NAME

Heimdal Kerberos 5 support functions -

Functions

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_acl_match_string** (krb5_context context, const char *string, const char *format,...)
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_acl_match_file** (krb5_context context, const char *file, const char *format,...)
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_config_parse_file_multi** (krb5_context context, const char *fname, krb5_config_section **res)
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_config_file_free** (krb5_context context, krb5_config_section *s)
KRB5_LIB_FUNCTION const krb5_config_binding *KRB5_LIB_CALL **krb5_config_get_list** (krb5_context context, const krb5_config_section *c,...)
KRB5_LIB_FUNCTION const krb5_config_binding *KRB5_LIB_CALL **krb5_config_vget_list** (krb5_context context, const krb5_config_section *c, va_list args)
KRB5_LIB_FUNCTION const char *KRB5_LIB_CALL **krb5_config_get_string** (krb5_context context, const krb5_config_section *c,...)
KRB5_LIB_FUNCTION const char *KRB5_LIB_CALL **krb5_config_vget_string** (krb5_context context, const krb5_config_section *c, va_list args)
KRB5_LIB_FUNCTION const char *KRB5_LIB_CALL **krb5_config_vget_string_default** (krb5_context context, const krb5_config_section *c, const char *def_value, va_list args)
KRB5_LIB_FUNCTION const char *KRB5_LIB_CALL **krb5_config_get_string_default** (krb5_context context, const krb5_config_section *c, const char *def_value,...)
KRB5_LIB_FUNCTION char **KRB5_LIB_CALL **krb5_config_vget_strings** (krb5_context context, const krb5_config_section *c, va_list args)
KRB5_LIB_FUNCTION char **KRB5_LIB_CALL **krb5_config_get_strings** (krb5_context context, const krb5_config_section *c,...)
KRB5_LIB_FUNCTION void KRB5_LIB_CALL **krb5_config_free_strings** (char **strings)
KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL **krb5_config_vget_bool_default** (krb5_context context, const krb5_config_section *c, krb5_boolean def_value, va_list args)
KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL **krb5_config_vget_bool** (krb5_context context, const krb5_config_section *c, va_list args)
KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL **krb5_config_get_bool_default** (krb5_context context, const krb5_config_section *c, krb5_boolean def_value,...)
KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL **krb5_config_get_bool** (krb5_context context, const krb5_config_section *c,...)
KRB5_LIB_FUNCTION int KRB5_LIB_CALL **krb5_config_vget_time_default** (krb5_context context, const krb5_config_section *c, int def_value, va_list args)
KRB5_LIB_FUNCTION int KRB5_LIB_CALL **krb5_config_vget_time** (krb5_context context, const

krb5_config_section *c, va_list args)
KRB5_LIB_FUNCTION int KRB5_LIB_CALL **krb5_config_get_time_default** (krb5_context context, const krb5_config_section *c, int def_value,...)
KRB5_LIB_FUNCTION int KRB5_LIB_CALL **krb5_config_get_time** (krb5_context context, const krb5_config_section *c,...)
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_expand_hostname** (krb5_context context, const char *orig_hostname, char **new_hostname)
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_expand_hostname_realms** (krb5_context context, const char *orig_hostname, char **new_hostname, char ***realms)
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_free_host_realm** (krb5_context context, krb5_realm *realmlist)
KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL **krb5_kuserok** (krb5_context context, krb5_principal principal, const char *luser)
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_plugin_register** (krb5_context context, enum krb5_plugin_type type, const char *name, void *symbol)

Detailed Description

Function Documentation

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_acl_match_file (krb5_context context, const char * file, const char * format, ...)

krb5_acl_match_file matches ACL format against each line in a file using **krb5_acl_match_string()**. Lines starting with # are treated like comments and ignored.

Parameters:

context Kerberos 5 context.
file file with acl listed in the file.
format format to match.
... parameter to format string.

Returns:

Return an error code or 0.

See also:

krb5_acl_match_string

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_acl_match_string (krb5_context context, const char * string, const char * format, ...)

krb5_acl_match_string matches ACL format against a string.

