

TEEP-DEVICE

National Institute of Advanced Industrial Science and Technology

2021-02-19

1 Overview of TEEP-DEVICE	2
1.1 Features	2
1.2 Keys used on teep-device	2
1.3 Diagram	2
2 TEEP-DEVICE Operations	2
2.1 Step1	2
2.2 Step2	2
3 Clone and Building	2
3.1 Install Doxygen-1.9.2	2
3.1.1 Install Required Packages	2
3.1.2 Build and Install	3
3.2 Tamproto Setup	3
3.3 Keystone	3
3.3.1 Clone and Build	3
3.3.2 Check teep-device by running hello-app and teep-broker-app	3
3.3.3 Run Tamproto (TAM Server)	3
3.3.4 Copy the hello-app and teep-broker-app binaries to Unleashed	4
3.3.5 Check hello-app and teep-broker-app on Unleashed	4
3.4 OPTEE	6
3.4.1 Clone and Build	6
3.4.2 Check teep-device by running hello-app and teep-broker-app on RPI3	6
3.4.3 Run Tamproto (TAM Server)	6
3.4.4 Copy the hello-app and teep-broker-app binaries to RPI3	6
3.4.5 Check hello-app and teep-broker-app on RPI3	7
3.5 SGX	7
3.5.1 Clone and Build on SGX	7
3.5.2 Check teep-device by running hello-app & teep-broker-app on SGX	7
3.5.3 Run Tamproto (TAM Server)	8
3.5.4 Copy hello-app & teep-broker-app binaries to SGX	8
3.5.5 Check hello-app and teep-broker-app on SGX	8
4 Class Index	8
4.1 Class List	8
5 File Index	9
5.1 File List	9
6 Class Documentation	10
6.1 broker_ctx Struct Reference	10
6.1.1 Member Data Documentation	10
6.2 Command Struct Reference	10
6.2.1 Member Data Documentation	10

6.3 CommandQueue Class Reference	11
6.3.1 Member Function Documentation	11
6.4 lao_rpc_io Struct Reference	12
6.4.1 Member Data Documentation	12
6.5 libteep_async Struct Reference	13
6.5.1 Member Data Documentation	13
6.6 out_fct_wrap_type Struct Reference	14
6.6.1 Member Data Documentation	14
6.7 ta_manifest Struct Reference	14
6.7.1 Member Data Documentation	14
6.8 teep_agent_session Struct Reference	15
6.8.1 Member Data Documentation	15
6.9 teep_buffer_array Struct Reference	16
6.9.1 Member Data Documentation	16
6.10 teep_message Struct Reference	17
6.10.1 Member Data Documentation	18
6.11 teep_message_encoder Struct Reference	21
6.11.1 Member Data Documentation	21
6.12 teep_tc_info Struct Reference	22
6.12.1 Member Data Documentation	22
6.13 teep_tc_info_array Struct Reference	23
6.13.1 Member Data Documentation	23
6.14 teep_uint32_array Struct Reference	24
6.14.1 Member Data Documentation	24
6.15 teep_uint32_option Struct Reference	24
6.15.1 Member Data Documentation	24
7 File Documentation	25
7.1 teep-device/docs/cloning_and_building.md File Reference	25
7.2 teep-device/docs/overview_of_teeep-device.md File Reference	25
7.3 teep-device/docs/teeep-device_operations.md File Reference	25
7.4 teep-device/hello-app/keystone-main.cpp File Reference	25
7.4.1 Function Documentation	26
7.5 teep-device/hello-app/optee-main.c File Reference	30
7.5.1 Function Documentation	31
7.5.2 Variable Documentation	32
7.6 teep-device/hello-ta/hello-ta.c File Reference	32
7.6.1 Macro Definition Documentation	33
7.6.2 Function Documentation	33
7.7 teep-device/hello-ta/user_ta_header_defines.h File Reference	35
7.7.1 Macro Definition Documentation	35
7.8 teep-device/teeep-agent-ta/user_ta_header_defines.h File Reference	36

7.8.1 Macro Definition Documentation	36
7.9 teep-device/libteep/lib/libteep.c File Reference	37
7.9.1 Function Documentation	37
7.10 teep-device/libteep/lib/libteep.h File Reference	40
7.10.1 Enumeration Type Documentation	41
7.10.2 Function Documentation	43
7.11 teep-device/teep-agent-ta/sys/time.h File Reference	44
7.12 teep-device/teep-agent-ta/ta-store.c File Reference	45
7.12.1 Macro Definition Documentation	45
7.12.2 Function Documentation	45
7.12.3 Variable Documentation	47
7.13 teep-device/teep-agent-ta/ta-store.h File Reference	48
7.13.1 Function Documentation	48
7.14 teep-device/teep-agent-ta/teep-agent-ta.c File Reference	49
7.14.1 Enumeration Type Documentation	50
7.14.2 Function Documentation	51
7.15 teep-device/teep-agent-ta/teep-agent-ta.h File Reference	55
7.16 teep-device/teep-agent-ta/tools.c File Reference	55
7.16.1 Function Documentation	56
7.17 teep-device/teep-agent-ta/vsnprintf.c File Reference	58
7.17.1 Macro Definition Documentation	59
7.17.2 Typedef Documentation	61
7.17.3 Function Documentation	61
7.18 teep-device/teep-broker-app/http-lws.c File Reference	69
7.18.1 Typedef Documentation	70
7.18.2 Enumeration Type Documentation	70
7.18.3 Function Documentation	71
7.18.4 Variable Documentation	71
7.19 teep-device/teep-broker-app/http.h File Reference	72
7.19.1 Function Documentation	72
7.20 teep-device/teep-broker-app/teec-keystone.cpp File Reference	73
7.20.1 Function Documentation	74
7.20.2 Variable Documentation	75
7.21 teep-device/teep-broker-app/teec-pc.c File Reference	76
7.21.1 Function Documentation	76
7.21.2 Variable Documentation	78
7.22 teep-device/teep-broker-app/teep-broker.c File Reference	78
7.22.1 Function Documentation	79
7.22.2 Variable Documentation	80

Index	83
--------------	-----------

1 Overview of TEEP-DEVICE

1.1 Features

- AIST will prepare

1.2 Keys used on teep-device

- AIST will prepare

1.3 Diagram

- AIST will prepare

2 TEEP-DEVICE Operations

2.1 Step1

- TO DO by Arun after getting refence from Tsukamoto-san

2.2 Step2

- TO DO by Arun after getting refence from Tsukamoto-san

3 Clone and Building

Clone the teep-device source code and build it for Keystone, OPTEE and SGX. To build please refer to ta-ref.pdf->preparation section

- <https://192.168.100.100/rinkai/ta-ref/-/blob/teep-device-tb-slim/docs/ta-ref.pdf>

3.1 Install Doxygen-1.9.2

This PDF was generated using Doxygen version 1.9.2. To install doxygen-1.9.2 following procedure is necessary.

3.1.1 Install Required Packages

Install following packages on Ubuntu 18.04

```
sudo apt install doxygen-latex graphviz texlive-full texlive-latex-base latex-cjk-all
```

Above packages required to generate PDF using doxygen.

3.1.2 Build and Install

```
git clone https://github.com/doxygen/doxygen.git
cd doxygen
mkdir build
cd build
cmake -G "Unix Makefiles" ..
make
sudo make install
```

3.2 Tamproto Setup

To test teep-device, have to run TAM server on the PC.

Prerequisites

```
sudo apt install rustc npm
sudo pip3 install --upgrade git+https://github.com/ARMmbed/suit-manifest-generator.git@v0.0.2
```

Build and Install

```
git clone https://github.com/ko-isobe/tamproto.git
cd tamproto
git checkout cef99c07b669a49c2748b0c0ff0412ec1628b686 -b 2020-12-18
npm install
```

Make sure your PC is configured with IP address for network connectivity with TEEP device for further testing.

3.3 Keystone

Build teep-device with Keystone. Make sure Keystone and its supporting sources have been built already.

3.3.1 Clone and Build

Prepare the environment setup

```
export TEE=keystone
export KEYSTONE_DIR=<path to keystone dir>
export PATH=$PATH:$KEYSTONE_DIR/riscv/bin
export KEYEDGE_DIR=<path to keyedge dir>
export KEEDGER8R_DIR=<path to keedger8r dir>
```

Clone and Build

```
git clone https://192.168.100.100/rinkai/teep-device.git
cd teep-device
git submodule sync --recursive
git submodule update --init --recursive
make
```

3.3.2 Check teep-device by running hello-app and teep-broker-app

To check teep-device on Unleashed, we need to run TAM server and networking with Unleashed dev board

3.3.3 Run Tamproto (TAM Server)

First start the TAM server on PC. Make sure IP address configured on PC and Unleashed development board.

```
cd tamproto
npm app.js
JWKBaseKeyObject {
  keystore: JWKStore {},
  length: 4096,
  kty: 'RSA',
  kid: 'sWpWma0lDp_RfHKdtKGSVTYQaMIVQaKhESVmzjaW9jc',
  use: '',
  alg: '' }
192.168.0.5
Express HTTP server listening on port 8888
Express HTTPS server listening on port 8443
```

Once TAM server is up, you see above messages

3.3.4 Copy the hello-app and teep-broker-app binaries to Unleased

3.3.4.1 Manual Copy

- Connect to Unleased over serial console then assign IP address `ifconfig eth0 192.168.0.6`
- Copy the binaries from build PC over SSH (user:root, password: sifive)

Here 192.168.0.6 is IP Address of Unleased board

```
scp platform/keystone/build/hello-ta/hello-ta root@192.168.0.6:/root/teep-device
scp platform/keystone/build/hello-app/hello-app root@192.168.0.6:/root/teep-device
scp platform/keystone/build/teep-agent-ta/teep-agent-ta root@192.168.0.6:/root/teep-device
scp platform/keystone/build/teep-broker-app/teep-broker-app root@192.168.0.6:/root/teep-device
scp $KEYSTONE_DIR/sdk/rts/eyrie/eyrie-rt root@192.168.0.6:/root/teep-device
scp platform/keystone/build/libteep/ree/mbedtls/library/lib* root@192.168.0.6:/usr/lib/
scp platform/keystone/build/libteep/ree/libwebsockets/lib/lib* root@192.168.0.6:/usr/lib/
```

3.3.4.2 Write to SD card

Please follow below steps to write the teep-device binaries to SD-card

- Insert SD card to your PC for Unleased
- Edit `platform/keystone/script/sktinst.sh`
 - Check SD-card device name detected on your PC and fix `prefix=?`
 - `export prefix=/dev/mmcblk0`
- execute `script/sktinst.sh` as follows
 - `cd platform/keystone; script/sktinst.sh`
- Move the sd to unleashed board and boot it

3.3.5 Check hello-app and teep-broker-app on Unleased

There are two methods to connect to Unleased.

- Serial Port using minicom (/dev/ttyUSB0)
- Over SSH: `ssh root@192.168.0.6; password is sifive`

Setup environment in Unleased (create /root/env.sh file and add following lines)

```
export PATH=$PATH:/root/teep-device
export TAM_HOST=tamproto-tam.api.1
export TAM_PORT=8888
insmod keystone-driver.ko
```

3.3.5.1 Run hello-app

```
$ source env.sh
[ 2380.618514] keystone_driver: loading out-of-tree module taints kernel.
[ 2380.625305] keystone_enclave: keystone enclave v0.2
$ cd teep-device/
$ ./hello-app hello-ta eyrie-rt
hello TA
$
```

3.3.5.2 Run teep-broker-app

Use the TAM server IP address (i.e 192.168.0.5)

```
./teep-broker-app --tamurl http://192.168.0.5:8888/api/tam.cbor
```

