

NAME

krb5_add_et_list, **krb5_add_extra_addresses**, **krb5_add_ignore_addresses**, **krb5_context**,
krb5_free_config_files, **krb5_free_context**, **krb5_get_default_config_files**,
krb5_get_dns_canonize_hostname, **krb5_get_extra_addresses**, **krb5_get_fcache_version**,
krb5_get_ignore_addresses, **krb5_get_kdc_sec_offset**, **krb5_get_max_time_skew**,
krb5_get_use_admin_kdc, **krb5_init_context**, **krb5_init_ets**, **krb5_prepend_config_files**,
krb5_prepend_config_files_default, **krb5_set_config_files**, **krb5_set_dns_canonize_hostname**,
krb5_set_extra_addresses, **krb5_set_fcache_version**, **krb5_set_ignore_addresses**,
krb5_set_max_time_skew, **krb5_set_use_admin_kdc**, - create, modify and delete **krb5_context**
structures

LIBRARY

Kerberos 5 Library (libkrb5, -lkrb5)

SYNOPSIS

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

```
struct krb5_context;
```

krb5_error_code

```
krb5_init_context(krb5_context *context);
```

void

```
krb5_free_context(krb5_context context);
```

void

```
krb5_init_ets(krb5_context context);
```

krb5_error_code

```
krb5_add_et_list(krb5_context context, void (*func)(struct et_list **));
```

krb5_error_code

```
krb5_add_extra_addresses(krb5_context context, krb5_addresses *addresses);
```

krb5_error_code

```
krb5_set_extra_addresses(krb5_context context, const krb5_addresses *addresses);
```

krb5_error_code

```
krb5_get_extra_addresses(krb5_context context, krb5_addresses *addresses);
```

krb5_error_code

krb5_add_ignore_addresses(*krb5_context context, krb5_addresses *addresses*);

krb5_error_code

krb5_set_ignore_addresses(*krb5_context context, const krb5_addresses *addresses*);

krb5_error_code

krb5_get_ignore_addresses(*krb5_context context, krb5_addresses *addresses*);

krb5_error_code

krb5_set_fcache_version(*krb5_context context, int version*);

krb5_error_code

krb5_get_fcache_version(*krb5_context context, int *version*);

void

krb5_set_dns_canonize_hostname(*krb5_context context, krb5_boolean flag*);

krb5_boolean

krb5_get_dns_canonize_hostname(*krb5_context context*);

krb5_error_code

krb5_get_kdc_sec_offset(*krb5_context context, int32_t *sec, int32_t *usec*);

krb5_error_code

krb5_set_config_files(*krb5_context context, char **filenames*);

krb5_error_code

krb5_prepend_config_files(*const char *filelist, char **pq, char ***ret_pp*);

krb5_error_code

krb5_prepend_config_files_default(*const char *filelist, char ***pfilenames*);

krb5_error_code

krb5_get_default_config_files(*char ***pfilenames*);

void

krb5_free_config_files(*char **filenames*);

void

```
krb5_set_use_admin_kdc(krb5_context context, krb5_boolean flag);
```

krb5_boolean

```
krb5_get_use_admin_kdc(krb5_context context);
```

time_t

```
krb5_get_max_time_skew(krb5_context context);
```

krb5_error_code

```
krb5_set_max_time_skew(krb5_context context, time_t time);
```

DESCRIPTION

The **krb5_init_context**() function initializes the *context* structure and reads the configuration file */etc/krb5.conf*.

The structure should be freed by calling **krb5_free_context**() when it is no longer being used.

krb5_init_context() returns 0 to indicate success. Otherwise an errno code is returned. Failure means either that something bad happened during initialization (typically [ENOMEM]) or that Kerberos should not be used [ENXIO].

krb5_init_ets() adds all com_err(3) libs to *context*. This is done by **krb5_init_context**().

krb5_add_et_list() adds a com_err(3) error-code handler *func* to the specified *context*. The error handler must be generated by the re-entrant version of the compile_et(1) program. **krb5_add_extra_addresses**() add a list of addresses that should be added when requesting tickets.

krb5_add_ignore_addresses() add a list of addresses that should be ignored when requesting tickets.

krb5_get_extra_addresses() get the list of addresses that should be added when requesting tickets.

krb5_get_ignore_addresses() get the list of addresses that should be ignored when requesting tickets.

krb5_set_ignore_addresses() set the list of addresses that should be ignored when requesting tickets.

krb5_set_extra_addresses() set the list of addresses that should be added when requesting tickets.

krb5_set_fcache_version() sets the version of file credentials caches that should be used.

krb5_get_fcache_version() gets the version of file credentials caches that should be used.

krb5_set_dns_canonize_hostname() sets if the context is configured to canonicalize hostnames using DNS.

krb5_get_dns_canonize_hostname() returns if the context is configured to canonicalize hostnames using DNS.

krb5_get_kdc_sec_offset() returns the offset between the localtime and the KDC's time. *sec* and *usec* are both optional argument and NULL can be passed in.

krb5_set_config_files() set the list of configuration files to use and re-initialize the configuration from the files.

krb5_prepend_config_files() parse the *filelist* and prepend the result to the already existing list *pq*. The result is returned in *ret_pp* and should be freed with **krb5_free_config_files()**.

krb5_prepend_config_files_default() parse the *filelist* and append that to the default list of configuration files.

krb5_get_default_config_files() get a list of default configuration files.

krb5_free_config_files() free a list of configuration files returned by **krb5_get_default_config_files()**, **krb5_prepend_config_files_default()**, or **krb5_prepend_config_files()**.

krb5_set_use_admin_kdc() sets if all KDC requests should go admin KDC.

krb5_get_use_admin_kdc() gets if all KDC requests should go admin KDC.

krb5_get_max_time_skew() and **krb5_set_max_time_skew()** get and sets the maximum allowed time skew between client and server.

SEE ALSO

errno(2), krb5(3), krb5_config(3), krb5_context(3), kerberos(8)