NAME

Heimdal Kerberos 5 address functions -

Functions

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_sockaddr2address** (krb5_context context, const struct sockaddr *sa, krb5_address *addr)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_sockaddr2port** (krb5_context context, const struct sockaddr *sa, int16_t *port)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_addr2sockaddr** (krb5_context context, const krb5_address *addr, struct sockaddr *sa, krb5_socklen_t *sa_size, int port)

KRB5_LIB_FUNCTION size_t KRB5_LIB_CALL **krb5_max_sockaddr_size** (void)

KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL **krb5_sockaddr_uninteresting** (const struct sockaddr *sa)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_h_addr2sockaddr** (krb5_context context, int af, const char *addr, struct sockaddr *sa, krb5_socklen_t *sa_size, int port)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_h_addr2addr** (krb5_context context, int af, const char *haddr, krb5_address *addr)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_anyaddr** (krb5_context context, int af, struct sockaddr *sa, krb5_socklen_t *sa_size, int port)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_print_address** (const krb5_address *addr, char *str, size_t len, size_t *ret_len)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_parse_address** (krb5_context context, const char *string, krb5_addresses *addresses)

KRB5_LIB_FUNCTION int KRB5_LIB_CALL **krb5_address_order** (krb5_context context, const krb5_address *addr1, const krb5_address *addr2)

KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL **krb5_address_compare** (krb5_context context, const krb5_address *addr1, const krb5_address *addr2)

KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL **krb5_address_search** (krb5_context context, const krb5_address *addr, const krb5_addresses *addrlist)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_free_address** (krb5_context context, krb5_address *address)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_free_addresses** (krb5_context context, krb5_addresses *addresses)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_copy_address** (krb5_context context, const krb5_address *inaddr, krb5_address *outaddr)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_copy_addresses** (krb5_context context, const krb5_addresses *inaddr, krb5_addresses *outaddr)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_append_addresses** (krb5_context context, krb5_addresses *dest, const krb5_addresses *source)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_make_addrport** (krb5_context context,

krb5_address **res, const krb5_address *addr, int16_t port)
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_address_prefixlen_boundary** (krb5_context context, const krb5_address *inaddr, unsigned long prefixlen, krb5_address *low, krb5_address *high)

Detailed Description

Function Documentation

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_addr2sockaddr (krb5_context context, const krb5_address * addr, struct sockaddr * sa, krb5_socklen_t * sa_size, int port)

krb5_addr2sockaddr sets the 'struct sockaddr sockaddr' from addr and port. The argument sa_size should initially contain the size of the sa and after the call, it will contain the actual length of the address. In case of the sa is too small to fit the whole address, the up to *sa_size will be stored, and then *sa_size will be set to the required length.

Parameters:

context a Keberos context
addr the address to copy the from
sa the struct sockaddr that will be filled in
sa_size pointer to length of sa, and after the call, it will contain the actual length of the address.
port set port in sa.

Returns:

Return an error code or 0. Will return KRB5_PROG_ATYPE_NOSUPP in case address type is not supported.

KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL krb5_address_compare (krb5_context context, const krb5_address * addr1, const krb5_address * addr2)

krb5_address_compare compares the addresses addr1 and addr2. Returns TRUE if the two addresses are the same.

Parameters:

context a Keberos context
addr1 address to compare
addr2 address to compare

Returns:

Return an TRUE is the address are the same FALSE if not

KRB5_LIB_FUNCTION int KRB5_LIB_CALL krb5_address_order (krb5_context context, const krb5_address * addr1, const krb5_address * addr2)

krb5_address_order compares the addresses addr1 and addr2 so that it can be used for sorting addresses. If the addresses are the same address krb5_address_order will return 0. Behavies like memcmp(2).

Parameters:

context a Keberos context
addr1 krb5_address to compare
addr2 krb5_address to compare

Returns:

< 0 if address addr1 in 'less' then addr2. 0 if addr1 and addr2 is the same address, > 0 if addr2 is 'less' then addr1.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_address_prefixlen_boundary (krb5_context context, const krb5_address * inaddr, unsigned long prefixlen, krb5_address * low, krb5_address * high)

Calculate the boundary addresses of 'inaddr'/'prefixlen' and store them in 'low' and 'high'.

Parameters:

context a Keberos context
inaddr address in prefixlen that the bondery searched
prefixlen width of boundery
low lowest address
high highest address

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL krb5_address_search (krb5_context context, const krb5_address * addr, const krb5_addresses * addrlist)

krb5_address_search checks if the address addr is a member of the address set list addrlist .

Parameters:

context a Keberos context.
addr address to search for.
addrlist list of addresses to look in for addr.