The ACL format has three format specifiers: s, f, and r. Each specifier will retrieve one argument from

the variable arguments for either matching or storing data. The input string is split up using ’ ’ (space) and ’\t’ (tab) as a delimiter; multiple and ’\t’ in a row are considered to be the same.

List of format specifiers:

- s Matches a string using strcmp(3) (case sensitive).
- f Matches the string with fnmatch(3). Theflags argument (the last argument) passed to the fnmatch function is 0.
- r Returns a copy of the string in the char ** passed in; the copy must be freed with free(3). There is no need to free(3) the string on error: the function will clean up and set the pointer to NULL.

Parameters:

context Kerberos 5 context
string string to match with
format format to match
... parameter to format string

Returns:

Return an error code or 0.

```
char *s;

ret = krb5_acl_match_string(context, ’foo’, ’s’, ’foo’);
if (ret)
   krb5_errx(context, 1, ’acl didn’t match’);
ret = krb5_acl_match_string(context, ’foo foo baz/kaka’,
   ’ss’, ’foo’, &s, ’foo/\*’);
if (ret) {
   // no need to free(s) on error
   assert(s == NULL);
   krb5_errx(context, 1, ’acl didn’t match’);
}
free(s);
```

See also:

krb5_acl_match_file

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_config_file_free (krb5_context

context, krb5_config_section * s)
Free configuration file section, the result of krb5_config_parse_file() and **krb5_config_parse_file_multi()**.

Parameters:
context A Kerberos 5 context
s the configuration section to free

Returns:
returns 0 on successes, otherwise an error code, see krb5_get_error_message()

KRB5_LIB_FUNCTION void KRB5_LIB_CALL krb5_config_free_strings (char ** strings)
Free the resulting strings from krb5_config-get_strings() and **krb5_config_vget_strings()**.

Parameters:
strings strings to free

KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL krb5_config_get_bool (krb5_context context, const krb5_config_section * c, ...)
Like **krb5_config_get_bool()** but with a va_list list of configuration selection.

Configuration value to a boolean value, where yes/true and any non-zero number means TRUE and other value is FALSE.

Parameters:
context A Kerberos 5 context.
c a configuration section, or NULL to use the section from context
... a list of names, terminated with NULL.

Returns:
TRUE or FALSE

KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL krb5_config_get_bool_default (krb5_context context, const krb5_config_section * c, krb5_boolean def_value, ...)
krb5_config_get_bool_default() will convert the configuration option value to a boolean value, where yes/true and any non-zero number means TRUE and other value is FALSE.

Parameters:
context A Kerberos 5 context.
c a configuration section, or NULL to use the section from context

def_value the default value to return if no configuration found in the database.
... a list of names, terminated with NULL.

Returns:
TRUE or FALSE

KRB5_LIB_FUNCTION const krb5_config_binding* KRB5_LIB_CALL krb5_config_get_list (krb5_context context, const krb5_config_section * c, ...)
Get a list of configuration binding list for more processing

Parameters:
context A Kerberos 5 context.
c a configuration section, or NULL to use the section from context
... a list of names, terminated with NULL.

Returns:
NULL if configuration list is not found, a list otherwise

KRB5_LIB_FUNCTION const char* KRB5_LIB_CALL krb5_config_get_string (krb5_context context, const krb5_config_section * c, ...)
Returns a 'const char *' to a string in the configuration database. The string may not be valid after a reload of the configuration database so a caller should make a local copy if it needs to keep the string.

Parameters:
context A Kerberos 5 context.
c a configuration section, or NULL to use the section from context
... a list of names, terminated with NULL.

Returns:
NULL if configuration string not found, a string otherwise

KRB5_LIB_FUNCTION const char* KRB5_LIB_CALL krb5_config_get_string_default (krb5_context context, const krb5_config_section * c, const char * def_value, ...)
Like **krb5_config_get_string()**, but instead of returning NULL, instead return a default value.

Parameters:
context A Kerberos 5 context.
c a configuration section, or NULL to use the section from context
def_value the default value to return if no configuration found in the database.
... a list of names, terminated with NULL.

