

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.