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_anyaddr (krb5_context context, int af, struct sockaddr * sa, krb5_socklen_t * sa_size, int port)

krb5_anyaddr fills in a 'struct sockaddr sa' that can be used to bind(2) to. The argument sa_size should initially contain the size of the sa, and after the call, it will contain the actual length of the address.

Parameters:

context a Keberos context
af address family
sa sockaddr
sa_size lenght of sa.
port for to fill into sa.

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_append_addresses (krb5_context context, krb5_addresses * dest, const krb5_addresses * source)

krb5_append_addresses adds the set of addresses in source to dest. While copying the addresses, duplicates are also sorted out.

Parameters:

context a Keberos context
dest destination of copy operation
source adresses that are going to be added to dest

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_copy_address (krb5_context context, const krb5_address * inaddr, krb5_address * outaddr)

krb5_copy_address copies the content of address inaddr to outaddr.

Parameters:

context a Keberos context
inaddr pointer to source address
outaddr pointer to destination address

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_copy_addresses (krb5_context context, const krb5_addresses * inaddr, krb5_addresses * outaddr)

krb5_copy_addresses copies the content of addresses inaddr to outaddr.

Parameters:

context a Keberos context
inaddr pointer to source addresses
outaddr pointer to destination addresses

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_free_address (krb5_context context, krb5_address * address)

krb5_free_address frees the data stored in the address that is alloced with any of the krb5_address functions.

Parameters:

context a Keberos context
address addresss to be freed.

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_free_addresses (krb5_context context, krb5_addresses * addresses)

krb5_free_addresses frees the data stored in the address that is alloced with any of the krb5_address functions.

Parameters:

context a Keberos context
addresses addressses to be freed.

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_h_addr2addr (krb5_context context, int af, const char * haddr, krb5_address * addr)

krb5_h_addr2addr works like krb5_h_addr2sockaddr with the exception that it operates on a krb5_address instead of a struct sockaddr.

Parameters:

context a Keberos context
af address family
haddr host address from struct hostent.
addr returned krb5_address.

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_h_addr2sockaddr (krb5_context context, int af, const char * addr, struct sockaddr * sa, krb5_socklen_t * sa_size, int port)

krb5_h_addr2sockaddr initializes a 'struct sockaddr sa' from af and the 'struct hostent' (see gethostbyname(3)) h_addr_list component. The argument sa_size should initially contain the size of the sa, and after the call, it will contain the actual length of the address.

Parameters:

context a Keberos context
af addresses
addr address
sa returned struct sockaddr
sa_size size of sa
port port to set in sa.

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_make_addrport (krb5_context context, krb5_address ** res, const krb5_address * addr, int16_t port)

Create an address of type KRB5_ADDRESS_ADDRPORT from (addr, port)

Parameters:

context a Keberos context
res built address from addr/port
addr address to use
port port to use

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION size_t KRB5_LIB_CALL krb5_max_sockaddr_size (void)

krb5_max_sockaddr_size returns the max size of the .Li struct sockaddr that the Kerberos library will return.

Returns:

Return an size_t of the maximum struct sockaddr.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_parse_address (krb5_context context, const char * string, krb5_addresses * addresses)

krb5_parse_address returns the resolved hostname in string to the krb5_addresses addresses .

Parameters:

context a Keberos context
string
addresses

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_print_address (const krb5_address * addr, char * str, size_t len, size_t * ret_len)

krb5_print_address prints the address in addr to the string string that have the length len. If ret_len is not NULL, it will be filled with the length of the string if size were unlimited (not including the final NUL) .

Parameters:

addr address to be printed
str pointer string to print the address into
len length that will fit into area pointed to by 'str'.
ret_len return length the str.

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_sockaddr2address (krb5_context context, const struct sockaddr * sa, krb5_address * addr)

krb5_sockaddr2address stores a address a 'struct sockaddr' sa in the krb5_address addr.

Parameters:

context a Keberos context
sa a struct sockaddr to extract the address from

addr an Kerberos 5 address to store the address in.

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_sockaddr2port (krb5_context context, const struct sockaddr * sa, int16_t * port)

krb5_sockaddr2port extracts a port (if possible) from a ’struct sockaddr.

Parameters:

context a Keberos context
sa a struct sockaddr to extract the port from
port a pointer to an int16_t store the port in.

Returns:

Return an error code or 0. Will return KRB5_PROG_ATYPE_NOSUPP in case address type is not supported.

KRB5_LIB_FUNCTION krb5_boolean KRB5_LIB_CALL krb5_sockaddr_uninteresting (const struct sockaddr * sa)

krb5_sockaddr_uninteresting returns TRUE for all .Fa sa that the kerberos library thinks are uninteresting. One example are link local addresses.

Parameters:

sa pointer to struct sockaddr that might be interesting.

Returns:

Return a non zero for uninteresting addresses.