NAME

provider−digest − The digest library <−> provider functions

SYNOPSIS

```
#include <openssl/core_dispatch.h>
#include <openssl/core_names.h>

/*
 * Digests support the following function signatures in OSSL_DISPATCH arrays.
 * (The function signatures are not actual functions).
 */

/* Context management */
void *OSSL_FUNC_digest_newctx(void *provctx);
void OSSL_FUNC_digest_freectx(void *dctx);
void *OSSL_FUNC_digest_dupctx(void *dctx);

/* Digest generation */
int OSSL_FUNC_digest_init(void *dctx, const OSSL_PARAM params[]);
int OSSL_FUNC_digest_update(void *dctx, const unsigned char *in, size_t inl);
int OSSL_FUNC_digest_final(void *dctx, unsigned char *out, size_t *outl,
                           size_t outsz);
int OSSL_FUNC_digest_digest(void *provctx, const unsigned char *in, size_t inl,
                            unsigned char *out, size_t *outl, size_t outsz);

/* Digest parameter descriptors */
const OSSL_PARAM *OSSL_FUNC_digest_gettable_params(void *provctx);

/* Digest operation parameter descriptors */
const OSSL_PARAM *OSSL_FUNC_digest_gettable_ctx_params(void *dctx,
                                                       void *provctx);
const OSSL_PARAM *OSSL_FUNC_digest_settable_ctx_params(void *dctx,
                                                       void *provctx);

/* Digest parameters */
int OSSL_FUNC_digest_get_params(OSSL_PARAM params[]);

/* Digest operation parameters */
int OSSL_FUNC_digest_set_ctx_params(void *dctx, const OSSL_PARAM params[]);
int OSSL_FUNC_digest_get_ctx_params(void *dctx, OSSL_PARAM params[]);
```

DESCRIPTION

This documentation is primarily aimed at provider authors. See **provider** (7) for further information.

The DIGEST operation enables providers to implement digest algorithms and make them available to applications via the API functions **EVP_DigestInit_ex** (3), **EVP_DigestUpdate** (3) and **EVP_DigestFinal** (3) (and other related functions).

All ‘‘functions’’ mentioned here are passed as function pointers between *libcrypto* and the provider in **OSSL_DISPATCH** arrays via **OSSL_ALGORITHM** arrays that are returned by the provider’s **provider_query_operation()** function (see ‘‘Provider Functions’’ in **provider−base** (7)).

All these ‘‘functions’’ have a corresponding function type definition named **OSSL_FUNC_{name}_fn**, and a helper function to retrieve the function pointer from an **OSSL_DISPATCH** element named **OSSL_FUNC_{name}**. For example, the ‘‘function’’ **OSSL_FUNC_digest_newctx()** has these:

```
typedef void *(OSSL_OSSL_FUNC_digest_newctx_fn)(void *provctx);
static ossl_inline OSSL_OSSL_FUNC_digest_newctx_fn
    OSSL_FUNC_digest_newctx(const OSSL_DISPATCH *opf);
```

OSSL_DISPATCH arrays are indexed by numbers that are provided as macros in **openssl-core_dispatch.h** (7), as follows:

```
OSSL_FUNC_digest_newctx              OSSL_FUNC_DIGEST_NEWCTX
OSSL_FUNC_digest_freectx             OSSL_FUNC_DIGEST_FREECTX
OSSL_FUNC_digest_dupctx              OSSL_FUNC_DIGEST_DUPCTX

OSSL_FUNC_digest_init                OSSL_FUNC_DIGEST_INIT
OSSL_FUNC_digest_update              OSSL_FUNC_DIGEST_UPDATE
OSSL_FUNC_digest_final               OSSL_FUNC_DIGEST_FINAL
OSSL_FUNC_digest_digest              OSSL_FUNC_DIGEST_DIGEST

OSSL_FUNC_digest_get_params          OSSL_FUNC_DIGEST_GET_PARAMS
OSSL_FUNC_digest_get_ctx_params      OSSL_FUNC_DIGEST_GET_CTX_PARAMS
OSSL_FUNC_digest_set_ctx_params      OSSL_FUNC_DIGEST_SET_CTX_PARAMS

OSSL_FUNC_digest_gettable_params     OSSL_FUNC_DIGEST_GETTABLE_PARAMS
OSSL_FUNC_digest_gettable_ctx_params OSSL_FUNC_DIGEST_GETTABLE_CTX_PARAMS
OSSL_FUNC_digest_settable_ctx_params OSSL_FUNC_DIGEST_SETTABLE_CTX_PARAMS
```

