NAME

EVP_DigestVerifyInit, EVP_DigestVerifyUpdate, EVP_DigestVerifyFinal - EVP signature verification functions

SYNOPSIS

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

int EVP_DigestVerifyInit(EVP_MD_CTX *ctx, EVP_PKEY_CTX **pctx,
const EVP_MD *type, ENGINE *e, EVP_PKEY *pkey);
int EVP_DigestVerifyUpdate(EVP_MD_CTX *ctx, const void *d, unsigned int cnt);
int EVP_DigestVerifyFinal(EVP_MD_CTX *ctx, const unsigned char *sig, size_t siglen);
```

DESCRIPTION

The EVP signature routines are a high level interface to digital signatures.

EVP_DigestVerifyInit() sets up verification context **ctx** to use digest **type** from ENGINE **impl** and public key **pkey**. **ctx** must be initialized with *EVP_MD_CTX_init()* before calling this function. If **pctx** is not NULL the EVP_PKEY_CTX of the verification operation will be written to ***pctx**: this can be used to set alternative verification options.

EVP_DigestVerifyUpdate() hashes **cnt** bytes of data at **d** into the verification context **ctx**. This function can be called several times on the same **ctx** to include additional data. This function is currently implemented using a macro.

EVP_DigestVerifyFinal() verifies the data in **ctx** against the signature in **sig** of length **siglen**.

RETURN VALUES

EVP_DigestVerifyInit() and *EVP_DigestVerifyUpdate()* return 1 for success and 0 or a negative value for failure. In particular a return value of -2 indicates the operation is not supported by the public key algorithm.

Unlike other functions the return value 0 from *EVP_DigestVerifyFinal()* only indicates that the signature did not verify successfully (that is tbs did not match the original data or the signature was of invalid form) it is not an indication of a more serious error.

The error codes can be obtained from *ERR_get_error(3)*.

NOTES

The **EVP** interface to digital signatures should almost always be used in preference to the low level interfaces. This is because the code then becomes transparent to the algorithm used and much more flexible.

In previous versions of OpenSSL there was a link between message digest types and public key algorithms. This meant that "clone" digests such as *EVP_dss1()* needed to be used to sign using SHA1 and DSA. This is no longer necessary and the use of clone digest is now discouraged.

For some key types and parameters the random number generator must be seeded or the operation will fail.

The call to *EVP_DigestVerifyFinal()* internally finalizes a copy of the digest context. This means that *EVP_VerifyUpdate()* and *EVP_VerifyFinal()* can be called later to digest and verify additional data.

Since only a copy of the digest context is ever finalized the context must be cleaned up after use by calling *EVP_MD_CTX_cleanup()* or a memory leak will occur.

SEE ALSO

EVP_DigestSignInit(3), *EVP_DigestInit(3)*, *err(3)*, *evp(3)*, *hmac(3)*, *md2(3)*, *md5(3)*, *mdc2(3)*, *ripemd(3)*, *sha(3)*, *dgst(1)*

HISTORY

EVP_DigestVerifyInit(), *EVP_DigestVerifyUpdate()* and *EVP_DigestVerifyFinal()* were first added to OpenSSL 1.0.0.