NAME

krb5_acl_match_file, **krb5_acl_match_string** - ACL matching functions

LIBRARY

Kerberos 5 Library (libkrb5, -lkrb5)

SYNOPSIS

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

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

DESCRIPTION

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

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 " " and "\t" as a delimiter; multiple " " and "\t" in a row are considered to be the same.

s Matches a string using strcmp(3) (case sensitive).

f Matches the string with fnmatch(3). The *flags* 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.

All unknown format specifiers cause an error.

EXAMPLES

```
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(3)