Upon execution, you see following log

```
teep-bro[ 2932.269897] -----[ cut here ]-----
[ 2932.274191] WARNING: CPU: 4 PID: 164 at
/home/aron/projects/ks-0.3/keystone/riscv-linux/mm/page_alloc.c:3926 __alloc_pages_nodemask+0x150/a
[ 2932.287053] Modules linked in: keystone_driver(O)
[ 2932.291716] CPU: 4 PID: 164 Comm: teep-broker-app Tainted: G          W O
4.15.0-00060-g65e929792fb9-dirty #4
[ 2932.301867] Call Trace:
[ 2932.304314] [<0000000036e46dc0>] walk_stackframe+0x0/0xa2
[ 2932.309686] [<00000000893dfefc>] show_stack+0x26/0x34
[ 2932.314725] [<00000000c57ed7ce>] dump_stack+0x5e/0x7c
[ 2932.319759] [<00000000a68ce031>] __warn+0xca/0xe0
[ 2932.324445] [<00000000bec1f8a6>] warn_slowpath_null+0x2c/0x3e
[ 2932.330176] [<00000000e8c56bf2>] __alloc_pages_nodemask+0x14c/0x8da
[ 2932.336426] [<00000000ec1f9596>] __get_free_pages+0xc/0x52
[ 2932.341920] [<000000003e8cccc8>] epm_init+0x158/0x1a0 [keystone_driver]
[ 2932.348502] [<0000000032e4188b>] create_enclave+0x56/0xb0 [keystone_driver]
[ 2932.355447] [<000000008a656a96>] keystone_create_enclave+0x16/0x40 [keystone_driver]
[ 2932.363174] [<000000003bbf2147>] keystone_ioctl1+0x132/0x164 [keystone_driver]
[ 2932.370288] [<00000000755f7993>] do_vfs_ioctl+0x76/0x4f4
[ 2932.375582] [<00000000b88b9c1d>] SyS_ioctl1+0x36/0x60
[ 2932.380533] [<00000000aae667a5>] check_syscall_nnr+0x1e/0x22
[ 2932.386132] ---[ end trace 66814e3a8c80ec12 ]---
ker.c compiled at Feb 16 2021 11:17:21
uri = http://192.168.0.5:8888/api/tam.cbor, cose=0, talist=
[1970/01/01 00:48:56:0796] NOTICE: POST: http://192.168.0.5:8888/api/tam.cbor
[1970/01/01 00:48:56:0798] NOTICE: (hexdump: zero length)
[1970/01/01 00:48:56:0801] NOTICE: created client ssl context for default
[1970/01/01 00:48:56:0802] NOTICE: http://192.168.0.3:8888/api/tam.cbor
[1970/01/01 00:48:56:0861] NOTICE:
[1970/01/01 00:48:56:0862] NOTICE: 0000: 83 01 A4 01 81 01 03 81 00 14 1A 77 77 77 77 04 .....www.
[1970/01/01 00:48:56:0862] NOTICE: 0010: 43 01 02 03 02 C....
[1970/01/01 00:48:56:0871] NOTICE: POST: http://192.168.0.5:8888/api/tam.cbor
[1970/01/01 00:48:56:0871] NOTICE:
[1970/01/01 00:48:56:0871] NOTICE: 0000: 82 02 A4 14 1A 77 77 77 77 08 80 0E 80 0F 80 .....www.....
[1970/01/01 00:48:56:0872] NOTICE:
[1970/01/01 00:48:56:0873] NOTICE: created client ssl context for default
[1970/01/01 00:48:56:0874] NOTICE: http://192.168.0.5:8888/api/tam.cbor
[1970/01/01 00:48:56:0962] NOTICE:
[1970/01/01 00:48:56:0962] NOTICE: 0000: 82 03 A2 0A 81 59 01 37 A2 02 58 72 81 58 6F D2 .....Y.7..Xr.Xo.
[1970/01/01 00:48:56:0963] NOTICE: 0010: 84 43 A1 01 26 A0 58 24 82 02 58 20 75 80 7C 54 .C.&.X$.X u.|T
[1970/01/01 00:48:56:0963] NOTICE: 0020: 62 40 D2 14 E5 7B D5 C4 6A 7C E5 2D ED B0 3D 0E b@...{.j|.---=.
[1970/01/01 00:48:56:0964] NOTICE: 0030: CC 80 75 F3 F7 E0 65 B3 60 CE AD 85 58 40 54 81 ..u...e.'...X@T.
[1970/01/01 00:48:56:0964] NOTICE: 0040: 49 CD CA D8 17 72 CC EA 61 4A 19 99 05 AB 97 33 I....r..aJ.....3
[1970/01/01 00:48:56:0965] NOTICE: 0050: EA 48 D7 1F 13 AE 33 0D 47 FF F5 B8 6C 5C 9B 7A .H....3.G...l\z
[1970/01/01 00:48:56:0965] NOTICE: 0060: BB 12 BC 2D FE 9C 20 6A C8 7F E2 28 58 74 E0 74 .XV....P.kJS....
[1970/01/01 00:48:56:0965] NOTICE: 0070: A3 BD C4 DA B9 20 C4 37 35 8F 67 46 90 76 03 58 .....j... (Xt.t
[1970/01/01 00:48:56:0966] NOTICE: 0080: BE A5 01 01 02 01 03 58 60 A2 02 44 81 81 41 00 .....75.gF.v.X
[1970/01/01 00:48:56:0966] NOTICE: 0090: 04 58 56 86 14 A4 01 50 FA 6B 4A 53 D5 AD 5F DF .....X`.D..A.
[1970/01/01 00:48:56:0967] NOTICE: 00A0: BE 9D E6 63 E4 D4 1F FE 02 50 14 92 AF 14 25 69 .c....P....%i
[1970/01/01 00:48:56:0967] NOTICE: 00B0: 5E 48 BF 42 9B 2D 51 F2 AB 45 03 58 24 82 02 58 ^H.B.-Q...E.X$.X
[1970/01/01 00:48:56:0968] NOTICE: 00C0: 20 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE ..."3DUfw.....
[1970/01/01 00:48:56:0968] NOTICE: 00D0: FF 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 ..#Eg.....vT2
[1970/01/01 00:48:56:0969] NOTICE: 00E0: 10 0E 19 87 D0 01 F6 02 F6 09 58 4E 86 13 A1 15 .....XN....
[1970/01/01 00:48:56:0969] NOTICE: 00F0: 78 44 68 74 74 70 3A 2F 2F 31 39 32 2E 31 36 38 xDhttp://192.168
[1970/01/01 00:48:56:0970] NOTICE: 0100: 2E 31 31 2E 33 3A 38 38 38 38 2F 54 41 73 2F 38 .0.5:8888/TAs/8
[1970/01/01 00:48:56:0970] NOTICE: 0110: 64 38 32 35 37 33 61 2D 39 32 36 64 2D 34 37 35 d82573a-926d-475
[1970/01/01 00:48:56:0971] NOTICE: 0120: 34 2D 39 33 35 33 2D 33 32 64 63 32 39 39 39 37 4-9353-32dc29997
[1970/01/01 00:48:56:0971] NOTICE: 0130: 66 37 34 2E 74 61 15 F6 03 F6 0A 43 82 03 F6 14 f74.ta....C....
[1970/01/01 00:48:56:0972] NOTICE: 0140: 1A 77 77 77 78 .wwwx
[1970/01/01 00:48:56:0972] NOTICE:
[1970/01/01 00:48:56:0983] NOTICE: GET: http://192.168.0.5:8888/TAs/8d82573a-926d-4754-9353-32dc29997f74.ta
[1970/01/01 00:48:56:0984] NOTICE: created client ssl context for default
[1970/01/01 00:48:56:0985] NOTICE: http://192.168.0.5:8888/TAs/8d82573a-926d-4754-9353-32dc29997f74.ta
teep_message.unwrap.ta_image: msg len 234110
Decrypt
Decrypt OK: length 174887
Verify
Signature OK 0 130552
ta_store_install: ta_image_len = 130552 ta_name=8d82573a-926d-4754-9353-32dc29997f74
[1970/01/01 00:49:01:9453] NOTICE: POST: http://192.168.0.5:8888/api/tam.cbor
[1970/01/01 00:49:01:9454] NOTICE:
[1970/01/01 00:49:01:9454] NOTICE: 0000: 82 05 A1 14 1A 77 77 77 77 .....www
[1970/01/01 00:49:01:9454] NOTICE:
[1970/01/01 00:49:01:9456] NOTICE: created client ssl context for default
[1970/01/01 00:49:01:9457] NOTICE: http://192.168.0.5:8888/api/tam.cbor
[1970/01/01 00:49:01:9505] NOTICE: (hexdump: zero length)
```


3.4 OPTEE

Build `teep-device` with OPTEE. So make sure OPTEE and its supporting sources have been build already.

3.4.1 Clone and Build

Prepare the environment setup

```
export TEE=optee
export OPTEE_DIR=<optee.3.9.0.rpi3 dir>
export PATH=$PATH:$OPTEE_DIR/toolchains/aarch64/bin:$OPTEE_DIR/toolchains/aarch32/bin
```

Clone and Build

```
git clone https://192.168.100.100/rinkai/teep-device.git
cd teep-device
git submodule sync --recursive
git submodule update --init --recursive
make
```

3.4.2 Check teep-device by running hello-app and teep-broker-app on RPI3

To check teep-device on RPI3, we need to run TAM server on PC and networking with RPI3 board

3.4.3 Run Tamproto (TAM Server)

First start the TAM server on PC. Make sure IP address configured on PC and RPI3 board.

```
cd tamproto
npm app.js
JWKBaseKeyObject {
  keystore: JWKStore {},
  length: 4096,
  kty: 'RSA',
  kid: 'sWpWma0lDp_RfHKdtkGSVTYQaMIVQaKhESVmzjaW9jc',
  use: '',
  alg: '' }
192.168.0.5
Express HTTP server listening on port 8888
Express HTTPS server listening on port 8443
```

Once TAM server is up, you see above messages

3.4.4 Copy the hello-app and teep-broker-app binaries to RPI3

3.4.4.1 Copy binaries over SSH to RPI3

- Connect to RPI3 over serial console(/dev/ttyUSB0) then assign IP address `ifconfig eth0 192.168.0.7`
- Copy the binaries from build PC over SSH (user:root) to RPI3

TODO - Further update required

3.4.4.2 Write to SD card

Please follow below steps to write the teep-device binaries to SD-card

- Insert SD card to your PC for Unleashed
- Copy the binaries to SD card
- Move the sd to RPI3 board and boot it

TODO - Further update required

3.4.5 Check hello-app and teep-broker-app on RPI3

There are two methods to connect to RPI3.

- Serial Port using minicom (/dev/ttyUSB0)
- Over SSH: `ssh root@192.168.0.7`

TODO - Further update required

3.4.5.1 Run hello-app

TODO - Further update required

3.4.5.2 Run teep-broker-app

Use the TAM server IP address (i.e 192.168.0.3)

```
./teep-broker-app --tamurl http://192.168.0.3:8888/api/tam.cbor
```

Execution logs

TODO - Further update required

3.5 SGX

Build teep-device with SGX. Make sure SGX and its supporting sources have been build already.

3.5.1 Clone and Build on SGX

Prepare the environment setup

```
export TEE=pc
source /opt/intel/sgxsdk/environment
```

Clone and Build

```
git clone https://192.168.100.100/rinkai/teep-device.git
cd teep-device
git submodule sync --recursive
git submodule update --init --recursive
make
```

3.5.2 Check teep-device by running hello-app & teep-broker-app on SGX

To check teep-device on SGX, we need to run TAM server on PC and networking with SGX machine

3.5.3 Run Tamproto (TAM Server)

First start the TAM server on PC. Make sure IP address configured on PC and SGX machine.

```
<p />
cd tamproto
npm app.js
JWKBaseKeyObject {
  keystore: JWKStore {},
  length: 4096,
  kty: 'RSA',
  kid: 'sWpWma0lDp_RfHKdtkGSVTYQaMIVQaKhESVmzjaW9jc',
  use: '',
  alg: '' }
192.168.0.5
Express HTTP  server listening on port 8888
Express HTTPS server listening on port 8443
<p />
```

Once TAM server is up, you see above messages

3.5.4 Copy hello-app & teep-broker-app binaries to SGX

Copy the binaries to SGX/NUC machine over SSH

TODO - Further update required

If source is build natively on the SGX/NUC machine, then just copy the binaries to test PATH.

TODO - Further update required

3.5.5 Check hello-app and teep-broker-app on SGX

TODO - Further update required

3.5.5.1 Run hello-app

TODO - Further update required

3.5.5.2 Run teep-broker-app

If your TAM server IP address is 192.168.0.3, then you

```
./teep-broker-app --tamurl http://192.168.0.3:8888/api/tam.cbor
```

Execution logs

TODO - Further update required

4 Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

broker.ctx	10
Command	10
CommandQueue	11
lao_rpc_io	12

libteep_async	13
out_fct_wrap_type	14
ta_manifest	14
teep_agent_session	15
teep_buffer_array	16
teep_message	17
teep_message_encoder	21
teep_tc_info	22
teep_tc_info_array	23
teep_uint32_array	24
teep_uint32_option	24

5 File Index

5.1 File List

Here is a list of all files with brief descriptions:

teep-device/hello-app/keystone-main.cpp	25
teep-device/hello-app/optee-main.c	30
teep-device/hello-ta/hello-ta.c	32
teep-device/hello-ta/user_ta_header_defines.h	35
teep-device/libteep/lib/libteep.c	37
teep-device/libteep/lib/libteep.h	40
teep-device/teep-agent-ta/ta-store.c	45
teep-device/teep-agent-ta/ta-store.h	48
teep-device/teep-agent-ta/teep-agent-ta.c	49
teep-device/teep-agent-ta/teep-agent-ta.h	55
teep-device/teep-agent-ta/tools.c	55
teep-device/teep-agent-ta/user_ta_header_defines.h	36
teep-device/teep-agent-ta/vsnprintf.c	58
teep-device/teep-agent-ta/sys/time.h	44
teep-device/teep-broker-app/http-lws.c	69
teep-device/teep-broker-app/http.h	72

teep-device/teep-broker-app/teec-keystone.cpp	73
teep-device/teep-broker-app/teec-pc.c	76
teep-device/teep-broker-app/teep-broker.c	78

6 Class Documentation

6.1 broker_ctx Struct Reference

Public Attributes

- TEEC_Context [tee_context](#)
- TEEC_Session [tee_session](#)

6.1.1 Member Data Documentation

6.1.1.1 tee_context TEEC_Context broker_ctx::tee_context

6.1.1.2 tee_session TEEC_Session broker_ctx::tee_session

The documentation for this struct was generated from the following file:

- [teep-device/teep-broker-app/teep-broker.c](#)

6.2 Command Struct Reference

Public Attributes

- invoke_command_t [command](#)
- TEEC_Operation * [operation](#)
- unsigned int [command_result](#)

6.2.1 Member Data Documentation

6.2.1.1 command invoke_command_t Command::command

6.2.1.2 command_result `unsigned int Command::command_result`

6.2.1.3 operation `TEEC_Operation* Command::operation`

The documentation for this struct was generated from the following file:

- [teep-device/teep-broker-app/teec-keystone.cpp](#)

6.3 CommandQueue Class Reference

Public Member Functions

- `invoke_command_t` [pull_invoke_command](#) ()
- `param_buffer_t` [read_invoke_param](#) (int index, unsigned int offset)
- void [write_invoke_param](#) (int index, unsigned int offset, unsigned int size, const char *buf)
- void [put_invoke_command_result](#) (const invoke_command_t &cmd, unsigned int result)
- void [put_invoke_command](#) (int commandID, TEEC_Operation *operation)
- int [pull_invoke_command_result](#) ()

6.3.1 Member Function Documentation

6.3.1.1 pull_invoke_command() `invoke_command_t CommandQueue::pull_invoke_command () [inline]`

6.3.1.2 pull_invoke_command_result() `int CommandQueue::pull_invoke_command_result () [inline]`

6.3.1.3 put_invoke_command() `void CommandQueue::put_invoke_command (`
`int commandID,`
`TEEC_Operation * operation) [inline]`

6.3.1.4 put_invoke_command_result() `void CommandQueue::put_invoke_command_result (`
`const invoke_command_t & cmd,`
`unsigned int result) [inline]`

6.3.1.5 read_invoke_param() `param.buffer_t CommandQueue::read_invoke_param (`
 `int index,`
 `unsigned int offset) [inline]`

6.3.1.6 write_invoke_param() `void CommandQueue::write_invoke_param (`
 `int index,`
 `unsigned int offset,`
 `unsigned int size,`
 `const char * buf) [inline]`

The documentation for this class was generated from the following file:

- [teep-device/teep-broker-app/teec-keystone.cpp](#)

6.4 lao_rpc_io Struct Reference

Public Attributes

- `void * in`
- `size_t in_len`
- `void * out`
- `size_t out_len`

6.4.1 Member Data Documentation

6.4.1.1 in `void* lao_rpc_io::in`

6.4.1.2 in_len `size_t lao_rpc_io::in_len`

6.4.1.3 out `void* lao_rpc_io::out`

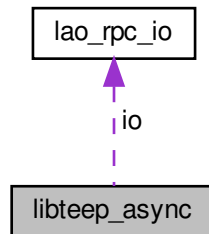
6.4.1.4 out_len `size_t lao_rpc_io::out_len`

The documentation for this struct was generated from the following file:

- [teep-device/teep-broker-app/http-lws.c](#)

6.5 libteep_async Struct Reference

Collaboration diagram for libteep_async:



