NAME

sasl_auxprop - Cyrus SASL documentation

SYNOPSIS

```
#include <sasl/prop.h>

struct propctx *prop_new(unsigned estimate)

int prop_dup(struct propctx *src_ctx,
             struct propctx *dst_ctx)

int prop_request(struct propctx *ctx,
                 const char **names)

const struct propval *prop_get(struct propctx *ctx)

int prop_getnames(struct propctx *ctx, const char **names,
                  struct propval *vals)

void prop_clear(struct propctx *ctx, int requests)

void prop_erase(struct propctx *ctx, const char *name)

void prop_dispose(struct propctx **ctx)

int prop_format(struct propctx *ctx, const char *sep, int seplen,
                char *outbuf, unsigned outmax, unsigned *outlen)

int prop_set(struct propctx *ctx, const char *name,
             const char *value, int vallen)

int prop_setvals(struct propctx *ctx, const char *name,
                 const char **values)
```

DESCRIPTION

SASL auxiliary properties are used to obtain properties from external sources during the authentication process. For example, a mechanism might need to query an LDAP server to obtain the authentication secret. The application probably needs other information from there as well, such as home directory or UID. The auxiliary property interface allows the two to cooperate, and only results in a single query against the LDAP server (or other property sources).

Property lookups take place directly after user canonicalization occurs. Therefore, all requests should be registered with the context before that time. Note that requests can also be registered using the sasl_auxprop_request(3) function. Most of the functions listed below, however, require a property context which can be obtained by calling sasl_auxprop_getctx(3).

API DESCRIPTION

struct *propctx* ***prop_new(unsigned estimate)**

Create a new property context. Probably unnecessary for application developers to call this at any point.

Parameters

- **estimate** - is the estimate of storage needed in total for requests & responses. A value of 0 implies the library default.

Returns

a new property context: *propctx*

int prop_dup(struct *propctx* ***src_ctx, struct** *propctx* ***dst_ctx)**

Duplicate a given property context.

Parameters

- **src_ctx** (*propctx*) - Property context to copy.
- **dst_ctx** (*propctx*) - Destination context to copy into.

Returns

SASL error code.

int prop_request(struct *propctx* ***ctx, const char ****names)**

Add properties to the request list of a given context.

Parameters

- **ctx** (*propctx*) - The property context to add add the request list to.
- **names** - is the NULL-terminated array of property names, and must persist until the requests are cleared or the context is disposed of with a call to *prop_dispose()*.

Returns

SASL error code

const struct *propval* ***prop_get(struct** *propctx* ***ctx)**

Fetch out the property values from a context.

Parameters

- **ctx** (*propctx*) - The property context to fetch from.

Returns

a NULL-terminated array of property values from the given context.

int prop_getnames(struct *propctx* ***ctx, const char **names, struct** *propval* ***vals)**

Fill in a (provided) array of property values based on a list of property names. This implies that the **vals** array is at least as long as the **names** array. The values that are filled in by this call persist until next call to *prop_request()*, *prop_clear()*, or *prop_dispose()* on context. If a name specified here was never requested, then its associated values entry will be set to NULL.

Parameters

- **ctx** (*propctx*) - The property context to fill in.

Returns

number of matching properties that were found, or a SASL error code.

void prop_clear(struct *propctx* ***ctx, int requests)**

Clear values and (optionally) requests from a property context.

Parameters

- **ctx** (*propctx*) - The property context to clear.
- **requests** - set to 1 if the requests should be cleared, 0 otherwise.

void prop_erase(struct *propctx* ***ctx, const char *name)**

Securely erase the value of a property from a context.

Parameters

- **ctx** (*propctx*) - The property context to find the property in.
- **name** - is the name of the property to erase.

void prop_dispose(struct *propctx* ****ctx)**

Disposes of a property context and NULLifys the pointer.

Parameters

- **ctx** (*propctx*) - The property context to clear.

int prop_format(struct *propctx* ***ctx, const char *sep, int seplen, char *outbuf, unsigned outmax, unsigned *outlen)**

Format the requested property names into a string. This not intended for use by the application (*only by auxprop plugins*).

Parameters

- **ctx** (*propctx*) - The property context to extract values from.
- **sep** - the separator to use for the string
- **outbuf** - destination string. Caller must allocate the buffer of length **outmax** (including NUL terminator).
- **outlen** - if non-NULL, will contain the length of the resulting string (excluding NUL terminator).

Returns

SASL error code.

int prop_set(struct *propctx* ***ctx, const char *name, const char *value, int vallen)**

Adds a property value to the context. *This is intended for use by auxprop plugins only.*

Parameters

- **ctx** (*propctx*) - The property context to add a value to.
- **name** - the name of the property to receive the new value, or NULL, which implies that the value will be added to the same property as the last call to either *prop_set()*

or *prop_setvals()*.

- **value** - the new value (of length *vallen*)
- **vallen** - the length of the string *value*.

Returns

SASL error code

int prop_setvals(struct *propctx* ***ctx, const char *name, const char **values)**

Adds multiple values to a single property. *This is intended for use by auxprop plugins only*.

Parameters

- **ctx** (*propctx*) - The property context to add values to.
- **name** - The name of the property to receive the new value, or NULL, which implies that the values will be added to the same property as the last call to either *prop_set()* or *prop_setvals()*.
- **values** - A NULL-terminated array of values to be added the property.

Returns

SASL error code

DATA STRUCTURES

type propval

A struct holding a name and its property values. A name can have zero or more values.

Param name

const char *. Name of this propval. NULL means end of list.

Param values

const char **. List of string values. If property not found, values == NULL. If property found with no values, then *values == NULL

type propctx

A property context.

Param values
List of property values in this context.

Type values
*propval **

RETURN VALUE

The property functions that return an int return SASL error codes. See sasl_errors(3). Those that return pointers will return a valid pointer on success, or NULL on any error.

CONFORMING TO

RFC 4422

SEE ALSO

sasl(3), sasl_errors(3), sasl_auxprop_request(3), sasl_auxprop_getctx(3)

AUTHOR

The Cyrus Team

COPYRIGHT