A digest algorithm implementation may not implement all of these functions. In order to be usable all or none of OSSL_FUNC_digest_newctx, OSSL_FUNC_digest_freectx, OSSL_FUNC_digest_init, OSSL_FUNC_digest_update and OSSL_FUNC_digest_final should be implemented. All other functions are optional.

Context Management Functions

OSSL_FUNC_digest_newctx() should create and return a pointer to a provider side structure for holding context information during a digest operation. A pointer to this context will be passed back in a number of the other digest operation function calls. The parameter *provctx* is the provider context generated during provider initialisation (see **provider** (7)).

OSSL_FUNC_digest_freectx() is passed a pointer to the provider side digest context in the *dctx* parameter. This function should free any resources associated with that context.

OSSL_FUNC_digest_dupctx() should duplicate the provider side digest context in the *dctx* parameter and return the duplicate copy.

Digest Generation Functions

OSSL_FUNC_digest_init() initialises a digest operation given a newly created provider side digest context in the *dctx* parameter. The*par ams*, if not NULL, should be set on the context in a manner similar to using **OSSL_FUNC_digest_set_ctx_params()**.

OSSL_FUNC_digest_update() is called to supply data to be digested as part of a previously initialised digest operation. The *dctx* parameter contains a pointer to a previously initialised provider side context. **OSSL_FUNC_digest_update()** should digest *inl* bytes of data at the location pointed to by *in*. **OSSL_FUNC_digest_update()** may be called multiple times for a single digest operation.

OSSL_FUNC_digest_final() generates a digest started through previous **OSSL_FUNC_digest_init()** and **OSSL_FUNC_digest_update()** calls. The *dctx* parameter contains a pointer to the provider side context. The digest should be written to **out* and the length of the digest to **outl*. The digest should not exceed *outsz* bytes.

OSSL_FUNC_digest_digest() is a ''oneshot'' digest function. No provider side digest context is used. Instead the provider context that was created during provider initialisation is passed in the *provctx* parameter (see **provider** (7)). *inl* bytes at *in* should be digested and the result should be stored at *out*. The length of the digest should be stored in **outl* which should not exceed *outsz* bytes.

Digest Parameters

See **OSSL_PARAM** (3) for further details on the parameters structure used by these functions.

OSSL_FUNC_digest_get_params() gets details of the algorithm implementation and stores them in *params*.

OSSL_FUNC_digest_set_ctx_params() sets digest operation parameters for the provider side digest context *dctx* to *params*. Any parameter settings are additional to any that were previously set. Passing NULL for *params* should return true.

OSSL_FUNC_digest_get_ctx_params() gets digest operation details details from the given provider side digest context *dctx* and stores them in *params*. Passing NULL for *params* should return true.

OSSL_FUNC_digest_gettable_params() returns a constant **OSSL_PARAM** array containing descriptors of the parameters that **OSSL_FUNC_digest_get_params()** can handle.

OSSL_FUNC_digest_gettable_ctx_params() and **OSSL_FUNC_digest_settable_ctx_params()** both return constant **OSSL_PARAM** arrays as descriptors of the parameters that **OSSL_FUNC_digest_get_ctx_params()** and **OSSL_FUNC_digest_set_ctx_params()** can handle, respectively. The array is based on the current state of the provider side context if *dctx* is not NULL and on the provider side algorithm *provctx* otherwise.

Parameters currently recognised by built-in digests with this function are as follows. Not all parameters are relevant to, or are understood by all digests:

‘‘blocksize’’ (**OSSL_DIGEST_PARAM_BLOCK_SIZE**) <unsigned integer>
: The digest block size. The length of the ‘‘blocksize’’ parameter should not exceed that of a **size_t**.

