NAME

EVP_SIGNATURE, EVP_SIGNATURE_fetch, EVP_SIGNATURE_free, EVP_SIGNATURE_up_ref, EVP_SIGNATURE_is_a, EVP_SIGNATURE_get0_provider, EVP_SIGNATURE_do_all_provided, EVP_SIGNATURE_names_do_all, EVP_SIGNATURE_get0_name, EVP_SIGNATURE_get0_description, EVP_SIGNATURE_gettable_ctx_params, EVP_SIGNATURE_settable_ctx_params - Functions to manage EVP_SIGNATURE algorithm objects

SYNOPSIS

```
#include <openssl/evp.h>

typedef struct evp_signature_st EVP_SIGNATURE;

EVP_SIGNATURE *EVP_SIGNATURE_fetch(OSSL_LIB_CTX *ctx, const char *algorithm,
                                   const char *properties);
void EVP_SIGNATURE_free(EVP_SIGNATURE *signature);
int EVP_SIGNATURE_up_ref(EVP_SIGNATURE *signature);
const char *EVP_SIGNATURE_get0_name(const EVP_SIGNATURE *signature);
int EVP_SIGNATURE_is_a(const EVP_SIGNATURE *signature, const char *name);
OSSL_PROVIDER *EVP_SIGNATURE_get0_provider(const EVP_SIGNATURE *signature);
void EVP_SIGNATURE_do_all_provided(OSSL_LIB_CTX *libctx,
                                   void (*fn)(EVP_SIGNATURE *signature,
                                              void *arg),
                                   void *arg);
int EVP_SIGNATURE_names_do_all(const EVP_SIGNATURE *signature,
                               void (*fn)(const char *name, void *data),
                               void *data);
const char *EVP_SIGNATURE_get0_name(const EVP_SIGNATURE *signature);
const char *EVP_SIGNATURE_get0_description(const EVP_SIGNATURE *signature);
const OSSL_PARAM *EVP_SIGNATURE_gettable_ctx_params(const EVP_SIGNATURE *sig);
const OSSL_PARAM *EVP_SIGNATURE_settable_ctx_params(const EVP_SIGNATURE *sig);
```

DESCRIPTION

EVP_SIGNATURE_fetch() fetches the implementation for the given **algorithm** from any provider offering it, within the criteria given by the **properties**. The algorithm will be one offering functions for performing signature related tasks such as signing and verifying. See "ALGORITHM FETCHING" in **crypto**(7) for further information.

The returned value must eventually be freed with **EVP_SIGNATURE_free()**.

EVP_SIGNATURE_free() decrements the reference count for the **EVP_SIGNATURE** structure. Typically this structure will have been obtained from an earlier call to **EVP_SIGNATURE_fetch()**. If the reference count drops to 0 then the structure is freed.

EVP_SIGNATURE_up_ref() increments the reference count for an **EVP_SIGNATURE** structure.

EVP_SIGNATURE_is_a() returns 1 if *signature* is an implementation of an algorithm that's identifiable with *name*, otherwise 0.

EVP_SIGNATURE_get0_provider() returns the provider that *signature* was fetched from.

EVP_SIGNATURE_do_all_provided() traverses all SIGNATURE implemented by all activated providers in the given library context *libctx*, and for each of the implementations, calls the given function *fn* with the implementation method and the given *arg* as argument.

EVP_SIGNATURE_get0_name() returns the algorithm name from the provided implementation for the given *signature*. Note that the *signature* may have multiple synonyms associated with it. In this case the first name from the algorithm definition is returned. Ownership of the returned string is retained by the *signature* object and should not be freed by the caller.

EVP_SIGNATURE_names_do_all() traverses all names for *signature*, and calls *fn* with each name and *data*.

EVP_SIGNATURE_get0_description() returns a description of the *signature*, meant for display and human consumption. The description is at the discretion of the *signature* implementation.

EVP_SIGNATURE_gettable_ctx_params() and **EVP_SIGNATURE_settable_ctx_params()** return a constant **OSSL_PARAM**(3) array that describes the names and types of key parameters that can be retrieved or set by a signature algorithm using **EVP_PKEY_CTX_get_params**(3) and **EVP_PKEY_CTX_set_params**(3).

RETURN VALUES

EVP_SIGNATURE_fetch() returns a pointer to an **EVP_SIGNATURE** for success or **NULL** for failure.

EVP_SIGNATURE_up_ref() returns 1 for success or 0 otherwise.

EVP_SIGNATURE_names_do_all() returns 1 if the callback was called for all names. A return value of 0 means that the callback was not called for any names.

EVP_SIGNATURE_gettable_ctx_params() and **EVP_SIGNATURE_settable_ctx_params()** return a constant **OSSL_PARAM**(3) array or NULL on error.

SEE ALSO

"ALGORITHM FETCHING" in **crypto**(7), **OSSL_PROVIDER**(3)

HISTORY

The functions described here were added in OpenSSL 3.0.

COPYRIGHT

Copyright 2019-2023 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use this file except in compliance with the License. You can obtain a copy in the file LICENSE in the source distribution or at <https://www.openssl.org/source/license.html>.