NAME

CTLOG_new_ex, CTLOG_new, CTLOG_new_from_base64, CTLOG_new_from_base64_ex, CTLOG_free, CTLOG_get0_name, CTLOG_get0_log_id, CTLOG_get0_public_key - encapsulates information about a Certificate Transparency log

SYNOPSIS

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

CTLOG *CTLOG_new_ex(EVP_PKEY *public_key, const char *name,
                    OSSL_LIB_CTX *libctx, const char *propq);
CTLOG *CTLOG_new(EVP_PKEY *public_key, const char *name);

int CTLOG_new_from_base64_ex(CTLOG **ct_log, const char *pkey_base64,
                             const char *name, OSSL_LIB_CTX *libctx,
                             const char *propq);
int CTLOG_new_from_base64(CTLOG ** ct_log,
                          const char *pkey_base64, const char *name);
void CTLOG_free(CTLOG *log);
const char *CTLOG_get0_name(const CTLOG *log);
void CTLOG_get0_log_id(const CTLOG *log, const uint8_t **log_id,
                       size_t *log_id_len);
EVP_PKEY *CTLOG_get0_public_key(const CTLOG *log);
```

DESCRIPTION

CTLOG_new_ex() returns a new CTLOG that represents the Certificate Transparency (CT) log with the given public key and associates it with the library context *libctx* and property query string *propq*. A name must also be provided that can be used to help users identify this log. Ownership of the public key is transferred.

CTLOG_new() does the same thing as **CTLOG_new_ex()** but with the default library context and the default property query string.

CTLOG_new_from_base64_ex() also creates a new CTLOG, but takes the public key in base64-encoded DER form and sets the ct_log pointer to point to the new CTLOG. The base64 will be decoded and the public key parsed. The CTLOG will be associated with the given library context *libctx* and property query string *propq*.

CTLOG_new_from_base64() does the same thing as **CTLOG_new_from_base64_ex()** except that the default library context and property query string are used.

Regardless of whether **CTLOG_new()** or **CTLOG_new_from_base64()** is used, it is the caller's responsibility to pass the CTLOG to **CTLOG_free()** once it is no longer needed. This will delete it and, if created by **CTLOG_new()**, the EVP_PKEY that was passed to it.

CTLOG_get0_name() returns the name of the log, as provided when the CTLOG was created. Ownership of the string remains with the CTLOG.

CTLOG_get0_log_id() sets *log_id to point to a string containing that log's LogID (see RFC 6962). It sets *log_id_len to the length of that LogID. For a v1 CT log, the LogID will be a SHA-256 hash (i.e. 32 bytes long). Ownership of the string remains with the CTLOG.

CTLOG_get0_public_key() returns the public key of the CT log. Ownership of the EVP_PKEY remains with the CTLOG.

RETURN VALUES

CTLOG_new() will return NULL if an error occurs.

CTLOG_new_from_base64() will return 1 on success, 0 otherwise.

SEE ALSO

ct(7)

HISTORY

The functions **CTLOG_new_ex()** and **CTLOG_new_from_base64_ex()** were added in OpenSSL 3.0. All other functions were added in OpenSSL 1.1.0.

COPYRIGHT