

TEC return codes

These return codes are returned by the various libtec functions.

[More...](#)

Macros

```
#define CTS\_OK 0
#define CTS\_NO\_CHIP 1
#define CTS\_IN\_PROGRESS 2
#define CTS\_TIMEOUT 3
#define CTS\_PARAM 4
#define CTS\_NOT\_STARTED 5
#define CTS\_STOPPED 6
#define CTS\_CTLS\_INIT 7
#define CTS\_ERROR 8
#define CTS\_CTLS\_NOT\_ALLOWED 9
#define CTS\_CTLS\_EMV\_NO\_CARD 10
#define CTS\_ERR\_LOAD 11
#define CTS\_VAS\_DECRYPTION\_NOT\_REQUIRED 12
#define CTS\_VAS\_DATA\_DECRYPTED\_OK 13
#define CTS\_VAS\_DATA\_DECRYPTED\_FAILED 14
#define CTS\_UX\_MSRDATA\_NOT\_AVAILABLE\_TIMEOUT 15
#define CTS\_NFC\_INIT 16
#define CTS\_CHIP\_WITH\_WRONG\_ATR 17
#define CTS\_API\_NOT\_ALLOWED 18
#define CTS\_VAS\_DATA\_ENCRYPTED\_OK 19
#define CTS\_UX\_NO\_MSR\_DATA\_AFTER\_CARD\_REMOVL 20
#define CTS\_MSR\_ONLY\_REQUEST\_BUT\_NO\_DEVICE 21
#define CTS\_FORMAT -101
#define CTS\_TLV -102
```

```
#define CTS\_PARAMETER -103
#define CTS\_UNKNOWN\_CLA -104
#define CTS\_UNKNOWN\_INS -105
#define CTS\_BLOCKED -106
#define CTS\_NO\_SERVER -201
#define CTS\_IPC -202
```

Detailed Description

These return codes are returned by the various libtec functions.

Macro Definition Documentation

? [CTS_API_NOT_ALLOWED](#)

```
#define CTS_API_NOT_ALLOWED 18
```

API not allowed because TEC-callback is still in progress.

? [CTS_BLOCKED](#)

```
#define CTS_BLOCKED -106
```

TEC service is blocked by another client/function call.

? [CTS_CHIP_WITH_WRONG_ATR](#)

```
#define CTS_CHIP_WITH_WRONG_ATR 17
```

Chip-Card is inserted but its ATR is not EMV conform. This value may be returned by [cts_WaitSelection\(\)](#) only if `EMV_CT_DETECT_WRONG_ATR` is set in the input parameter `options[4]` of the function [cts_StartSelection\(\)](#).

? [CTS_CTLs_EMV_NO_CARD](#)

```
#define CTS_CTLs_EMV_NO_CARD 10
```

ADK-EMV has detected no medium to perform a contactless payment (returned by [cts_WaitSelection\(\)](#)).

? CTS_CTLS_INIT

```
#define CTS_CTLS_INIT 7
```

Contactless transaction was not set up correctly (returned by [cts_WaitSelection\(\)](#)).

? CTS_CTLS_NOT_ALLOWED

```
#define CTS_CTLS_NOT_ALLOWED 9
```

VFI-Reader has not yet finished previous transaction (returned by [cts_WaitSelection\(\)](#)).

? CTS_ERR_LOAD

```
#define CTS_ERR_LOAD 11
```

Error loading dynamic library (NFC_Client.so/NFC_Client.vsl).

? CTS_ERROR

```
#define CTS_ERROR 8
```

Internal error.

? CTS_FORMAT

```
#define CTS_FORMAT -101
```

Malformed message.

? CTS_IN_PROGRESS

```
#define CTS_IN_PROGRESS 2
```

Operation in progress (returned by [cts_StartSelection\(\)](#), [cts_WaitSelection\(\)](#), [cts_WaitCardRemoval\(\)](#)).

? CTS_IPC

```
#define CTS_IPC -202
```

IPC error (e.g. connection to server lost).

? CTS_MSR_ONLY_REQUEST_BUT_NO_DEVICE

```
#define CTS_MSR_ONLY_REQUEST_BUT_NO_DEVICE 21
```

"MSR only" requested but the device does not have a magstripe reader ([cts_StartSelection\(\)](#)).