‘‘size’’ (**OSSL_DIGEST_PARAM_SIZE**) <unsigned integer>
: The digest output size. The length of the ‘‘size’’ parameter should not exceed that of a **size_t**.

‘‘flags’’ (**OSSL_DIGEST_PARAM_FLAGS**) <unsigned integer>
: Diverse flags that describe exceptional behaviour for the digest:

  EVP_MD_FLAG_ONESHOT
  : This digest method can only handle one block of input.

  EVP_MD_FLAG_XOF
  : This digest method is an extensible-output function (XOF) and supports setting the **OSSL_DIGEST_PARAM_XOFLEN** parameter.

  EVP_MD_FLAG_DIGALGID_NULL
  : When setting up a DigestAlgorithmIdentifier, this flag will have the parameter set to NULL by default. Use this for PKCS#1. *Note: if combined with EVP_MD_FLAG_DIGALGID_ABSENT, the latter will override.*

  EVP_MD_FLAG_DIGALGID_ABSENT
  : When setting up a DigestAlgorithmIdentifier, this flag will have the parameter be left absent by default. *Note: if combined with EVP_MD_FLAG_DIGALGID_NULL, the latter will be overridden.*

  EVP_MD_FLAG_DIGALGID_CUSTOM
  : Custom DigestAlgorithmIdentifier handling via ctrl, with **EVP_MD_FLAG_DIGALGID_ABSENT** as default. *Note: if combined with EVP_MD_FLAG_DIGALGID_NULL, the latter will be overridden.* Currently unused.

  The length of the ‘‘flags’’ parameter should equal that of an **unsigned long int**.

Digest Context Parameters

OSSL_FUNC_digest_set_ctx_params() sets digest parameters associated with the given provider side digest context *dctx* to *params*. Any parameter settings are additional to any that were previously set. See **OSSL_PARAM** (3) for further details on the parameters structure.

OSSL_FUNC_digest_get_ctx_params() gets details of currently set parameters values associated with the give provider side digest context *dctx* and stores them in *params*. See **OSSL_PARAM** (3) for further details on the parameters structure.

RETURN VALUES

OSSL_FUNC_digest_newctx() and **OSSL_FUNC_digest_dupctx()** should return the newly created provider side digest context, or NULL on failure.

OSSL_FUNC_digest_init(), **OSSL_FUNC_digest_update()**, **OSSL_FUNC_digest_final()**, **OSSL_FUNC_digest_digest()**, **OSSL_FUNC_digest_set_params()** and **OSSL_FUNC_digest_get_params()** should return 1 for success or 0 on error.

OSSL_FUNC_digest_size() should return the digest size.

OSSL_FUNC_digest_block_size() should return the block size of the underlying digest algorithm.

BUGS

The **EVP_Q_digest()**, **EVP_Digest()** and **EVP_DigestFinal_ex()** API calls do not expect the digest size to be larger than EVP_MAX_MD_SIZE. Any algorithm which produces larger digests is unusable with those API calls.

SEE ALSO

provider (7), **OSSL_PROVIDER-FIPS** (7), **OSSL_PROVIDER-default** (7), **OSSL_PROVIDER-legacy** (7), **EVP_MD-common** (7), **EVP_MD-BLAKE2** (7), **EVP_MD-MD2** (7), **EVP_MD-MD4** (7), **EVP_MD-MD5** (7), **EVP_MD-MD5-SHA1** (7), **EVP_MD-MDC2** (7), **EVP_MD-RIPEMD160** (7), **EVP_MD-SHA1** (7), **EVP_MD-SHA2** (7), **EVP_MD-SHA3** (7), **EVP_MD-SHAKE** (7), **EVP_MD-SM3** (7), **EVP_MD-WHIRLPOOL** (7), **life_cycle-digest** (7), **EVP_DigestInit** (3)

HISTORY

The provider DIGEST interface was introduced in OpenSSL 3.0.

COPYRIGHT