Returns:

a configuration string

KRB5_LIB_FUNCTION char KRB5_LIB_CALL krb5_config_get_strings (krb5_context context, const krb5_config_section * c, ...)**

Get a list of configuration strings, free the result with **krb5_config_free_strings()**.

Parameters:

context A Kerberos 5 context.

c a configuration section, or NULL to use the section from context

... a list of names, terminated with NULL.

Returns:

TRUE or FALSE

KRB5_LIB_FUNCTION int KRB5_LIB_CALL krb5_config_get_time (krb5_context context, const krb5_config_section * c, ...)

Get the time from the configuration file using a relative time, for example: 1h30s

Parameters:

context A Kerberos 5 context.

c a configuration section, or NULL to use the section from context

... a list of names, terminated with NULL.

Returns:

parsed the time or -1 on error

KRB5_LIB_FUNCTION int KRB5_LIB_CALL krb5_config_get_time_default (krb5_context context, const krb5_config_section * c, int def_value, ...)

Get the time from the configuration file using a relative time, for example: 1h30s

Parameters:

context A Kerberos 5 context.

c a configuration section, or NULL to use the section from context

def_value the default value to return if no configuration found in the database.

... a list of names, terminated with NULL.

Returns:

parsed the time (or def_value on parse error)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_config_parse_file_multi (krb5_context context, const char * fname, krb5_config_section ** res)
Parse a configuration file and add the result into res. This interface can be used to parse several configuration files into one resulting krb5_config_section by calling it repeatably.

Parameters:

context a Kerberos 5 context.
fname a file name to a Kerberos configuration file
res the returned result, must be free with **krb5_free_config_files()**.

Returns:

Return an error code or 0, see krb5_get_error_message().

If the fname starts with '~/' parse configuration file in the current users home directory. The behavior can be disabled and enabled by calling **krb5_set_home_dir_access()**.

KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL krb5_config_vget_bool (krb5_context context, const krb5_config_section * c, va_list args)
krb5_config_get_bool() will convert the configuration option value to a boolean value, where yes/true and any non-zero number means TRUE and other value is FALSE.

Parameters:

context A Kerberos 5 context.
c a configuration section, or NULL to use the section from context
args a va_list of arguments

Returns:

TRUE or FALSE

KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL krb5_config_vget_bool_default (krb5_context context, const krb5_config_section * c, krb5_boolean def_value, va_list args)
Like **krb5_config_get_bool_default()** but with a va_list list of configuration selection.

Configuration value to a boolean value, where yes/true and any non-zero number means TRUE and other value is FALSE.

Parameters:

context A Kerberos 5 context.
c a configuration section, or NULL to use the section from context
def_value the default value to return if no configuration found in the database.

args a va_list of arguments

Returns:
TRUE or FALSE

KRB5_LIB_FUNCTION const krb5_config_binding* KRB5_LIB_CALL krb5_config_vget_list (krb5_context context, const krb5_config_section * c, va_list args)
Get a list of configuration binding list for more processing

Parameters:
context A Kerberos 5 context.
c a configuration section, or NULL to use the section from context
args a va_list of arguments

Returns:
NULL if configuration list is not found, a list otherwise

KRB5_LIB_FUNCTION const char* KRB5_LIB_CALL krb5_config_vget_string (krb5_context context, const krb5_config_section * c, va_list args)
Like **krb5_config_get_string()**, but uses a va_list instead of ...

Parameters:
context A Kerberos 5 context.
c a configuration section, or NULL to use the section from context
args a va_list of arguments

Returns:
NULL if configuration string not found, a string otherwise

KRB5_LIB_FUNCTION const char* KRB5_LIB_CALL krb5_config_vget_string_default (krb5_context context, const krb5_config_section * c, const char * def_value, va_list args)
Like **krb5_config_vget_string()**, but instead of returning NULL, instead return a default value.

