: use `krb5_vset_error_message()`

**Parameters:**

*context* Kerberos context

*msg* error message to free

**Returns:**

Return an error code or 0.