Public Attributes

- struct `lao_rpc_io` * `io`
- struct `lws` * `wsi`
- `size_t` `max_out_len`
- `int` `http_resp`
- `tam_result` `result`

6.5.1 Member Data Documentation

6.5.1.1 http_resp `int libteep_async::http_resp`

6.5.1.2 io `struct lao_rpc_io* libteep_async::io`

6.5.1.3 max_out_len `size_t libteep_async::max_out_len`

6.5.1.4 result `tam_result libteep_async::result`

6.5.1.5 wsi `struct lws* libteep_async::wsi`

The documentation for this struct was generated from the following file:

- [teep-device/teep-broker-app/http-lws.c](#)

6.6 out_fct_wrap_type Struct Reference

Public Attributes

- `void(* fct)(char character, void *arg)`
- `void *arg`

6.6.1 Member Data Documentation

6.6.1.1 arg `void* out_fct_wrap_type::arg`

6.6.1.2 fct `void(* out_fct_wrap_type::fct)(char character, void *arg)`

The documentation for this struct was generated from the following file:

- [teep-device/teep-agent-ta/vsnprintf.c](#)

6.7 ta_manifest Struct Reference

Public Attributes

- `char id[128]`
- `char uri[TEEP_MAX_URI_LEN]`

6.7.1 Member Data Documentation

6.7.1.1 id `char ta_manifest::id[128]`

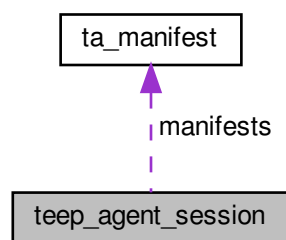
6.7.1.2 uri `char ta_manifest::uri[TEEP_MAX_URI_LEN]`

The documentation for this struct was generated from the following file:

- [teep-device/teep-agent-ta/teep-agent-ta.c](#)

6.8 teep_agent_session Struct Reference

Collaboration diagram for teep_agent_session:



Public Attributes

- enum [agent_state](#) `state`
- char [tam_uri](#) [TEEP_MAX_URI_LEN]
- struct broker_task * [on_going_task](#)
- struct broker_task [task_buffer](#)
- uint64_t [token](#)
- struct [ta_manifest](#) * [manifests](#)
- size_t [manifests_len](#)
- size_t [download_ta_index](#)
- uint64_t [data_item_requested](#)

6.8.1 Member Data Documentation

6.8.1.1 data_item_requested `uint64_t teep_agent_session::data_item_requested`

6.8.1.2 download.ta.index `size_t teep_agent_session::download.ta.index`

6.8.1.3 manifests `struct ta_manifest* teep_agent_session::manifests`

6.8.1.4 manifests.len `size_t teep_agent_session::manifests.len`

6.8.1.5 on_going_task `struct broker_task* teep_agent_session::on_going_task`

6.8.1.6 state `enum agent_state teep_agent_session::state`

6.8.1.7 tam_uri `char teep_agent_session::tam_uri[TEEP_MAX_URI_LEN]`

6.8.1.8 task_buffer `struct broker_task teep_agent_session::task_buffer`

6.8.1.9 token `uint64_t teep_agent_session::token`

The documentation for this struct was generated from the following file:

- `teep-device/teep-agent-ta/teep-agent-ta.c`

6.9 teep_buffer_array Struct Reference

```
#include <libteep.h>
```

Public Attributes

- bool `have_value`
- UsefulBufC * `array`
- size_t `len`

6.9.1 Member Data Documentation

6.9.1.1 array UsefulBufC* teep_buffer_array::array

6.9.1.2 have_value bool teep_buffer_array::have_value

6.9.1.3 len size_t teep_buffer_array::len

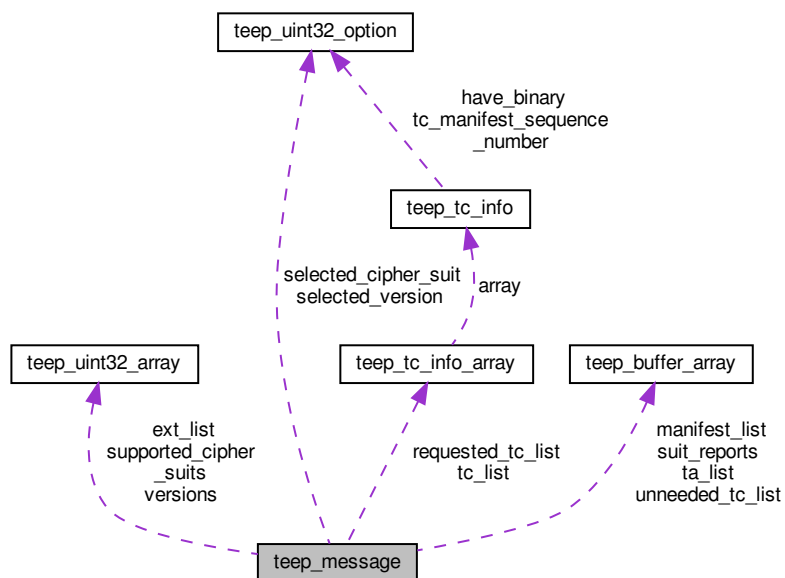
The documentation for this struct was generated from the following file:

- teep-device/libteep/lib/libteep.h

6.10 teep_message Struct Reference

```
#include <libteep.h>
```

Collaboration diagram for teep_message:



Public Attributes

- enum [teep_message_type](#) type
- uint64_t [token](#)
- union {
 - struct {
 - struct [teep_uint32_array](#) supported_cipher_suits
 - UsefulBufC [challenge](#)
 - struct [teep_uint32_array](#) versions
 - UsefulBufC [ocsp_data](#)
 - uint64_t [data_item_requested](#)
 - } [query_request](#)
 - struct {
 - struct [teep_uint32_option](#) selected_cipher_suit
 - struct [teep_uint32_option](#) selected_version
 - UsefulBufC [evidence_format](#)
 - UsefulBufC [evidence](#)
 - struct [teep_tc_info_array](#) tc_list
 - struct [teep_tc_info_array](#) requested_tc_list
 - struct [teep_buffer_array](#) unneeded_tc_list
 - struct [teep_uint32_array](#) ext_list
 - } [query_response](#)
 - struct {
 - struct [teep_buffer_array](#) manifest_list
 - } [teep_install](#)
 - struct {
 - struct [teep_buffer_array](#) ta_list
 - } [teep_delete](#)
 - struct {
 - UsefulBufC [msg](#)
 - struct [teep_buffer_array](#) suit_reports
 - } [teep_success](#)
 - struct {
 - int64_t [err_code](#)
 - UsefulBufC [err_msg](#)
 - struct [teep_uint32_array](#) supported_cipher_suits
 - struct [teep_uint32_array](#) versions
 - struct [teep_buffer_array](#) suit_reports
 - } [teep_error](#)

6.10.1 Member Data Documentation

6.10.1.1 union { ... }

6.10.1.2 challenge UsefulBufC teep_message::challenge

6.10.1.3 data_item_requested uint64_t teep_message::data_item_requested

6.10.1.4 err.code int64_t teep_message::err.code

6.10.1.5 err.msg UsefulBufC teep_message::err.msg

6.10.1.6 evidence UsefulBufC teep_message::evidence

6.10.1.7 evidence_format UsefulBufC teep_message::evidence_format

6.10.1.8 ext_list struct [teep_uint32_array](#) teep_message::ext_list

6.10.1.9 manifest_list struct [teep_buffer_array](#) teep_message::manifest_list

6.10.1.10 msg UsefulBufC teep_message::msg

6.10.1.11 ocsp_data UsefulBufC teep_message::ocsp_data

6.10.1.12 struct { ... } teep_message::query_request

6.10.1.13 struct { ... } teep_message::query_response

6.10.1.14 requested_tc_list struct `teep_tc_info_array` `teep_message::requested_tc_list`

6.10.1.15 selected_cipher_suit struct `teep_uint32_option` `teep_message::selected_cipher_suit`

6.10.1.16 selected_version struct `teep_uint32_option` `teep_message::selected_version`

6.10.1.17 suit_reports struct `teep_buffer_array` `teep_message::suit_reports`

6.10.1.18 supported_cipher_suits struct `teep_uint32_array` `teep_message::supported_cipher_suits`

6.10.1.19 ta_list struct `teep_buffer_array` `teep_message::ta_list`

6.10.1.20 tc_list struct `teep_tc_info_array` `teep_message::tc_list`

6.10.1.21 struct { ... } `teep_message::teep_delete`

6.10.1.22 struct { ... } `teep_message::teep_error`

6.10.1.23 struct { ... } `teep_message::teep_install`

6.10.1.24 struct { ... } `teep_message::teep_success`

6.10.1.25 token `uint64_t teep_message::token`

6.10.1.26 type `enum teep_message_type teep_message::type`

6.10.1.27 unneeded_tc_list `struct teep_buffer_array teep_message::unneeded_tc_list`

6.10.1.28 versions `struct teep_uint32_array teep_message::versions`

The documentation for this struct was generated from the following file:

- [teep-device/libteep/lib/libteep.h](#)

6.11 teep_message_encoder Struct Reference

```
#include <libteep.h>
```

Public Attributes

- `QCBOREncodeContext` [EC](#)

6.11.1 Member Data Documentation

6.11.1.1 EC `QCBOREncodeContext teep_message_encoder::EC`

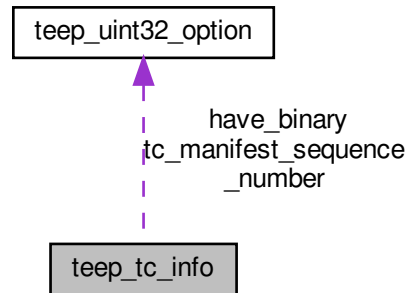
The documentation for this struct was generated from the following file:

- [teep-device/libteep/lib/libteep.h](#)

6.12 teep_tc_info Struct Reference

```
#include <libteep.h>
```

Collaboration diagram for teep_tc_info:



Public Attributes

- UsefulBufC [component_id](#)
- struct [teep_uint32_option](#) `tc_manifest_sequence_number`
- struct [teep_uint32_option](#) `have_binary`

6.12.1 Member Data Documentation

6.12.1.1 `component_id` UsefulBufC `teep_tc_info::component_id`

6.12.1.2 `have_binary` struct [teep_uint32_option](#) `teep_tc_info::have.binary`

6.12.1.3 `tc_manifest_sequence_number` struct [teep_uint32_option](#) `teep_tc_info::tc_manifest_sequence↔
_number`

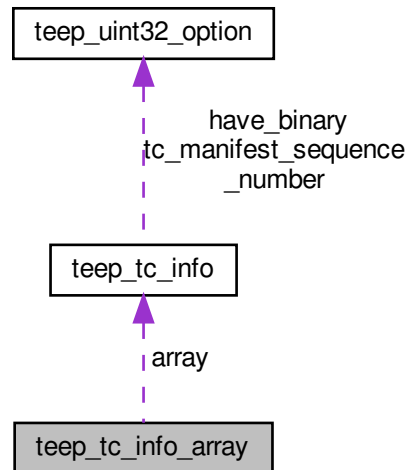
The documentation for this struct was generated from the following file:

- `teep-device/libteep/lib/libteep.h`

6.13 teep_tc_info_array Struct Reference

```
#include <libteep.h>
```

Collaboration diagram for teep_tc_info_array:



Public Attributes

- bool `have_value`
- struct `teep_tc_info` * `array`
- size_t `len`

6.13.1 Member Data Documentation

6.13.1.1 array struct `teep_tc_info`* `teep_tc_info_array::array`

6.13.1.2 have_value bool `teep_tc_info_array::have_value`

6.13.1.3 len size_t `teep_tc_info_array::len`

The documentation for this struct was generated from the following file:

- `teep-device/libteep/lib/libteep.h`

6.14 teep_uint32_array Struct Reference

```
#include <libteep.h>
```

Public Attributes

- bool [have_value](#)
- uint32_t * [array](#)
- size_t [len](#)

6.14.1 Member Data Documentation

6.14.1.1 array uint32_t* teep_uint32_array::array

6.14.1.2 have_value bool teep_uint32_array::have_value

6.14.1.3 len size_t teep_uint32_array::len

The documentation for this struct was generated from the following file:

- teep-device/libteep/lib/[libteep.h](#)

6.15 teep_uint32_option Struct Reference

```
#include <libteep.h>
```

Public Attributes

- bool [have_value](#)
- uint32_t [value](#)

6.15.1 Member Data Documentation

6.15.1.1 have_value bool teep_uint32_option::have_value

6.15.1.2 value uint32_t teep_uint32_option::value

The documentation for this struct was generated from the following file:

- [teep-device/libteep/lib/libteep.h](#)

7 File Documentation

7.1 teep-device/docs/cloning_and_building.md File Reference

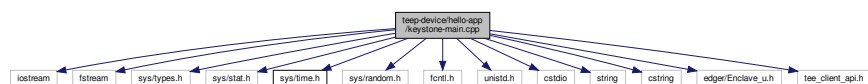
7.2 teep-device/docs/overview_of_tee-device.md File Reference