Parameters:
context A Kerberos 5 context.
c a configuration section, or NULL to use the section from context
def_value the default value to return if no configuration found in the database.
args a va_list of arguments

Returns:

a configuration string

KRB5_LIB_FUNCTION char KRB5_LIB_CALL krb5_config_vget_strings (krb5_context context, const krb5_config_section * c, va_list args)**
Get a list of configuration strings, free the result with **krb5_config_free_strings()**.

Parameters:

context A Kerberos 5 context.
c a configuration section, or NULL to use the section from context
args a va_list of arguments

Returns:

TRUE or FALSE

KRB5_LIB_FUNCTION int KRB5_LIB_CALL krb5_config_vget_time (krb5_context context, const krb5_config_section * c, va_list args)
Get the time from the configuration file using a relative time, for example: 1h30s

Parameters:

context A Kerberos 5 context.
c a configuration section, or NULL to use the section from context
args a va_list of arguments

Returns:

parsed the time or -1 on error

KRB5_LIB_FUNCTION int KRB5_LIB_CALL krb5_config_vget_time_default (krb5_context context, const krb5_config_section * c, int def_value, va_list args)
Get the time from the configuration file using a relative time.

Like **krb5_config_get_time_default()** but with a va_list list of configuration selection.

Parameters:

context A Kerberos 5 context.
c a configuration section, or NULL to use the section from context
def_value the default value to return if no configuration found in the database.
args a va_list of arguments

Returns:

parsed the time (or def_value on parse error)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_expand_hostname (krb5_context context, const char * orig_hostname, char ** new_hostname)

krb5_expand_hostname() tries to make orig_hostname into a more canonical one in the newly allocated space returned in new_hostname.

Parameters:

context a Keberos context
orig_hostname hostname to canonicalise.
new_hostname output hostname, caller must free hostname with krb5_xfree().

Returns:

Return an error code or 0, see krb5_get_error_message().

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_expand_hostname_realms (krb5_context context, const char * orig_hostname, char ** new_hostname, char * realms)**

krb5_expand_hostname_realms() expands orig_hostname to a name we believe to be a hostname in newly allocated space in new_hostname and return the realms new_hostname is believed to belong to in realms.

Parameters:

context a Keberos context
orig_hostname hostname to canonicalise.
new_hostname output hostname, caller must free hostname with krb5_xfree().
realms output possible realms, is an array that is terminated with NULL. Caller must free with **krb5_free_host_realm()**.

Returns:

Return an error code or 0, see krb5_get_error_message().

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_free_host_realm (krb5_context context, krb5_realm * realmlist)

Free all memory allocated by ‘realmlist’

Parameters:

context A Kerberos 5 context.
realmlist realmlist to free, NULL is ok

Returns:

a Kerberos error code, always 0.

KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL krb5_kuserok (krb5_context context, krb5_principal principal, const char * luser)

This function takes the name of a local user and checks if principal is allowed to log in as that user.

The user may have a ~/.k5login file listing principals that are allowed to login as that user. If that file does not exist, all principals with a first component identical to the username, and a realm considered local, are allowed access.

The .k5login file must contain one principal per line, be owned by user and not be writable by group or other (but must be readable by anyone).

Note that if the file exists, no implicit access rights are given to user@LOCALREALM.

Optionally, a set of files may be put in ~/.k5login.d (a directory), in which case they will all be checked in the same manner as .k5login. The files may be called anything, but files starting with a hash (#) , or ending with a tilde (~) are ignored. Subdirectories are not traversed. Note that this directory may not be checked by other Kerberos implementations.

If no configuration file exists, match user against local domains, ie luser@LOCAL-REALMS-IN-CONFIGURATION-FILES.

Parameters:

context Kerberos 5 context.
principal principal to check if allowed to login
luser local user id

Returns:

returns TRUE if access should be granted, FALSE otherwise.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_plugin_register (krb5_context context, enum krb5_plugin_type type, const char * name, void * symbol)

Register a plugin symbol name of specific type.

Parameters:

context a Keberos context
type type of plugin symbol
name name of plugin symbol
symbol a pointer to the named symbol

Returns:

In case of error a non zero error com_err error is returned and the Kerberos error string is set.