

## NAME

hx509 name functions -

### Functions

int **hx509\_name\_to\_string** (const hx509\_name name, char \*\*str)  
int **hx509\_name\_cmp** (hx509\_name n1, hx509\_name n2)  
int **hx509\_parse\_name** (hx509\_context context, const char \*str, hx509\_name \*name)  
int **hx509\_name\_copy** (hx509\_context context, const hx509\_name from, hx509\_name \*to)  
int **hx509\_name\_to\_Name** (const hx509\_name from, Name \*to)  
int **hx509\_name\_expand** (hx509\_context context, hx509\_name name, hx509\_env env)  
void **hx509\_name\_free** (hx509\_name \*name)  
int **hx509\_unparse\_der\_name** (const void \*data, size\_t length, char \*\*str)  
int **hx509\_name\_binary** (const hx509\_name name, heim\_octet\_string \*os)  
int **hx509\_name\_is\_null\_p** (const hx509\_name name)  
int **hx509\_general\_name\_unparse** (GeneralName \*name, char \*\*str)

### Detailed Description

See the **PKIX/X.509 Names** for description and examples.

### Function Documentation

**int hx509\_general\_name\_unparse (GeneralName \* name, char \*\* str)**

Unparse the hx509 name in name into a string.

#### Parameters:

*name* the name to print  
*str* an allocated string returns the name in string form

#### Returns:

An hx509 error code, see **hx509\_get\_error\_string()**.

**int hx509\_name\_binary (const hx509\_name name, heim\_octet\_string \* os)**

Convert a hx509\_name object to DER encoded name.

#### Parameters:

*name* name to concert  
*os* data to a DER encoded name, free the resulting octet string with **hx509\_xfree(os->data)**.

#### Returns:

An hx509 error code, see **hx509\_get\_error\_string()**.

**int hx509\_name\_cmp (hx509\_name n1, hx509\_name n2)**

Compare to hx509 name object, useful for sorting.

**Parameters:**

*n1* a hx509 name object.

*n2* a hx509 name object.

**Returns:**

0 the objects are the same, returns > 0 if n1 is 'larger' than n2, < 0 if n1 is 'smaller' than n2.

**int hx509\_name\_copy (hx509\_context context, const hx509\_name from, hx509\_name \* to)**

Copy a hx509 name object.

**Parameters:**

*context* A hx509 context.

*from* the name to copy from

*to* the name to copy to

**Returns:**

An hx509 error code, see **hx509\_get\_error\_string()**.

**int hx509\_name\_expand (hx509\_context context, hx509\_name name, hx509\_env env)**

Expands variables in the name using env. Variables are on the form \${name}. Useful when dealing with certificate templates.

**Parameters:**

*context* A hx509 context.

*name* the name to expand.

*env* environment variable to expand.

**Returns:**

An hx509 error code, see **hx509\_get\_error\_string()**.

Only UTF8String rdnsSequence names are allowed

**void hx509\_name\_free (hx509\_name \* name)**

Free a hx509 name object, upon return \*name will be NULL.

**Parameters:**

*name* a hx509 name object to be freed.

**int hx509\_name\_is\_null\_p (const hx509\_name name)**

Unparse the hx509 name in name into a string.

**Parameters:**

*name* the name to check if its empty/null.

**Returns:**

non zero if the name is empty/null.

**int hx509\_name\_to\_Name (const hx509\_name from, Name \* to)**

Convert a hx509\_name into a Name.

**Parameters:**

*from* the name to copy from

*to* the name to copy to

**Returns:**

An hx509 error code, see **hx509\_get\_error\_string()**.

**int hx509\_name\_to\_string (const hx509\_name name, char \*\* str)**

Convert the hx509 name object into a printable string. The resulting string should be freed with free().

**Parameters:**

*name* name to print

*str* the string to return

**Returns:**

An hx509 error code, see **hx509\_get\_error\_string()**.

**int hx509\_parse\_name (hx509\_context context, const char \* str, hx509\_name \* name)**

Parse a string into a hx509 name object.

**Parameters:**

*context* A hx509 context.

*str* a string to parse.

*name* the resulting object, NULL in case of error.

**Returns:**

An hx509 error code, see **hx509\_get\_error\_string()**.

**int hx509\_unparse\_der\_name (const void \* data, size\_t length, char \*\* str)**

Convert a DER encoded name info a string.

**Parameters:**

*data* data to a DER/BER encoded name

*length* length of data

*str* the resulting string, is NULL on failure.

**Returns:**

An hx509 error code, see **hx509\_get\_error\_string()**.