NAME

krb5_rcache, **krb5_rc_close**, **krb5_rc_default**, **krb5_rc_default_name**, **krb5_rc_default_type**, **krb5_rc_destroy**, **krb5_rc_expunge**, **krb5_rc_get_lifespan**, **krb5_rc_get_name**, **krb5_rc_get_type**, **krb5_rc_initialize**, **krb5_rc_recover**, **krb5_rc_resolve**, **krb5_rc_resolve_full**, **krb5_rc_resolve_type**, **krb5_rc_store**, **krb5_get_server_rcache** - Kerberos 5 replay cache

LIBRARY

Kerberos 5 Library (libkrb5, -lkrb5)

SYNOPSIS

#include <krb5.h>

struct krb5_rcache;

krb5_error_code
krb5_rc_close(*krb5_context context, krb5_rcache id*);

krb5_error_code
krb5_rc_default(*krb5_context context, krb5_rcache *id*);

*const char **
krb5_rc_default_name(*krb5_context context*);

*const char **
krb5_rc_default_type(*krb5_context context*);

krb5_error_code
krb5_rc_destroy(*krb5_context context, krb5_rcache id*);

krb5_error_code
krb5_rc_expunge(*krb5_context context, krb5_rcache id*);

krb5_error_code
krb5_rc_get_lifespan(*krb5_context context, krb5_rcache id, krb5_deltat *auth_lifespan*);

*const char**
krb5_rc_get_name(*krb5_context context, krb5_rcache id*);

*const char**
krb5_rc_get_type(*krb5_context context, krb5_rcache id*);

krb5_error_code
krb5_rc_initialize(*krb5_context context, krb5_rcache id, krb5_deltat auth_lifespan*);

krb5_error_code
krb5_rc_recover(*krb5_context context, krb5_rcache id*);

krb5_error_code
krb5_rc_resolve(*krb5_context context, krb5_rcache id, const char *name*);

krb5_error_code
krb5_rc_resolve_full(*krb5_context context, krb5_rcache *id, const char *string_name*);

krb5_error_code
krb5_rc_resolve_type(*krb5_context context, krb5_rcache *id, const char *type*);

krb5_error_code
krb5_rc_store(*krb5_context context, krb5_rcache id, krb5_donot_replay *rep*);

krb5_error_code
krb5_get_server_rcache(*krb5_context context, const krb5_data *piece, krb5_rcache *id*);

DESCRIPTION

The krb5_rcache structure holds a storage element that is used for data manipulation. The structure contains no public accessible elements.

krb5_rc_initialize() Creates the reply cache *id* and sets it lifespan to *auth_lifespan*. If the cache already exists, the content is destroyed.

SEE ALSO

krb5(3), krb5_data(3), kerberos(8)