7.3 teep-device/docs/tee-device_operations.md File Reference

7.4 teep-device/hello-app/keystone-main.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/time.h>
#include <sys/random.h>
#include <fcntl.h>
#include <unistd.h>
#include <cstdio>
#include <string>
#include <cstring>
#include "edger/Enclave_u.h"
#include "tee_client_api.h"
```

Include dependency graph for keystone-main.cpp:



Functions

- int [main](#) (int argc, char **argv)
- EDGE_EXTERN_C_BEGIN invoke_command_t [ocall_pull_invoke_command](#) ()
- param_buffer_t [ocall_read_invoke_param](#) (int index, size_t offset)
[ocall_read_invoke_param\(\)](#) is for returning the size of the buffer.
- void [ocall_write_invoke_param](#) (int index, size_t, size_t size, const char *buf)
[read_invoke_param\(\)](#) is used for calling the [write_invoke_param\(\)](#).
- void [ocall_put_invoke_command_result](#) (invoke_command_t cmd, unsigned int result)
- EDGE_EXTERN_C_END TEEC_Result [TEEC_InitializeContext](#) (const char *name, TEEC_Context *context)
- void [TEEC_FinalizeContext](#) (TEEC_Context *context)
- TEEC_Result [TEEC_OpenSession](#) (TEEC_Context *context, TEEC_Session *session, const TEEC.UUID *destination, uint32_t connectionMethod, const void *connectionData, TEEC_Operation *operation, uint32_t *returnOrigin)
- void [TEEC_CloseSession](#) (TEEC_Session *session)
- TEEC_Result [TEEC_InvokeCommand](#) (TEEC_Session *session, uint32_t commandID, TEEC_Operation *operation, uint32_t *returnOrigin)

7.4.1 Function Documentation

7.4.1.1 main() `int main (`
 `int argc,`
 `char ** argv)`

[main\(\)](#) - To launch the enclave.

The enclave parameters contain such as size of free memory and the address/size of the untrusted shared buffer. In order to handle the edge calls (including system calls), the enclave must register the edge call handler and initialize the buffer addresses and finally the host launches the enclave.

Parameters

<i>argc</i>	integer argument count.
<i>argv</i>	character type argument vector.

Returns

0 based on the enclave launch.

7.4.1.2 ocall_pull_invoke_command() `EDGE_EXTERN_BEGIN invoke_command_t ocall_pull_invoke_command`
`()`

[ocall_pull_invoke_command\(\)](#) - Invokes the pull command.

7.4.1.3 ocall_put_invoke_command_result() `void ocall_put_invoke_command_result (`
 `invoke_command_t cmd,`
 `unsigned int result)`

[ocall_put_invoke_command_result\(\)](#) - Put the invokes command result

Parameters

<i>cmd</i>	invoke command.
<i>result</i>	invoke command result.

7.4.1.4 ocall_read_invoke_param() `param_buffer_t ocall_read_invoke_param (`
 `int index,`
 `size_t offset)`

[ocall_read_invoke_param\(\)](#) is for returning the size of the buffer.

Parameters

in	<i>index</i>	integer type parameter position.
in	<i>offset</i>	indicating the distance between the beginning of the object.

return the param_buffer_t structure

Returns

param_buffer_t structure.

7.4.1.5 ocall_write_invoke_param() `void ocall_write_invoke_param (`
 `int index,`
 `size_t ,`
 `size_t size,`
 `const char * buf)`

read.invoke_param() is used for calling the write_invoke_param().

Parameters

in	<i>index</i>	integer type parameter position.
in	<i>offset</i>	indicating the distance between the beginning of the object.

7.4.1.6 TEEC_CloseSession() `void TEEC_CloseSession (`
 `TEEC_Session * session)`

[TEEC_CloseSession\(\)](#) - Closes the session which has been opened with the specific trusted application.

Parameters

<i>session</i>	The opened session to close.
----------------	------------------------------

7.4.1.7 TEEC_FinalizeContext() `void TEEC_FinalizeContext (`
 `TEEC_Context * context)`

[TEEC_FinalizeContext\(\)](#) - Destroys a context holding connection information on the specific TEE.

This function destroys an initialized TEE context, closing the connection between the client application and the TEE. This function must only be called when all sessions related to this TEE context have been closed and all shared memory blocks have been released.

Parameters

<i>context</i>	The context to be destroyed.
----------------	------------------------------

7.4.1.8 TEEC.InitializeContext() `EDGE_EXTERNC_END TEEC_Result TEEC.InitializeContext (`
`const char * name,`
`TEEC_Context * context)`

[TEEC.InitializeContext\(\)](#) - Initializes a context holding connection information on the specific TEE, designated by the name string.

Parameters

<i>name</i>	A zero-terminated string identifying the TEE to connect to. If name is set to NULL, the default TEE is connected to. NULL is the only supported value in this version of the API implementation.
<i>context</i>	The context structure which is to be initialized.

Returns

TEEC_SUCCESS The initialization was successful.

7.4.1.9 TEEC.InvokeCommand() `TEEC_Result TEEC.InvokeCommand (`
`TEEC_Session * session,`
`uint32_t commandID,`
`TEEC_Operation * operation,`
`uint32_t * returnOrigin)`

[TEEC.InvokeCommand\(\)](#) - Executes a command in the specified trusted application.

Parameters

<i>session</i>	A handle to an open connection to the trusted application.
<i>commandID</i>	Identifier of the command in the trusted application to invoke.
<i>operation</i>	An operation structure to use in the invoke command. May be set to NULL to signify no operation structure needed.
<i>returnOrigin</i>	A parameter which will hold the error origin if this function returns any value other than TEEC_SUCCESS.

Returns

TEEC_SUCCESS OpenSession successfully opened a new session.

TEEC_Result Something failed.

7.4.1.10 TEEC.OpenSession() `TEEC_Result TEEC.OpenSession (`
`TEEC_Context * context,`
`TEEC_Session * session,`
`const TEEC_UUID * destination,`
`uint32_t connectionMethod,`
`const void * connectionData,`
`TEEC_Operation * operation,`
`uint32_t * returnOrigin)`

TEEC.OpenSession() - Opens a new session with the specified trusted application.

Parameters

<i>context</i>	The initialized TEE context structure in which scope to open the session.
<i>session</i>	The session to initialize.
<i>destination</i>	A structure identifying the trusted application with which to open a session.
<i>connectionMethod</i>	The connection method to use.
<i>connectionData</i>	Any data necessary to connect with the chosen connection method. Not supported, should be set to NULL.
<i>operation</i>	An operation structure to use in the session. May be set to NULL to signify no operation structure needed.
<i>returnOrigin</i>	A parameter which will hold the error origin if this function returns any value other than TEEC_SUCCESS.

Returns

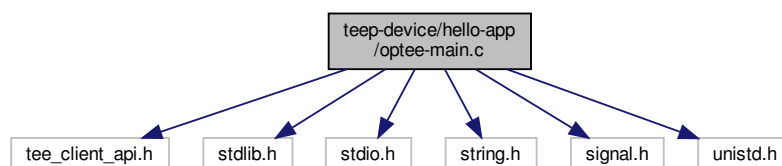
TEEC_SUCCESS OpenSession successfully opened a new session.

TEEC_Result Something failed.

7.5 teep-device/hello-app/optee-main.c File Reference

```
#include <tee_client_api.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <signal.h>
#include <unistd.h>
```

Include dependency graph for optee-main.c:



Functions

- TEEC_Result [sp_hello_app](#) ()
- int [main](#) (int argc, char *argv[])

Variables

- const TEEC_UUID [uuid](#)
- TEEC_Context * [context](#)
- TEEC_Session * [session](#)
- TEEC_SharedMemory [shm](#)
- uint8_t * [filecontents](#)
- size_t [file_length](#)

7.5.1 Function Documentation

7.5.1.1 main()

```
int main (
    int argc,
    char * argv[ ] )
```

[main\(\)](#) - Creates context, session and starts hello-app.

In this function the context is initialized for connecting to the TEE by calling [TEEC_InitializeContext\(\)](#). If the session value is null it invokes "bail2". Upon the valid session, [TEEC_OpenSession\(\)](#) is invoked. If the session is opened successfully then it invokes [sp_hello_app\(\)](#) and returns 0. Upon the failure of [sp_hello_app\(\)](#) it will go to "bail4".

Parameters

<i>argc</i>	argument count
<i>argv[]</i>	argument list

Returns

0 on success

7.5.1.2 sp_hello_app()

```
TEEC_Result sp_hello_app ( )
```

[sp_hello_app\(\)](#) - Start the hello app.

This function invokes the command and copies 0 to the first character of TEEC_Operation type op. Then the command is invoked within the specified session by calling [TEEC_InvokeCommand\(\)](#). Upon success it returns the n value.

Returns

n TEEC_Result on success invoke command.

7.5.2 Variable Documentation

7.5.2.1 context `TEEC_Context* context`

7.5.2.2 file_length `size_t file_length`

7.5.2.3 filecontents `uint8_t* filecontents`

7.5.2.4 session `TEEC_Session* session`

7.5.2.5 shm `TEEC_SharedMemory shm`

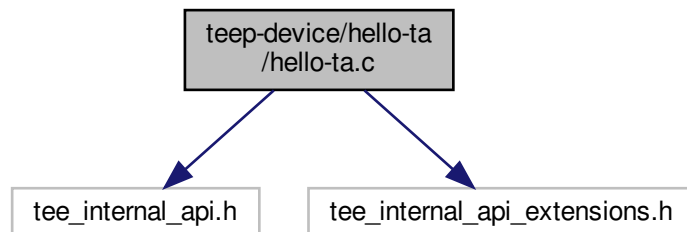
7.5.2.6 uuid `const TEEC_UUID uuid`

Initial value:

```
=  
{ 0x8d82573a, 0x926d, 0x4754,  
  { 0x93, 0x53, 0x32, 0xdc, 0x29, 0x99, 0x7f, 0x74 } }
```

7.6 teep-device/hello-ta/hello-ta.c File Reference

```
#include <tee_internal_api.h>  
#include <tee_internal_api_extensions.h>  
Include dependency graph for hello-ta.c:
```



Macros

- `#define STR_TRACE_USER_TA "AIST_TB"`

Functions

- `TEE_Result TA_CreateEntryPoint (void)`
- `void TA_DestroyEntryPoint (void)`
- `TEE_Result TA_OpenSessionEntryPoint (uint32_t param_types, TEE_Param __maybe_unused params[4], void __maybe_unused **sess_ctx)`
- `void TA_CloseSessionEntryPoint (void __maybe_unused *sess_ctx)`
- `TEE_Result TA_InvokeCommandEntryPoint (void __maybe_unused *sess_ctx, uint32_t cmd_id, uint32_t param_types, TEE_Param params[4])`

7.6.1 Macro Definition Documentation

7.6.1.1 STR_TRACE_USER_TA `#define STR_TRACE_USER_TA "AIST_TB"`

7.6.2 Function Documentation

7.6.2.1 TA_CloseSessionEntryPoint() `void TA_CloseSessionEntryPoint (void __maybe_unused * sess_ctx)`

[TA_CloseSessionEntryPoint\(\)](#) - It is called when the client closes a session and disconnects from the Trusted Application instance.

Parameters

<code>sess_ctx</code>	The value of the void* opaque data pointer set by the Trusted Application.
-----------------------	--

7.6.2.2 TA_CreateEntryPoint() `TEE_Result TA_CreateEntryPoint (void)`

[TA_CreateEntryPoint\(\)](#) - Trusted Application's constructor,

This function is used to register instance data. the implementation of this constructor can use either global variables or the function `TEE_SetInstanceData`.

Returns

It returns `TEE_SUCCESS`.

7.6.2.3 TA_DestroyEntryPoint() `void TA_DestroyEntryPoint (`
`void)`

[TA_DestroyEntryPoint\(\)](#) - Trusted Application's destructor.

When the function `TA_DestroyEntryPoint` is called, the Framework guarantees that no client session is currently open. Once the call to `TA_DestroyEntryPoint` has been completed, no other entry point of this instance will ever be called.

7.6.2.4 TA_InvokeCommandEntryPoint() `TEE_Result TA_InvokeCommandEntryPoint (`
`void __maybe_unused * sess_ctx,`
`uint32_t cmd_id,`
`uint32_t param_types,`
`TEE_Param params[4])`

The [TA_InvokeCommandEntryPoint\(\)](#) - When the client invokes a command within the given session.

The Trusted Application can access the parameters sent by the client through the `paramTypes` and `params` arguments. It can also use these arguments to transfer response data back to the client. A specification of how to handle the operation parameters. During the call to `TA_InvokeCommandEntryPoint` the client may request to cancel the operation.

Parameters

<i>sess_ctx</i>	The value of the <code>void*</code> opaque data pointer set by the Trusted Application in the function <code>TA_OpenSessionEntryPoint</code>
<i>cmd_id</i>	A Trusted Application-specific code that identifies the command to be invoked
<i>param_types</i>	The types of the four parameters.
<i>params</i>	A pointer to an array of four parameters

Returns

Its return `TEE_ERROR_NOT_IMPLEMENTED`.

7.6.2.5 TA_OpenSessionEntryPoint() `TEE_Result TA_OpenSessionEntryPoint (`
`uint32_t param_types,`
`TEE_Param __maybe_unused params[4],`
`void __maybe_unused ** sess_ctx)`

[TA_OpenSessionEntryPoint\(\)](#) - When a client requests to open a session with the Trusted Application.

This function client can specify parameters in an open operation which are passed to the Trusted Application instance in the arguments `paramTypes` and `params`.

Parameters

<i>param_types</i>	The types of the four parameters
<i>params</i>	A pointer to an array of four parameters
<i>sess_ctx</i>	A pointer to a variable that can be filled by the Trusted Application instance with pointer.

Returns

It returns TEEC_SUCCESS.

7.7 teep-device/hello-ta/user_ta_header_defines.h File Reference

Macros

- `#define TA_UUID`
- `#define TA_FLAGS (TA_FLAG_MULTI_SESSION | TA_FLAG_EXEC_DDR)`
- `#define TA_STACK_SIZE (48 * 1024)`
- `#define TA_DATA_SIZE (64 * 1024)`
- `#define TA_CURRENT_TA_EXT_PROPERTIES`

7.7.1 Macro Definition Documentation

7.7.1.1 TA_CURRENT_TA_EXT_PROPERTIES `#define TA_CURRENT_TA_EXT_PROPERTIES`

Value:

```
{ "gp.ta.description", USER_TA_PROP_TYPE_STRING, \
  "AIST OTrP Test TA" }, \
{ "gp.ta.version", USER_TA_PROP_TYPE_U32, &(const uint32_t){ 0x0010 } }
```

7.7.1.2 TA_DATA_SIZE `#define TA_DATA_SIZE (64 * 1024)`

7.7.1.3 TA_FLAGS `#define TA_FLAGS (TA_FLAG_MULTI_SESSION | TA_FLAG_EXEC_DDR)`

7.7.1.4 TA_STACK_SIZE `#define TA_STACK_SIZE (48 * 1024)`

7.7.1.5 TA_UUID `#define TA_UUID`

Value:

```
{ 0x8d82573a, 0x926d, 0x4754, \
  { 0x93, 0x53, 0x32, 0xdc, 0x29, 0x99, 0x7f, 0x74 } }
```

7.8 teep-device/teep-agent-ta/user_ta_header_defines.h File Reference

Macros

- #define [TA_UUID](#)
- #define [TA_FLAGS](#) (TA_FLAG_MULTI_SESSION | TA_FLAG_EXEC_DDR)
- #define [TA_STACK_SIZE](#) (48 * 1024)
- #define [TA_DATA_SIZE](#) (4 * 1024 * 1024)
- #define [TA_CURRENT_TA_EXT_PROPERTIES](#)

7.8.1 Macro Definition Documentation

7.8.1.1 [TA_CURRENT_TA_EXT_PROPERTIES](#) #define TA_CURRENT_TA_EXT_PROPERTIES

Value:

