NAME

OSSL_PROVIDER-default - OpenSSL default provider

DESCRIPTION

The OpenSSL default provider supplies the majority of OpenSSL's diverse algorithm implementations. If an application doesn't specify anything else explicitly (e.g. in the application or via config), then this is the provider that will be used as fallback: It is loaded automatically the first time that an algorithm is fetched from a provider or a function acting on providers is called and no other provider has been loaded yet.

If an attempt to load a provider has already been made (whether successful or not) then the default provider won't be loaded automatically. Therefore if the default provider is to be used in conjunction with other providers then it must be loaded explicitly. Automatic loading of the default provider only occurs a maximum of once; if the default provider is explicitly unloaded then the default provider will not be automatically loaded again.

Properties

The implementations in this provider specifically have this property defined:

"provider=default"

It may be used in a property query string with fetching functions such as **EVP_MD_fetch**(3) or **EVP_CIPHER_fetch**(3), as well as with other functions that take a property query string, such as **EVP_PKEY_CTX_new_from_name**(3).

It isn't mandatory to query for this property, except to make sure to get implementations of this provider and none other.

Some implementations may define additional properties. Exact information is listed below

OPERATIONS AND ALGORITHMS

The OpenSSL default provider supports these operations and algorithms:

Hashing Algorithms / Message Digests

SHA1, see **EVP_MD-SHA1**(7)
SHA2, see **EVP_MD-SHA2**(7)
SHA3, see **EVP_MD-SHA3**(7)
KECCAK-KMAC, see **EVP_MD-KECCAK-KMAC**(7)
SHAKE, see **EVP_MD-SHAKE**(7)
BLAKE2, see **EVP_MD-BLAKE2**(7)

SM3, see **EVP_MD-SM3**(7)
MD5, see **EVP_MD-MD5**(7)
MD5-SHA1, see **EVP_MD-MD5-SHA1**(7)
RIPEMD160, see **EVP_MD-RIPEMD160**(7)
NULL, see **EVP_MD-NULL**(7)

Symmetric Ciphers

AES, see **EVP_CIPHER-AES**(7)
ARIA, see **EVP_CIPHER-ARIA**(7)
CAMELLIA, see **EVP_CIPHER-CAMELLIA**(7)
3DES, see **EVP_CIPHER-DES**(7)
SEED, see **EVP_CIPHER-SEED**(7)
SM4, see **EVP_CIPHER-SM4**(7)
ChaCha20, see **EVP_CIPHER-CHACHA**(7)
ChaCha20-Poly1305, see **EVP_CIPHER-CHACHA**(7)
NULL, see **EVP_CIPHER-NULL**(7)

Message Authentication Code (MAC)

BLAKE2, see **EVP_MAC-BLAKE2**(7)
CMAC, see **EVP_MAC-CMAC**(7)
GMAC, see **EVP_MAC-GMAC**(7)
HMAC, see **EVP_MAC-HMAC**(7)
KMAC, see **EVP_MAC-KMAC**(7)
SIPHASH, see **EVP_MAC-Siphash**(7)
POLY1305, see **EVP_MAC-Poly1305**(7)

Key Derivation Function (KDF)

HKDF, see **EVP_KDF-HKDF**(7)
SSKDF, see **EVP_KDF-SS**(7)
PBKDF2, see **EVP_KDF-PBKDF2**(7)
PKCS12KDF, see **EVP_KDF-PKCS12KDF**(7)
SSHKDF, see **EVP_KDF-SSHKDF**(7)
TLS1-PRF, see **EVP_KDF-TLS1_PRF**(7)
KBKDF, see **EVP_KDF-KB**(7)
X942KDF-ASN1, see **EVP_KDF-X942-ASN1**(7)
X942KDF-CONCAT, see **EVP_KDF-X942-CONCAT**(7)
X963KDF, see **EVP_KDF-X963**(7)
SCRYPT, see **EVP_KDF-SCRYPT**(7)
KRB5KDF, see **EVP_KDF-KRB5KDF**(7)

Key Exchange

DH, see **EVP_KEYEXCH-DH**(7)
ECDH, see **EVP_KEYEXCH-ECDH**(7)
X25519, see **EVP_KEYEXCH-X25519**(7)
X448, see **EVP_KEYEXCH-X448**(7)

Asymmetric Signature

DSA, see **EVP_SIGNATURE-DSA**(7)
RSA, see **EVP_SIGNATURE-RSA**(7)
HMAC, see **EVP_SIGNATURE-HMAC**(7)
SIPHASH, see **EVP_SIGNATURE-Siphash**(7)
POLY1305, see **EVP_SIGNATURE-Poly1305**(7)
CMAC, see **EVP_SIGNATURE-CMAC**(7)

Asymmetric Cipher

RSA, see **EVP_ASYM_CIPHER-RSA**(7)
SM2, see **EVP_ASYM_CIPHER-SM2**(7)

Asymmetric Key Encapsulation

RSA, see **EVP_KEM-RSA**(7)

Asymmetric Key Management

DH, see **EVP_KEYMGMT-DH**(7)
DHX, see **EVP_KEYMGMT-DHX**(7)
DSA, see **EVP_KEYMGMT-DSA**(7)
RSA, see **EVP_KEYMGMT-RSA**(7)
EC, see **EVP_KEYMGMT-EC**(7)
X25519, see **EVP_KEYMGMT-X25519**(7)
X448, see **EVP_KEYMGMT-X448**(7)

Random Number Generation

CTR-DRBG, see **EVP_RAND-CTR-DRBG**(7)
HASH-DRBG, see **EVP_RAND-HASH-DRBG**(7)
HMAC-DRBG, see **EVP_RAND-HMAC-DRBG**(7)
SEED-SRC, see **EVP_RAND-SEED-SRC**(7)
TEST-RAND, see **EVP_RAND-TEST-RAND**(7)

Asymmetric Key Encoder

The default provider also includes all of the encoding algorithms present in the base provider. Some of these have the property "fips=yes", to allow them to be used together with the FIPS provider.

RSA, see **OSSL_ENCODER-RSA**(7)
DH, see **OSSL_ENCODER-DH**(7)
DSA, see **OSSL_ENCODER-DSA**(7)
EC, see **OSSL_ENCODER-EC**(7)
X25519, see **OSSL_ENCODER-X25519**(7)
X448, see **OSSL_ENCODER-X448**(7)

SEE ALSO

openssl-core.h(7), **openssl-core_dispatch.h**(7), **provider**(7), **OSSL_PROVIDER-base**(7)

HISTORY

The RIPEMD160 digest was added to the default provider in OpenSSL 3.0.7.

All other functionality was added in OpenSSL 3.0.

COPYRIGHT