? CTS_NFC_INIT

```
#define CTS_NFC_INIT 16
```

NFC not initialized (returned by [cts_WaitSelection\(\)](#)). Application has to call NFC_Client_Init before.

? CTS_NO_CHIP

```
#define CTS_NO_CHIP 1
```

Card without chip or with broken chip is inserted or card is inserted and [CTS_NO_POWERON](#) is set (returned by [cts_WaitSelection\(\)](#)).

? CTS_NO_SERVER

```
#define CTS_NO_SERVER -201
```

No server configured.

? CTS_NOT_STARTED

```
#define CTS_NOT_STARTED 5
```

Technology selection is not running (returned by [cts_WaitSelection\(\)](#), [cts_StopSelection\(\)](#), [cts_RemoveTechnologies\(\)](#)).

? CTS_OK

```
#define CTS_OK 0
```

No error, function executed successfully.

? CTS_PARAM

```
#define CTS_PARAM 4
```

Parameter error (returned by [cts_SetOptions\(\)](#), [cts_WaitSelection\(\)](#), [cts_WaitCardRemoval\(\)](#), [cts_AddTechnologies\(\)](#)).

? CTS_PARAMETER

```
#define CTS_PARAMETER -103
```

Missing parameter.

? CTS_STOPPED

```
#define CTS_STOPPED 6
```

Technology selection was aborted by [cts_StopSelection\(\)](#) or [cts_RemoveTechnologies\(\)](#) (returned by [cts_WaitSelection\(\)](#)).

? CTS_TIMEOUT

```
#define CTS_TIMEOUT 3
```

Timeout occurred (returned by [cts_WaitSelection\(\)](#), [cts_WaitCardRemoval2\(\)](#)).

? CTS_TLV

```
#define CTS_TLV -102
```

Error parsing TLV input buffer.

? CTS_UNKNOWN_CLA

```
#define CTS_UNKNOWN_CLA -104
```

Unknown class.

? CTS_UNKNOWN_INS

```
#define CTS_UNKNOWN_INS -105
```

Unknown instruction.

? CTS_UX_MSRDATA_NOT_AVAILABLE_TIMEOUT

```
#define CTS_UX_MSRDATA_NOT_AVAILABLE_TIMEOUT 15
```

Technology selection has been terminated because MSR-Data was not read during the time window set in the input parameter options[8..9] of the function [cts_StartSelection\(\)](#). This return code, which is returned by [cts_WaitSelection\(\)](#), is relevant and possible for UX-Devices only. It can be interpreted by the application as "card inserted incorrectly" or "magstripe reader malfunction".

? CTS_UX_NO_MSR_DATA_AFTER_CARD_REMOVL

```
#define CTS_UX_NO_MSR_DATA_AFTER_CARD_REMOVL 20
```

Ux30x only, card was removed but no MSR data could be read.

? CTS_VAS_DATA_DECRYPTED_FAILED

```
#define CTS_VAS_DATA_DECRYPTED_FAILED 14
```

VAS output data decryption failed (returned by TEC if `NFC_VAS_Data_Decrypt() == VAS_FAIL`).

TLV tag [CTS_DATA_TAG_VAS_DECRYPT_DATA_RESULT](#)

? CTS_VAS_DATA_DECRYPTED_OK

```
#define CTS_VAS_DATA_DECRYPTED_OK 13
```

VAS output data decrypted successfully (returned by TEC if `NFC_VAS_Data_Decrypt() == VAS_OK`).

TLV tag [CTS_DATA_TAG_VAS_DECRYPT_DATA_RESULT](#)

? CTS_VAS_DATA_ENCRYPTED_OK

```
#define CTS_VAS_DATA_ENCRYPTED_OK 19
```

Encrypted VAS output data was NOT decrypted. Reason: [CTS_VAS_DONT_DECRYPT](#) was put in by application.

TLV tag [CTS_DATA_TAG_VAS_DECRYPT_DATA_RESULT](#)

? CTS_VAS_DECRYPTION_NOT_REQUIRED

```
#define CTS_VAS_DECRYPTION_NOT_REQUIRED 12
```

VAS output data is not encrypted (returned by TEC if [NFC_VAS_Activate\(\)](#) reports data-not-encrypted).

TLV tag [CTS_DATA_TAG_VAS_DECRYPT_DATA_RESULT](#)