```
{ "gp.ta.description", USER_TA_PROP_TYPE_STRING, \
  "AIST OTrP TA" }, \
{ "gp.ta.version", USER_TA_PROP_TYPE_U32, &(const uint32_t){ 0x0010 } }
```

7.8.1.2 [TA_DATA_SIZE](#) #define TA_DATA_SIZE (4 * 1024 * 1024)

7.8.1.3 [TA_FLAGS](#) #define TA_FLAGS (TA_FLAG_MULTI_SESSION | TA_FLAG_EXEC_DDR)

7.8.1.4 [TA_STACK_SIZE](#) #define TA_STACK_SIZE (48 * 1024)

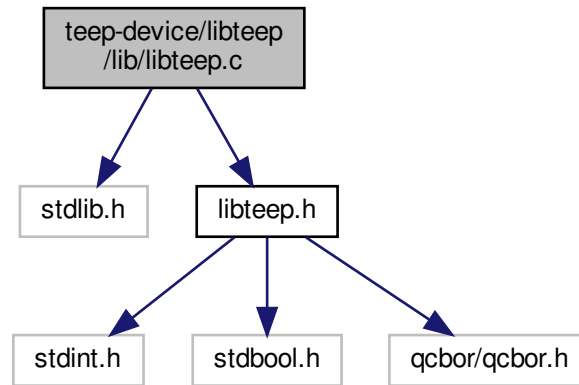
7.8.1.5 [TA_UUID](#) #define TA_UUID

Value:

```
{ 0x68373894, 0x5bb3, 0x403c, \
  { 0x9e, 0xec, 0x31, 0x14, 0xa1, 0xf5, 0xd3, 0xfc } }
```

7.9 teep-device/libteep/lib/libteep.c File Reference

```
#include <stdlib.h>
#include "libteep.h"
Include dependency graph for libteep.c:
```



Functions

- static int [parse_uint32_array](#) (struct [teep_uint32_array](#) *p, QCBORDecodeContext *DC, const QCBORItem *option)
- static int [parse_uint32_option](#) (struct [teep_uint32_option](#) *p, QCBORDecodeContext *DC, const QCBORItem *option)
- static int [parse_buffer_array](#) (struct [teep_buffer_array](#) *p, QCBORDecodeContext *DC, const QCBORItem *option, bool binary)
- static int [parse_tc_info_array](#) (struct [teep_tc_info_array](#) *p, QCBORDecodeContext *DC, const QCBORItem *option, bool requested)
- static int [parse_option](#) (struct [teep_message](#) *m, QCBORDecodeContext *DC, QCBORItem *option, uint8_t nest_level)
- static int [parse_options](#) (struct [teep_message](#) *m, QCBORDecodeContext *DC)
- struct [teep_message](#) * [parse_teep_message](#) (UsefulBufC cbor)
- void [free_parsed_teep_message](#) (struct [teep_message](#) *message)
- void [teep_message_encoder_init](#) (struct [teep_message_encoder](#) *encoder, UsefulBufC buffer)
- void [teep_message_encoder_add_header](#) (struct [teep_message_encoder](#) *encoder, enum [teep_message_type](#) type, uint64_t token)
- void [teep_message_encoder_open_options](#) (struct [teep_message_encoder](#) *encoder)
- void [teep_message_encoder_open_ta_list](#) (struct [teep_message_encoder](#) *encoder)
- void [teep_message_encoder_add_ta_to_ta_list](#) (struct [teep_message_encoder](#) *encoder, const char *ta)
- void [teep_message_encoder_close_ta_list](#) (struct [teep_message_encoder](#) *encoder)
- void [teep_message_encoder_close_options](#) (struct [teep_message_encoder](#) *encoder)
- void [teep_message_encoder_add_err_code](#) (struct [teep_message_encoder](#) *encoder, uint64_t err_code)
- QCBORError [teep_message_encoder_finish](#) (struct [teep_message_encoder](#) *encoder, UsefulBufC *encoded)

7.9.1 Function Documentation

7.9.1.1 free_parsed_teep_message() `void free_parsed_teep_message (`
`struct teep_message * message)`

7.9.1.2 parse_buffer_array() `static int parse_buffer_array (`
`struct teep_buffer_array * p,`
`QCBORDecodeContext * DC,`
`const QCBORItem * option,`
`bool binary) [static]`

7.9.1.3 parse_option() `static int parse_option (`
`struct teep_message * m,`
`QCBORDecodeContext * DC,`
`QCBORItem * option,`
`uint8_t nest_level) [static]`

7.9.1.4 parse_options() `static int parse_options (`
`struct teep_message * m,`
`QCBORDecodeContext * DC) [static]`

7.9.1.5 parse_tc_info_array() `static int parse_tc_info_array (`
`struct teep_tc_info_array * p,`
`QCBORDecodeContext * DC,`
`const QCBORItem * option,`
`bool requested) [static]`

7.9.1.6 parse_teep_message() `struct teep_message* parse_teep_message (`
`UsefulBufC cbor)`

7.9.1.7 parse_uint32_array() `static int parse_uint32_array (`
`struct teep_uint32_array * p,`
`QCBORDecodeContext * DC,`
`const QCBORItem * option) [static]`

7.9.1.8 parse_uint32_option() static int parse_uint32_option (
 struct teep_uint32_option * p,
 QCBORDecodeContext * DC,
 const QCBORItem * option) [static]

7.9.1.9 teep_message_encoder_add_err_code() void teep_message_encoder_add_err_code (
 struct teep_message_encoder * encoder,
 uint64_t err_code)

7.9.1.10 teep_message_encoder_add_header() void teep_message_encoder_add_header (
 struct teep_message_encoder * encoder,
 enum teep_message_type type,
 uint64_t token)

7.9.1.11 teep_message_encoder_add_ta_to_ta_list() void teep_message_encoder_add_ta_to_ta_list (
 struct teep_message_encoder * encoder,
 const char * ta)

7.9.1.12 teep_message_encoder_close_options() void teep_message_encoder_close_options (
 struct teep_message_encoder * encoder)

7.9.1.13 teep_message_encoder_close_ta_list() void teep_message_encoder_close_ta_list (
 struct teep_message_encoder * encoder)

7.9.1.14 teep_message_encoder_finish() QCBORError teep_message_encoder_finish (
 struct teep_message_encoder * encoder,
 UsefulBufC * encoded)

7.9.1.15 teep_message_encoder_init() void teep_message_encoder_init (
 struct teep_message_encoder * encoder,
 UsefulBufC buffer)

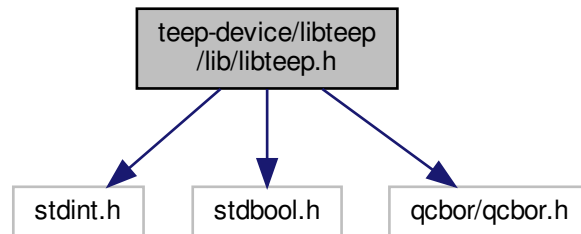
7.9.1.16 teep_message_encoder_open_options() void teep_message_encoder_open_options (struct teep_message_encoder * encoder)

7.9.1.17 teep_message_encoder_open_ta_list() void teep_message_encoder_open_ta_list (struct teep_message_encoder * encoder)

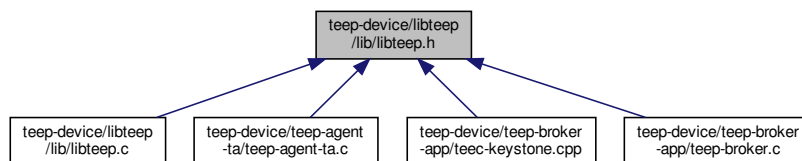
7.10 teep-device/libteep/lib/libteep.h File Reference

```
#include <stdint.h>
#include <stdbool.h>
#include <qcbor/qcbor.h>
```

Include dependency graph for libteep.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [teep_uint32_array](#)
- struct [teep_uint32_option](#)
- struct [teep_buffer_array](#)
- struct [teep_tc_info](#)
- struct [teep_tc_info_array](#)
- struct [teep_message](#)
- struct [teep_message_encoder](#)

Enumerations

- enum `teep_message_type` {
`TEEP_QUERY_REQUEST` = 1 , `TEEP_QUERY_RESPONSE` = 2 , `TEEP_INSTALL` = 3 , `TEEP_DELETE` = 4 ,
`TEEP_SUCCESS` = 5 , `TEEP_ERROR` = 6 }
- enum `teep_data_item` { `TEEP_DATA_ATTESTATION` = 1 , `TEEP_DATA_TRUSTED_COMPONENTS` = 2 ,
`TEEP_DATA_EXTENSIONS` = 4 , `TEEP_DATA_SUIT_COMMANDS` = 8 }
- enum `teep_suite` { `TEEP_AES_CCM_16_64_128_HMAC256_256_X25519_EdDSA` = 1 , `TEEP_AES_CCM_16_64_128_HMAC256_256_X25519_EdDSA` = 2 }
- enum `teep_error` {
`TEEP_ERR_ILLEGAL_PARAMETER` = 1 , `TEEP_ERR_UNSUPPORTED_EXTENSION` = 2 , `TEEP_ERR_REQUEST_SIGNATURE` = 3 ,
`TEEP_ERR_UNSUPPORTED_MSG_VERSION` = 4 ,
`TEEP_ERR_UNSUPPORTED_CRYPTO_ALG` = 5 , `TEEP_ERR_BAD_CERTIFICATE` = 6 , `TEEP_ERR_UNSUPPORTED_CERTIFICATE` = 7 ,
`TEEP_ERR_CERTIFICATE_REVOKED` = 8 ,
`TEEP_ERR_CERTIFICATE_EXPIRED` = 9 , `TEEP_ERR_INTERNAL_ERROR` = 10 , `TEEP_ERR_TC_NOT_FOUND` = 12 ,
`TEEP_ERR_MANIFEST_PROCESSING_FAILED` = 17 }
- enum `teep_option_key` {
`TEEP_OPTION_SUPPORTED_CIPHER_SUITS` = 1 , `TEEP_OPTION_CHALLENGE` = 2 , `TEEP_OPTION_VERSIONS` = 3 ,
`TEEP_OPTION_OCSP_DATA` = 4 ,
`TEEP_OPTION_SELECTED_CIPHER_SUIT` = 5 , `TEEP_OPTION_SELECTED_VERSION` = 6 , `TEEP_OPTION_EVIDENCE` = 7 ,
`TEEP_OPTION_TC_LIST` = 8 ,
`TEEP_OPTION_EXT_LIST` = 9 , `TEEP_OPTION_MANIFEST_LIST` = 10 , `TEEP_OPTION_MSG` = 11 ,
`TEEP_OPTION_ERR_MSG` = 12 ,
`TEEP_OPTION_EVIDENCE_FORMAT` = 13 , `TEEP_OPTION_REQUESTED_TC_LIST` = 14 , `TEEP_OPTION_UNNEEDED_TC_LIST` = 15 ,
`TEEP_OPTION_COMPONENT_ID` = 16 ,
`TEEP_OPTION_TC_MANIFEST_SEQUENCE_NUMBER` = 17 , `TEEP_OPTION_HAVE_BINARY` = 18 ,
`TEEP_OPTION_SUIT_REPORTS` = 19 }

Functions

- struct `teep_message` * `parse_teeep_message` (UsefulBufC cbor)
- void `free_parsed_teeep_message` (struct `teep_message` *message)
- void `teep_message_encoder_init` (struct `teep_message_encoder` *encoder, UsefulBuf buffer)
- void `teep_message_encoder_add_header` (struct `teep_message_encoder` *encoder, enum `teep_message_type` type, uint64_t token)
- void `teep_message_encoder_open_options` (struct `teep_message_encoder` *encoder)
- void `teep_message_encoder_open_ta_list` (struct `teep_message_encoder` *encoder)
- void `teep_message_encoder_add_ta_to_ta_list` (struct `teep_message_encoder` *encoder, const char *ta)
- void `teep_message_encoder_close_ta_list` (struct `teep_message_encoder` *encoder)
- void `teep_message_encoder_close_options` (struct `teep_message_encoder` *encoder)
- void `teep_message_encoder_add_err_code` (struct `teep_message_encoder` *encoder, uint64_t err_code)
- QCBORError `teep_message_encoder_finish` (struct `teep_message_encoder` *encoder, UsefulBufC *encoded)

7.10.1 Enumeration Type Documentation

7.10.1.1 `teep_data_item` enum `teep_data_item`

Enumerator

<code>TEEP_DATA_ATTESTATION</code>	
<code>TEEP_DATA_TRUSTED_COMPONENTS</code>	
<code>TEEP_DATA_EXTENSIONS</code>	
<code>TEEP_DATA_SUIT_COMMANDS</code>	

7.10.1.2 teep.error enum `teep.error`

Enumerator

TEEP_ERR_ILLEGAL_PARAMETER	
TEEP_ERR_UNSUPPORTED_EXTENSION	
TEEP_ERR_REQUEST_SIGNATURE_FAILED	
TEEP_ERR_UNSUPPORTED_MSG_VERSION	
TEEP_ERR_UNSUPPORTED_CRYPTO_ALG	
TEEP_ERR_BAD_CERTIFICATE	
TEEP_ERR_UNSUPPORTED_CERTIFICATE	
TEEP_ERR_CERTIFICATE_REVOKED	
TEEP_ERR_CERTIFICATE_EXPIRED	
TEEP_ERR_INTERNAL_ERROR	
TEEP_ERR_TC_NOT_FOUND	
TEEP_ERR_MANIFEST_PROCESSING_FAILED	

7.10.1.3 teep.message.type enum `teep.message.type`

Enumerator

TEEP_QUERY_REQUEST	
TEEP_QUERY_RESPONSE	
TEEP_INSTALL	
TEEP_DELETE	
TEEP_SUCCESS	
TEEP_ERROR	

7.10.1.4 teep.option.key enum `teep.option.key`

Enumerator

TEEP_OPTION_SUPPORTED_CIPHER_SUITS	
TEEP_OPTION_CHALLENGE	
TEEP_OPTION_VERSIONS	
TEEP_OPTION_OCSP_DATA	
TEEP_OPTION_SELECTED_CIPHER_SUIT	
TEEP_OPTION_SELECTED_VERSION	
TEEP_OPTION_EVIDENCE	
TEEP_OPTION_TC_LIST	
TEEP_OPTION_EXT_LIST	
TEEP_OPTION_MANIFEST_LIST	
Enum paramter list continued on next page	

TEEP_OPTION_MSG	
TEEP_OPTION_ERR_MSG	
TEEP_OPTION_EVIDENCE_FORMAT	
TEEP_OPTION_REQUESTED_TC_LIST	
TEEP_OPTION_UNNEEDED_TC_LIST	
TEEP_OPTION_COMPONENT_ID	
TEEP_OPTION_TC_MANIFEST_SEQUENCE_NUMBER	
TEEP_OPTION_HAVE_BINARY	
TEEP_OPTION_SUIT_REPORTS	

7.10.1.5 teep_suite enum teep_suite

Enumerator

TEEP_AES_CCM_16_64_128_HMAC256_256_X25519_EdDSA	
TEEP_AES_CCM_16_64_128_HMAC256_256_P_256_ES256	

7.10.2 Function Documentation

7.10.2.1 free_parsed_teep_message() void free_parsed_teep_message (struct teep_message * message)

7.10.2.2 parse_teep_message() struct teep_message* parse_teep_message (UsefulBufC cbor)

7.10.2.3 teep_message_encoder_add_err_code() void teep_message_encoder_add_err_code (struct teep_message_encoder * encoder, uint64_t err_code)

7.10.2.4 teep_message_encoder_add_header() void teep_message_encoder_add_header (struct teep_message_encoder * encoder, enum teep_message_type type, uint64_t token)

7.10.2.5 teep_message_encoder_add_ta_to_ta_list() void teep_message_encoder_add_ta_to_ta_list (struct teep_message_encoder * encoder, const char * ta)

7.10.2.6 teep_message_encoder_close_options() void teep_message_encoder_close_options (struct teep_message_encoder * encoder)

7.10.2.7 teep_message_encoder_close_ta_list() void teep_message_encoder_close_ta_list (struct teep_message_encoder * encoder)

7.10.2.8 teep_message_encoder_finish() QCBORError teep_message_encoder_finish (struct teep_message_encoder * encoder, UsefulBufC * encoded)

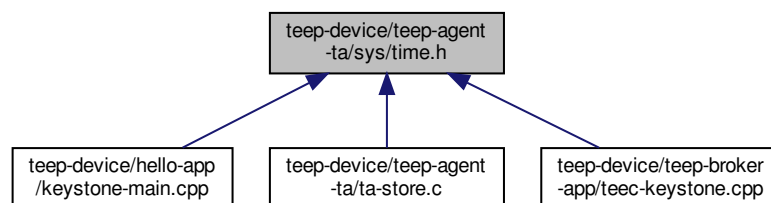
7.10.2.9 teep_message_encoder_init() void teep_message_encoder_init (struct teep_message_encoder * encoder, UsefulBufC buffer)

7.10.2.10 teep_message_encoder_open_options() void teep_message_encoder_open_options (struct teep_message_encoder * encoder)

7.10.2.11 teep_message_encoder_open_ta_list() void teep_message_encoder_open_ta_list (struct teep_message_encoder * encoder)

7.11 teep-device/teep-agent-ta/sys/time.h File Reference

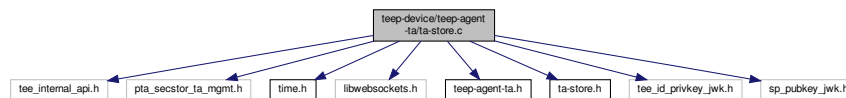
This graph shows which files directly or indirectly include this file:



7.12 teep-device/teep-agent-ta/ta-store.c File Reference

