

**NAME**

CTLOG\_STORE\_new\_ex, CTLOG\_STORE\_new, CTLOG\_STORE\_free,  
 CTLOG\_STORE\_load\_default\_file, CTLOG\_STORE\_load\_file - Create and populate a Certificate  
 Transparency log list

**SYNOPSIS**

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

```
CTLOG_STORE *CTLOG_STORE_new_ex(OSSL_LIB_CTX *libctx, const char *propq);
```

```
CTLOG_STORE *CTLOG_STORE_new(void);
```

```
void CTLOG_STORE_free(CTLOG_STORE *store);
```

```
int CTLOG_STORE_load_default_file(CTLOG_STORE *store);
```

```
int CTLOG_STORE_load_file(CTLOG_STORE *store, const char *file);
```

**DESCRIPTION**

A CTLOG\_STORE is a container for a list of CTLOGs (Certificate Transparency logs). The list can be loaded from one or more files and then searched by LogID (see RFC 6962, Section 3.2, for the definition of a LogID).

**CTLOG\_STORE\_new\_ex()** creates an empty list of CT logs associated with the library context *libctx* and the property query string *propq*.

**CTLOG\_STORE\_new()** does the same thing as **CTLOG\_STORE\_new\_ex()** but with the default library context and property query string.

The CTLOG\_STORE is then populated by **CTLOG\_STORE\_load\_default\_file()** or **CTLOG\_STORE\_load\_file()**. **CTLOG\_STORE\_load\_default\_file()** loads from the default file, which is named *ct\_log\_list.cnf* in OPENSSLDIR (see the output of **openssl-version(1)**). This can be overridden using an environment variable named **CTLOG\_FILE**. **CTLOG\_STORE\_load\_file()** loads from a caller-specified file path instead. Both of these functions append any loaded CT logs to the CTLOG\_STORE.

The expected format of the file is:

```
enabled_logs=foo,bar
```

```
[foo]
```

```
description = Log 1
```

```
key = <base64-encoded DER SubjectPublicKeyInfo here>
```

```
[bar]
description = Log 2
key = <base64-encoded DER SubjectPublicKeyInfo here>
```

Once a CTLOG\_STORE is no longer required, it should be passed to **CTLOG\_STORE\_free()**. This will delete all of the CTLOGs stored within, along with the CTLOG\_STORE itself.

## NOTES

If there are any invalid CT logs in a file, they are skipped and the remaining valid logs will still be added to the CTLOG\_STORE. A CT log will be considered invalid if it is missing a "key" or "description" field.

## RETURN VALUES

Both **CTLOG\_STORE\_load\_default\_file** and **CTLOG\_STORE\_load\_file** return 1 if all CT logs in the file are successfully parsed and loaded, 0 otherwise.

## SEE ALSO

**ct(7)**, **CTLOG\_STORE\_get0\_log\_by\_id(3)**, **SSL\_CTX\_set\_ctlog\_list\_file(3)**

## HISTORY

CTLOG\_STORE\_new\_ex was added in OpenSSL 3.0. All other functions were added in OpenSSL 1.1.0.

## COPYRIGHT

Copyright 2016-2020 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>.