```
#include <tee_internal_api.h>
#include <pta_secstor_ta_mgmt.h>
#include <time.h>
#include <libwebsockets.h>
#include "teep-agent-ta.h"
#include "ta-store.h"
#include "tee_id_privkey_jwk.h"
#include "sp_pubkey_jwk.h"
```

Include dependency graph for ta-store.c:



Macros

- `#define TEMP_BUF_SIZE (800 * 1024)`

Functions

- `int hex (char c)`
- `int string_to_uuid_octets (const char *s, uint8_t *octets16)`
- `static struct lws_context * get_lws_context ()`
- `static int teep_message_unwrap_ta_image (const char *msg, int msg_len, char *out, uint32_t *out_len)`
- `int ta_store_install (const char *ta_image_ciphertext, size_t ta_image_ciphertext_len, const char *ta_name, size_t ta_name_len)`
- `int ta_store_delete (const char *uuid_string, size_t uuid_string_len)`

Variables

- `static char temp_buf [TEMP_BUF_SIZE]`
- `static const char *const tee_id_privkey_jwk =`
- `static const char *const sp_pubkey_jwk =`
- `static ta_image_buf [TEMP_BUF_SIZE]`

7.12.1 Macro Definition Documentation

7.12.1.1 TEMP_BUF_SIZE `#define TEMP_BUF_SIZE (800 * 1024)`

7.12.2 Function Documentation

7.12.2.1 get_lws_context() `static struct lws_context* get_lws_context () [static]`

7.12.2.2 hex() `int hex (`
`char c)`

7.12.2.3 string_to_uuid_octets() `int string_to_uuid_octets (`
`const char * s,`
`uint8_t * octets16)`

7.12.2.4 ta_store_delete() `int ta_store_delete (`
`const char * uuid_string,`
`size_t uuid_string_len)`

[ta_store_delete\(\)](#) - Deletes a TA Image corresponds to UUID from secure storage using optee pta.

Parameters

<i>uuid_string</i>	uuid_string is a type of the constant character.
<i>uuid_string_len</i>	uuid_string_len is a type of the unsigned integer data type.

Returns

0 if success else, error occurred.

7.12.2.5 ta_store_install() `int ta_store_install (`
`const char * ta_image_ciphertext,`
`size_t ta_image_ciphertext_len,`
`const char * ta_name,`
`size_t ta_name_len)`

[ta_store_install\(\)](#) - Installs the given TA Image into secure storage using optee pta.

If defined value is PCTEST then it will send libwebsockets notification like "stub called ta_image_len" with ta image length. If defined value is PLAT_KEYSTONE then it will send libwebsockets notification like "ta image length" and "ta name" and then invokes the install plain(), storage sector() and storage sector plain(). If it is not defined anything then it will open ta session in tee and invoke the command and then finally it will close the ta session. If [ta_store_install\(\)](#) function success then it will send notification like "Wrote TA to secure storage".

Parameters

<i>ta_image</i>	ta_image is a type of the constant character.
<i>ta_image_len</i>	ta_image_len is a type of the unsigned integer data type.
<i>ta_name</i>	ta_name is a type of the constant character.
<i>ta_name_len</i>	ta_name_len is a type of the unsigned integer data type.

Returns

0 if success else error occurred.

7.12.2.6 teep_message_unwrap_ta_image() `static int teep_message_unwrap_ta_image (`
 `const char * msg,`
 `int msg_len,`
 `char * out,`
 `uint32_t * out_len) [static]`

7.12.3 Variable Documentation

7.12.3.1 sp_pubkey_jwk `const char* const sp_pubkey_jwk = [static]`

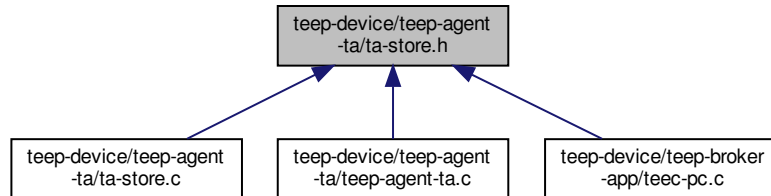
7.12.3.2 ta_image_buf `ta_image_buf[TEMP_BUF_SIZE] [static]`

7.12.3.3 tee_id_privkey_jwk `const char* const tee_id_privkey_jwk = [static]`

7.12.3.4 temp_buf `char temp_buf[TEMP_BUF_SIZE] [static]`

7.13 teep-device/teep-agent-ta/ta-store.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int [ta_store_install](#) (const char *ta_image, size_t ta_image_len, const char *ta_name, size_t ta_name_len)
- int [ta_store_delete](#) (const char *uuid_string, size_t uuid_string_len)

7.13.1 Function Documentation

7.13.1.1 ta_store_delete() int ta_store_delete (

```

    const char * uuid_string,
    size_t uuid_string_len )

```

[ta_store_delete\(\)](#) - Deletes a TA Image corresponds to UUID from secure storage using optee pta.

Parameters

<i>uuid_string</i>	uuid_string is a type of the constant character.
<i>uuid_string_len</i>	uuid_string_len is a type of the unsigned integer data type.

Returns

0 if success else, error occurred.

7.13.1.2 ta_store_install() int ta_store_install (

```

    const char * ta_image_ciphertext,
    size_t ta_image_ciphertext_len,
    const char * ta_name,
    size_t ta_name_len )

```

[ta_store_install\(\)](#) - Installs the given TA Image into secure storage using optee pta.

If defined value is PCTEST then it will send libwebsockets notification like "stub called ta_image_len" with ta image length. If defined value is PLAT_KEYSTONE then it will send libwebsockets notification like "ta image length" and "ta name" and then invokes the install plain(), storage sector() and storage sector plain(). If it is not defined anything then it will open ta session in tee and invoke the command and then finally it will close the ta session. If [ta_store_install\(\)](#) function success then it will send notification like "Wrote TA to secure storage".

Parameters

<i>ta_image</i>	ta_image is a type of the constant character.
<i>ta_image_len</i>	ta_image_len is a type of the unsigned integer data type.
<i>ta_name</i>	ta_name is a type of the constant character.
<i>ta_name_len</i>	ta_name_len is a type of the unsigned integer data type.

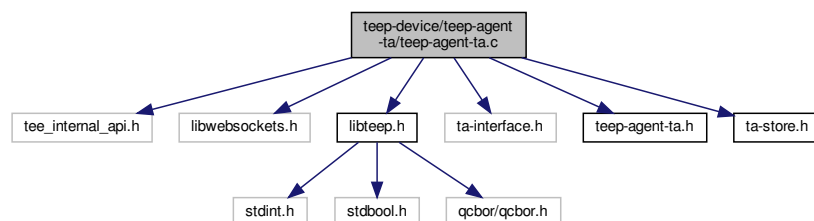
Returns

0 if success else error occurred.

7.14 teep-device/teep-agent-ta/teep-agent-ta.c File Reference

```
#include <tee_internal_api.h>
#include <libwebsockets.h>
#include <libteep.h>
#include "ta-interface.h"
#include "teep-agent-ta.h"
#include "ta-store.h"
```

Include dependency graph for teep-agent-ta.c:



Classes

- struct [ta_manifest](#)
- struct [teep_agent_session](#)

Enumerations

- enum [agent_state](#) {
[AGENT_INIT](#) , [AGENT_POSTING_INITIAL_REQUEST](#) , [AGENT_POSTING_QUERY_RESPONSE](#) ,
[AGENT_POSTING_SUCCESS](#) ,
[AGENT_POSTING_ERROR](#) , [AGENT_DOWNLOAD_TA](#) , [AGENT_FINISH](#) }

Functions

- static struct `teep_agent_session` * `teep_agent_session_create` ()
- static void `teep_agent_session_destroy` (struct `teep_agent_session` *`session`)
- static TEE_Result `set_dev_option` (struct `teep_agent_session` *`session`, enum `agent_dev_option` `option`, const char *`value`)
- static void `teep_error` (struct `teep_agent_session` *`session`, const char *`message`)
- static int `set_manifest_from_uri` (struct `ta_manifest` *`manifest`, UsefulBufC `src_uri`)
- static int `set_manifest_from_suit_install` (struct `ta_manifest` *`manifest`, UsefulBufC `suit_install`)
- static int `set_manifest_from_suit_manifest` (struct `ta_manifest` *`manifest`, UsefulBufC `suit_manifest`)
- static int `set_manifest_from_suit` (struct `ta_manifest` *`manifest`, UsefulBufC `suit_envelope`)
- static void `handle_tam_message` (struct `teep_agent_session` *`session`, const void *`buffer`, size_t `len`)
- static void `handle_ta_download` (struct `teep_agent_session` *`session`, const void *`buffer`, size_t `len`, const char *`uri`)
- static TEE_Result `broker_task_done` (struct `teep_agent_session` *`session`, const void *`buffer`, size_t `len`)
- static int `build_query_response` (struct `teep_agent_session` *`session`, void *`dst`, size_t *`dst_len`)
- static int `build_success` (struct `teep_agent_session` *`session`, void *`dst`, size_t *`dst_len`)
- static int `build_error` (struct `teep_agent_session` *`session`, void *`dst`, size_t *`dst_len`)
- static struct broker_task * `query_next_broker_task` (struct `teep_agent_session` *`session`)
- TEE_Result `TA_CreateEntryPoint` (void)
- void `TA_DestroyEntryPoint` (void)
- TEE_Result `TA_OpenSessionEntryPoint` (uint32_t `param_types`, TEE_Param `params`[4], void **`sess_ctx`)
- void `TA_CloseSessionEntryPoint` (void *`sess_ctx`)
- static TEE_Result `handle_TEEP_AGENT_SET_DEV_OPTION` (struct `teep_agent_session` *`session`, uint32_t `param_types`, TEE_Param `params`[TEE_NUM_PARAMS])
- static TEE_Result `handle_TEEP_AGENT_BROKER_TASK_DONE` (struct `teep_agent_session` *`session`, uint32_t `param_types`, TEE_Param `params`[TEE_NUM_PARAMS])
- static int `copyout_param` (TEE_Param *`param`, const void *`buffer`, size_t `len`)
- static TEE_Result `handle_TEEP_AGENT_QUERY_NEXT_BROKER_TASK` (struct `teep_agent_session` *`session`, uint32_t `param_types`, TEE_Param `params`[TEE_NUM_PARAMS])
- TEE_Result `TA_InvokeCommandEntryPoint` (void *`sess_ctx`, uint32_t `cmd_id`, uint32_t `param_types`, TEE_Param `params`[TEE_NUM_PARAMS])

7.14.1 Enumeration Type Documentation

7.14.1.1 agent_state enum agent_state

Enumerator

AGENT_INIT	
AGENT_POSTING_INITIAL_REQUEST	
AGENT_POSTING_QUERY_RESPONSE	
AGENT_POSTING_SUCCESS	
AGENT_POSTING_ERROR	
AGENT_DOWNLOAD_TA	
AGENT_FINISH	

7.14.2 Function Documentation

7.14.2.1 broker_task_done() static TEE_Result broker_task_done (
 struct `teep_agent_session` * *session*,
 const void * *buffer*,
 size_t *len*) [static]

7.14.2.2 build_error() static int build_error (
 struct `teep_agent_session` * *session*,
 void * *dst*,
 size_t * *dst_len*) [static]

7.14.2.3 build_query_response() static int build_query_response (
 struct `teep_agent_session` * *session*,
 void * *dst*,
 size_t * *dst_len*) [static]

7.14.2.4 build_success() static int build_success (
 struct `teep_agent_session` * *session*,
 void * *dst*,
 size_t * *dst_len*) [static]

7.14.2.5 copyout_param() static int copyout_param (
 TEE_Param * *param*,
 const void * *buffer*,
 size_t *len*) [static]

7.14.2.6 handle_ta_download() static void handle_ta_download (
 struct `teep_agent_session` * *session*,
 const void * *buffer*,
 size_t *len*,
 const char * *uri*) [static]

7.14.2.7 handle_tam_message() static void handle_tam_message (

```
    struct teep_agent_session * session,
    const void * buffer,
    size_t len ) [static]
```

7.14.2.8 handle_TEEP_AGENT_BROKER_TASK_DONE() static TEE_Result handle_TEEP_AGENT_BROKER_↵
TASK_DONE (

```
    struct teep_agent_session * session,
    uint32_t param_types,
    TEE_Param params[TEE_NUM_PARAMS] ) [static]
```

7.14.2.9 handle_TEEP_AGENT_QUERY_NEXT_BROKER_TASK() static TEE_Result handle_TEEP_AGENT_↵
QUERY_NEXT_BROKER_TASK (

```
    struct teep_agent_session * session,
    uint32_t param_types,
    TEE_Param params[TEE_NUM_PARAMS] ) [static]
```

7.14.2.10 handle_TEEP_AGENT_SET_DEV_OPTION() static TEE_Result handle_TEEP_AGENT_SET_DEV_↵
OPTION (

```
    struct teep_agent_session * session,
    uint32_t param_types,
    TEE_Param params[TEE_NUM_PARAMS] ) [static]
```

7.14.2.11 query_next_broker_task() static struct broker_task* query_next_broker_task (

```
    struct teep_agent_session * session ) [static]
```

7.14.2.12 set_dev_option() static TEE_Result set_dev_option (

```
    struct teep_agent_session * session,
    enum agent_dev_option option,
    const char * value ) [static]
```

7.14.2.13 set_manifest_from_suit() static int set_manifest_from_suit (

```
    struct ta_manifest * manifest,
    UsefulBufC suit_envelope ) [static]
```

7.14.2.14 set_manifest_from_suit_install() static int set_manifest_from_suit_install (
 struct [ta_manifest](#) * manifest,
 UsefulBufC suit_install) [static]

7.14.2.15 set_manifest_from_suit_manifest() static int set_manifest_from_suit_manifest (
 struct [ta_manifest](#) * manifest,
 UsefulBufC suit_manifest) [static]

7.14.2.16 set_manifest_from_uri() static int set_manifest_from_uri (
 struct [ta_manifest](#) * manifest,
 UsefulBufC src_uri) [static]

7.14.2.17 TA_CloseSessionEntryPoint() void TA_CloseSessionEntryPoint (
 void * sess_ctx)

[TA_CloseSessionEntryPoint\(\)](#) - The Framework calls the function to close a client session.

The Trusted Application implementation is responsible for freeing any resources consumed by the session.

Parameters

sess_ctx	The value of the void opaque data pointer set by the Trusted Application in the function TA_OpenSessionEntryPoint for this session.
------------------------------------	---

Returns

TEE_SUCCESS for success, else any other value.

7.14.2.18 TA_CreateEntryPoint() TEE_Result TA_CreateEntryPoint (
 void)

[TA_CreateEntryPoint\(\)](#) - Creates the entry point for TA.

The function set the log level for TA.

Returns

TEE_SUCCESS for success, else any other value.

7.14.2.19 TA_DestroyEntryPoint() `void TA_DestroyEntryPoint (`
`void)`

[TA_DestroyEntryPoint\(\)](#) - Destroys the entry point for TA.

7.14.2.20 TA_InvokeCommandEntryPoint() `TEE_Result TA_InvokeCommandEntryPoint (`
`void * sess_ctx,`
`uint32_t cmd_id,`
`uint32_t param_types,`
`TEE_Param params[TEE_NUM_PARAMS])`

[TA_InvokeCommandEntryPoint\(\)](#) - Invokes the command entry point for TA.

Based on command id, verify the tee parameter type and invokes `teep_message_wrap()`, `teep_message_unwrap()`, `otrp_message_verify()`, [ta_store_install\(\)](#), [ta_store_delete\(\)](#), [teep_message_unwrap_ta_image\(\)](#), `otrp_message_sign`, `otrp_message_encrypt()`, and `otrp_message_decrypt()`.

Parameters

<i>sess_ctx</i>	The value of the void opaque data pointer set by the Trusted Application in the function <code>TA_OpenSessionEntryPoint</code> .
<i>cmd_id</i>	A Trusted Application-specific code that identifies the command to be invoked.
<i>param_types</i>	The types of the four parameters
<i>params[TEE_NUM_PARAMS]</i>	A pointer to an array of four parameters

Returns

TEE_SUCCESS for success, else any other value.

7.14.2.21 TA_OpenSessionEntryPoint() `TEE_Result TA_OpenSessionEntryPoint (`
`uint32_t param_types,`
`TEE_Param params[4],`
`void ** sess_ctx)`

[TA_OpenSessionEntryPoint\(\)](#) - Opens the session entry point for TA.

The Framework calls the function `TA_OpenSessionEntryPoint` when a client requests to open a session with the Trusted Application.

Parameters

<i>param_types</i>	<i>param_types</i> is a numeric type that guarantees 32 bits.
<i>params[]</i>	A pointer to an array of four parameters
<i>**sess_ctx</i>	A pointer to a variable that can be filled by the Trusted Application instance with an opaque void* data pointer

Returns

TEE_SUCCESS If the session is successfully opened, else any other value.

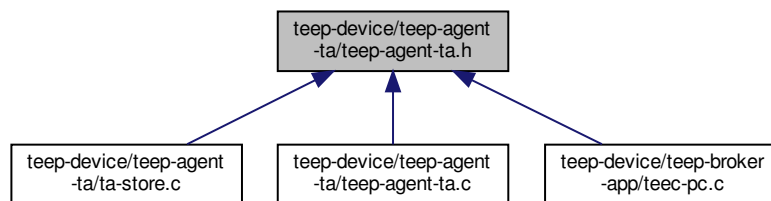
7.14.2.22 teep_agent_session_create() static struct [teep_agent_session](#)* teep_agent_session_create () [static]

7.14.2.23 teep_agent_session_destroy() static void teep_agent_session_destroy (struct [teep_agent_session](#) * session) [static]

7.14.2.24 teep_error() static void [teep_error](#) (struct [teep_agent_session](#) * session, const char * message) [static]

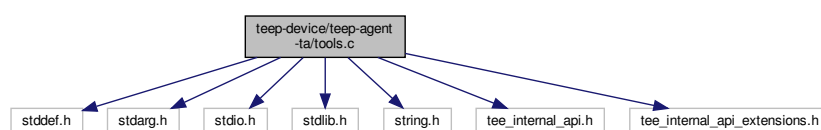
7.15 teep-device/teep-agent-ta/teep-agent-ta.h File Reference

This graph shows which files directly or indirectly include this file:

**7.16 teep-device/teep-agent-ta/tools.c File Reference**

```
#include <stddef.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <tee_internal_api.h>
#include <tee_internal_api_extensions.h>
```

Include dependency graph for tools.c:



Functions

- static unsigned int `_strlen` (const char *str)
- char * `strcpy` (char *dst, const char *src)
- char * `strncpy` (char *dst, const char *src, size_t n)
- char * `strncat` (char *dest, const char *src, size_t n)
- char * `strdup` (const char *s)
- char * `strchr` (const char *s, int c)
- char * `strchr` (const char *s, int c)
- int `atoi` (const char *s)
- time_t `time` (time_t *tloc)
- int `gettimeofday` (struct timeval *tv, struct timezone *tz)

7.16.1 Function Documentation

7.16.1.1 `_strlen()` static unsigned int `_strlen` (
const char * *str*) [inline], [static]

`_strlen()` - Computes the length of the string.

This function used for loop to parse the string and return length of the string as an unsigned interger format.

Parameters

<i>str</i>	string whose length is to be found.
------------	-------------------------------------

7.16.1.2 `atoi()` int `atoi` (
const char * *s*)

7.16.1.3 `gettimeofday()` int `gettimeofday` (
struct timeval * *tv*,
struct timezone * *tz*)

7.16.1.4 `strchr()` char* `strchr` (
const char * *s*,
int *c*)

`strchr()` - Searches for the first occurrence of the character *c* (an unsigned char) in the string pointed to by the argument *str*. And it just returns with an another function called `__builtin_strchr(s, c)`.

Parameters

<i>s</i>	The string to be scanned.
<i>c</i>	The character to be searched in str format.

7.16.1.5 strcpy() `char* strcpy (`
 `char * dst,`
 `const char * src)`

[strcpy\(\)](#) - Copies the string from source to destination.

This function iniializes the destination "dst" variable to copy the string using the while loop.

Parameters

<i>dst</i>	A pointer to the destination array where the content is to be copied.
<i>src</i>	The string to be copied.

Returns

It returns a pointer to the destination string dest.

7.16.1.6 strdup() `char* strdup (`
 `const char * s)`

7.16.1.7 strncat() `char* strncat (`
 `char * dest,`
 `const char * src,`
 `size_t n)`

7.16.1.8 strncpy() `char* strncpy (`
 `char * dst,`
 `const char * src,`
 `size_t n)`

[strncpy\(\)](#) - Copies up to n characters from the string pointed from source to destination.

This fucntion begins with a loop and assigns destination dst value to source src value. It runs an another loop that returns the destination dest value.

Parameters

<i>dst</i>	A pointer to the destination array
<i>src</i>	The string to be copied.
<i>n</i>	The number of characters to be copied from source.

Returns

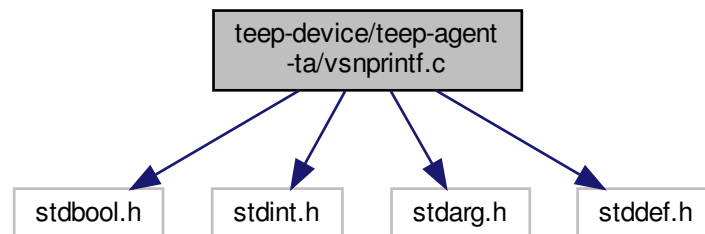
It returns the final copy of the copied string.

7.16.1.9 strrchr() `char* strrchr (`
 `const char * s,`
 `int c)`

7.16.1.10 time() `time_t time (`
 `time_t * tloc)`

7.17 teep-device/teep-agent-ta/vsnprintf.c File Reference

```
#include <stdbool.h>
#include <stdint.h>
#include <stdarg.h>
#include <stddef.h>
Include dependency graph for vsnprintf.c:
```

**Classes**

- struct [out_fct_wrap_type](#)

Macros

- #define `PRINTF_NTOA_BUFFER_SIZE` 32U
- #define `PRINTF_FTOA_BUFFER_SIZE` 32U
- #define `PRINTF_SUPPORT_FLOAT`
- #define `PRINTF_SUPPORT_LONG_LONG`
- #define `PRINTF_SUPPORT_PTRDIFF_T`
- #define `FLAGS_ZEROPAD` (1U << 0U)
- #define `FLAGS_LEFT` (1U << 1U)
- #define `FLAGS_PLUS` (1U << 2U)
- #define `FLAGS_SPACE` (1U << 3U)
- #define `FLAGS_HASH` (1U << 4U)
- #define `FLAGS_UPPERCASE` (1U << 5U)
- #define `FLAGS_CHAR` (1U << 6U)
- #define `FLAGS_SHORT` (1U << 7U)
- #define `FLAGS_LONG` (1U << 8U)
- #define `FLAGS_LONG_LONG` (1U << 9U)
- #define `FLAGS_PRECISION` (1U << 10U)
- #define `_putchar` putchar

Typedefs

- typedef void(* `out_fct_type`) (char character, void *buffer, size_t idx, size_t maxlen)

Functions

- int `putchar` (char ch)
- static void `_out_buffer` (char character, void *buffer, size_t idx, size_t maxlen)
- static void `_out_null` (char character, void *buffer, size_t idx, size_t maxlen)
- static void `_out_char` (char character, void *buffer, size_t idx, size_t maxlen)
- static void `_out_fct` (char character, void *buffer, size_t idx, size_t maxlen)
- static unsigned int `_strlen` (const char *str)
- static bool `_is_digit` (char ch)
- static unsigned int `_atoi` (const char **str)
- static size_t `_ntoa_format` (`out_fct_type` out, char *buffer, size_t idx, size_t maxlen, char *buf, size_t len, bool negative, unsigned int base, unsigned int prec, unsigned int width, unsigned int flags)
- static size_t `_ntoa_long` (`out_fct_type` out, char *buffer, size_t idx, size_t maxlen, unsigned long value, bool negative, unsigned long base, unsigned int prec, unsigned int width, unsigned int flags)
- static size_t `_ntoa_long_long` (`out_fct_type` out, char *buffer, size_t idx, size_t maxlen, unsigned long long value, bool negative, unsigned long long base, unsigned int prec, unsigned int width, unsigned int flags)
- static size_t `_ftoa` (`out_fct_type` out, char *buffer, size_t idx, size_t maxlen, double value, unsigned int prec, unsigned int width, unsigned int flags)
- static int `_vsnprintf` (`out_fct_type` out, char *buffer, const size_t maxlen, const char *format, va_list va)
- int `sprintf` (char *buffer, const char *format,...)
- int `snprintf` (char *buffer, size_t count, const char *format,...)
- int `vsnprintf` (char *buffer, size_t count, const char *format, va_list va)
- int `fctprintf` (void(*out)(char character, void *arg), void *arg, const char *format,...)

7.17.1 Macro Definition Documentation

7.17.1.1 `_putchar` `#define _putchar putchar`

7.17.1.2 `FLAGS_CHAR` `#define FLAGS_CHAR (1U << 6U)`

7.17.1.3 `FLAGS_HASH` `#define FLAGS_HASH (1U << 4U)`

7.17.1.4 `FLAGS_LEFT` `#define FLAGS_LEFT (1U << 1U)`

7.17.1.5 `FLAGS_LONG` `#define FLAGS_LONG (1U << 8U)`

7.17.1.6 `FLAGS_LONG_LONG` `#define FLAGS_LONG_LONG (1U << 9U)`

7.17.1.7 `FLAGS_PLUS` `#define FLAGS_PLUS (1U << 2U)`

7.17.1.8 `FLAGS_PRECISION` `#define FLAGS_PRECISION (1U << 10U)`

7.17.1.9 `FLAGS_SHORT` `#define FLAGS_SHORT (1U << 7U)`

7.17.1.10 `FLAGS_SPACE` `#define FLAGS_SPACE (1U << 3U)`

7.17.1.11 `FLAGS_UPPERCASE` `#define FLAGS_UPPERCASE (1U << 5U)`

7.17.1.12 FLAGS_ZEROPAD `#define FLAGS_ZEROPAD (1U << 0U)`

7.17.1.13 PRINTF_FTOA_BUFFER_SIZE `#define PRINTF_FTOA_BUFFER_SIZE 32U`

7.17.1.14 PRINTF_NTOA_BUFFER_SIZE `#define PRINTF_NTOA_BUFFER_SIZE 32U`

7.17.1.15 PRINTF_SUPPORT_FLOAT `#define PRINTF_SUPPORT_FLOAT`

7.17.1.16 PRINTF_SUPPORT_LONG_LONG `#define PRINTF_SUPPORT_LONG_LONG`

7.17.1.17 PRINTF_SUPPORT_PTRDIFF_T `#define PRINTF_SUPPORT_PTRDIFF_T`

7.17.2 Typedef Documentation

7.17.2.1 out_fct_type `typedef void(* out_fct_type) (char character, void *buffer, size_t idx, size_t maxlen)`

7.17.3 Function Documentation

7.17.3.1 _atoi() `static unsigned int _atoi (const char ** str) [static]`

[_atoi\(\)](#) - Converting the internal ASCII string into unsigned integer.

This function is to convert internal ASCII string into unsigned integer.

Parameters

<i>str</i>	string representation of an integral number.
------------	--

Returns

i unsigned integer value.

7.17.3.2 `_ftoa()` `static size_t _ftoa (`
`out.fct_type out,`
`char * buffer,`
`size_t idx,`
`size_t maxlen,`
`double value,`
`unsigned int prec,`
`unsigned int width,`
`unsigned int flags) [static]`

`_ftoa()` - Converts a given floating-point number or a double to a string. Use of standard library functions for direct conversion is not allowed.

The ftoa function is used to convert float point into string. firstly it initialize the variables and test case is added to check value is negative or not. set up the default precision to 6, if not set explicitly, its nothing but format specifier, And after limit precision to nine, because a prec greater than or equal to ten can lead to overflow errors. Initialize some variable for precision roll-over, round up also added if it is required to round up the value, For very large numbers switch back to native sprintf for exponentials. Some fractional part adds some extra zeros. if required using for loop string reverse is performed and append pad spaces up to given width.

Parameters

<i>out</i>	type of out.fct_type
<i>buffer</i>	Pointer to a character string to write the result.
<i>idx</i>	idx bytes of size_t
<i>maxlen</i>	Maximum number of characters to write.
<i>negative</i>	boolean type
<i>base</i>	an unsigned long data type
<i>prec</i>	an unsigned integral data type
<i>width</i>	an unsigned integral data type
<i>flags</i>	an unsigned integral data type

Returns

idx It returns interger idx

7.17.3.3 `_is_digit()` `static bool _is_digit (`
`char ch) [inline], [static]`

`_is_digit()` - To check if char contains digit(0-9)

Parameters

<i>ch</i>	This is the character to be checked.
-----------	--------------------------------------

Returns

true if char is a digit (0-9)

7.17.3.4 `_ntoa_format()` `static size_t _ntoa_format (`
`out.fct.type out,`
`char * buffer,`
`size_t idx,`
`size_t maxlen,`
`char * buf,`
`size_t len,`
`bool negative,`
`unsigned int base,`
`unsigned int prec,`
`unsigned int width,`
`unsigned int flags) [static]`

`_ntoa_format()` - Convert the string into the defined format structure.

This function converts the string type into a specified format. A while condition for padding lead zeros in the given values, To handle hash and sign flags and if else conditions are used. A function is used to reverse string. Based on given width padding spaces are appended.

Parameters

<i>out</i>	type of out.fct.type
<i>buffer</i>	Pointer to a character string to write the result.
<i>idx</i>	idx bytes of size_t
<i>maxlen</i>	Maximum number of characters to write.
<i>negative</i>	boolean type
<i>base</i>	an unsigned long data type
<i>prec</i>	an unsigned integral data type
<i>width</i>	an unsigned integral data type
<i>flags</i>	an unsigned integral data type

Returns

idx non integer value.

```

7.17.3.5 _ntoa_long() static size_t _ntoa_long (
    out.fct_type out,
    char * buffer,
    size_t idx,
    size_t maxlen,
    unsigned long value,
    bool negative,
    unsigned long base,
    unsigned int prec,
    unsigned int width,
    unsigned int flags ) [static]

```

[_ntoa_long\(\)](#) - Function is used for string into structure value.

In the `_ntoa_long` function, char buffer is initialized. If condition to check for no hash value for zero and flags precision is valid or not and the digits specified in the format matches with the string, some uppercase case constraints are included to validate the string finally returns the value with ntoa internal itoa for 'long' type.

Parameters

<i>out</i>	type of out.fct_type
<i>buffer</i>	Pointer to a character string to write the result.
<i>idx</i>	idx bytes of size_t
<i>maxlen</i>	Maximum number of characters to write.
<i>negative</i>	boolean type
<i>base</i>	an unsigned long data type
<i>prec</i>	an unsigned integral data type
<i>width</i>	an unsigned integral data type
<i>flags</i>	an unsigned integral data type

Returns

`_ntoa_format` It returns `_ntoa_format` function.

```

7.17.3.6 _ntoa_long_long() static size_t _ntoa_long_long (
    out.fct_type out,
    char * buffer,
    size_t idx,
    size_t maxlen,
    unsigned long long value,
    bool negative,
    unsigned long long base,
    unsigned int prec,
    unsigned int width,
    unsigned int flags ) [static]

```

[_ntoa_long_long\(\)](#) - Function to convert string to a struct.

This `_ntoa_long_long` function firstly initializes the variables and checks for no hash for zero values. using the do while condition it checks for buf length, digits less than ten or not and its return the computed values into ntoa format.

Parameters

<i>out</i>	type of out.fct_type
<i>buffer</i>	Pointer to a character string to write the result.
<i>idx</i>	idx bytes of size_t
<i>maxlen</i>	Maximum number of characters to write.
<i>negative</i>	boolean type
<i>base</i>	an unsigned long data type
<i>prec</i>	an unsigned integral data type
<i>width</i>	an unsigned integral data type
<i>flags</i>	an unsigned integral data type

Returns

_ntoa_format It returns _ntoa_format function.

7.17.3.7 `_out_buffer()` `static void _out_buffer (`
 `char character,`
 `void * buffer,`
 `size_t idx,`
 `size_t maxlen)` `[inline], [static]`

[_out_buffer\(\)](#) - Internal buffer output

This function checks the idx and maxlen, If "idx" is less than "maxlen" then it will assign "character" value into the typecasting char "buffer[idx]"

Parameters

<i>character</i>	character type string
<i>buffer</i>	Pointer to a character string to write the result.
<i>idx</i>	bytes of size_t
<i>maxlen</i>	Maximum number of characters to write.

7.17.3.8 `_out_char()` `static void _out_char (`
 `char character,`
 `void * buffer,`
 `size_t idx,`
 `size_t maxlen)` `[inline], [static]`

[_out_char\(\)](#) - Internal putchar wrapper

The typecasting of arguments with void is to avoid unused variable warnings in some compilers. Checks the character value, if the condition satisfies then [putchar\(\)](#) writes a character into stdout.

Parameters

<i>character</i>	character type string
<i>buffer</i>	Pointer to a character string to write the result.
<i>idx</i>	bytes of size_t
<i>maxlen</i>	Maximum number of characters to write.

7.17.3.9 `_out_fct()` `static void _out_fct (`
 `char character,`
 `void * buffer,`
 `size_t idx,`
 `size_t maxlen) [inline], [static]`

`_out_fct()` - Internal output function wrapper

This function typecasting idx and maxlen arguments to avoid compiler error. output function wrapper and the buffer is the output fct pointer.

Parameters

<i>character</i>	character type string
<i>buffer</i>	Pointer to a character string to write the result.
<i>idx</i>	bytes of size_t
<i>maxlen</i>	Maximum number of characters to write.

7.17.3.10 `_out_null()` `static void _out_null (`
 `char character,`
 `void * buffer,`
 `size_t idx,`
 `size_t maxlen) [inline], [static]`

`_out_null()` - Internal null output.

The typecasting of arguments with void is to avoid unused variable warnings in some compilers.

Parameters

<i>character</i>	character type string
<i>buffer</i>	Pointer to a character string to write the result.
<i>idx</i>	bytes of size_t
<i>maxlen</i>	Maximum number of characters to write.

7.17.3.11 `_strlen()` `static unsigned int _strlen (`
`const char * str) [inline], [static]`

`_strlen()` - Calculate length of characters in str.

Parameters

<i>str</i>	str is argument of type pointer.
------------	----------------------------------

Returns

string dtring length is returned

7.17.3.12 `_vsnprintf()` `static int _vsnprintf (`
`out_fct_type out,`
`char * buffer,`
`const size_t maxlen,`
`const char * format,`
`va_list va) [static]`

`_vsnprintf()` - Function write formatted output to a character array, up to a maximum number of characters (varargs) and evaluation of format specifiers are happening in this function.

The `_vsnprintf` function firstly initialize the variables of format specifiers like flags, width, precision in this they evaluating all the specifiers individually. firstly checks the buffer equal to zero for null out function. after that flags evaluation is done by using switch case, then width field evaluation is processed. Length field is evaluated. if `PRINTF_SUPPORT_PTRDIFF_T` is true then respective case is invoked and format is incremented. Specifier is evaluated and base value is assigned based on respective conditions. Finally converts into integer format and returns type-casted idx

Parameters

<i>out</i>	type of out_fct.type.
<i>buffer</i>	pointer to the buffer where you want to function to store the formatted string.
<i>maxlen</i>	maximum number of characters to store in the buffer.
<i>format</i>	string that specifies the format of the output.
<i>va</i>	variable-argument list of the additional argument.

Returns

Its return the typecasted int of idx if success

7.17.3.13 `fctprintf()` `int fctprintf (`
`void(*) (char character, void *arg) out,`

```
void * arg,  
const char * format,  
... )
```

[fctprintf\(\)](#) - Function uses the library macros of variable arguments like `vastart` and `vaend`.

In this function `va_start()` is invoked and in variable 'ret' the value from `_vsnprintf` function will be assigned and then `va_end()` function is invoked. Finally ret value is returned.

Parameters

<i>out</i>	An output function which takes one character and an argument pointer.
<i>arg</i>	An argument pointer for user data passed to output function.
<i>format</i>	A string that specifies the format of the output.

Returns

The number of characters that are sent to the output function, not counting the terminating null character.

7.17.3.14 putchar() `int putchar (`
`char ch)`

7.17.3.15 snprintf() `int snprintf (`
`char * buffer,`
`size_t count,`
`const char * format,`
`...)`

[snprintf\(\)](#) - Places the generated output into the character array pointed to by `buf`, instead of writing it to a file

In this function `va_start()` is invoked and in variable 'ret' the value from `_vsnprintf` function will be assigned and then `va_end()` function is invoked. Finally ret value is returned.

Parameters

<i>buffer</i>	pointer to buffer where you want to function to store the formatted string.
<i>count</i>	maximum number of characters to store in the buffer.
<i>format</i>	string that specifies the format of the output.

Returns

ret returns the ret value as an integer type.

```
7.17.3.16 sprintf() int sprintf (
    char * buffer,
    const char * format,
    ... )
```

[sprintf\(\)](#) - Function sends formatted output to a string pointed to by the argument *buffer*.

In this function `va_start()` is invoked and in variable 'ret' the value from `_vsprintf` function will be assigned and then `va_end()` function is invoked. Finally ret value is returned.

Parameters

<i>buffer</i>	pointer to an array of char elements resulting string will store.
<i>format</i>	string that contains the text to be written to buffer.

Returns

ret Its returns the ret value as an integer type.

```
7.17.3.17 vsnprintf() int vsnprintf (
    char * buffer,
    size_t count,
    const char * format,
    va_list va )
```

[vsnprintf\(\)](#) - Returns another function called `_vsnprintf()` with some arguments.

Parameters

<i>buffer</i>	Pointer to the buffer where you want to function to store the formatted string.
<i>count</i>	maximum number of characters to store in the buffer.
<i>format</i>	string that specifies the format of the output.

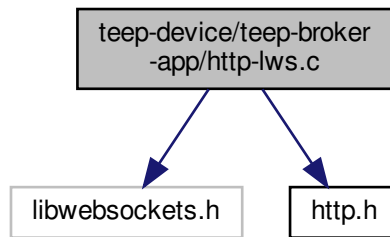
Returns

Its return the typecasted int of idx if success otherwise error occurred.

7.18 teep-device/teep-broker-app/http-lws.c File Reference

```
#include <libwebsockets.h>
#include "http.h"
```


Include dependency graph for `http-lws.c`:



Classes

- struct `lao_rpc_io`
- struct `libteep_async`

Typedefs

- typedef enum `tam_result` `tam_result`

Enumerations

- enum `tam_result` {
`TR_ONGOING` = 1 , `TR_OKAY` = 0 , `TR_FAIL_START` = -1 , `TR_FAIL_CONN_ERR` = -2 ,
`TR_FAIL_REFUSED` = -3 , `TR_FAIL_OVERSIZE` = -4 , `TR_FAIL_CLOSED` = -5 }

Functions

- static int `callback_teep` (struct lws *wsi, enum lws_callback_reasons reason, void *user, void *in, size_t len)
- static int `callback_download_ta_image` (struct lws *wsi, enum lws_callback_reasons reason, void *user, void *in, size_t len)
- int `http_get` (const char *url, void *out, size_t *out_len)
- int `http_post` (const char *url, const void *in, size_t in_len, void *out, size_t *out_len)

Variables

- static const struct lws_protocols `protocols` []

7.18.1 Typedef Documentation

7.18.1.1 `tam_result` typedef enum `tam_result` `tam_result`

7.18.2 Enumeration Type Documentation

7.18.2.1 `tam_result` enum `tam_result`

Enumerator

TR_ONGOING	
TR_OKAY	
TR_FAIL_START	
TR_FAIL_CONN_ERR	
TR_FAIL_REFUSED	
TR_FAIL_OVERSIZE	
TR_FAIL_CLOSED	

7.18.3 Function Documentation

7.18.3.1 callback.download_ta_image() static int callback.download_ta_image (

```

    struct lws * wsi,
    enum lws_callback_reasons reason,
    void * user,
    void * in,
    size_t len ) [static]
```

7.18.3.2 callback.teep() static int callback.teep (

```

    struct lws * wsi,
    enum lws_callback_reasons reason,
    void * user,
    void * in,
    size_t len ) [static]
```

7.18.3.3 http.get() int http.get (

```

    const char * url,
    void * out,
    size_t * out_len )
```

7.18.3.4 http.post() int http.post (

```

    const char * url,
    const void * in,
    size_t in_len,
    void * out,
    size_t * out_len )
```

7.18.4 Variable Documentation

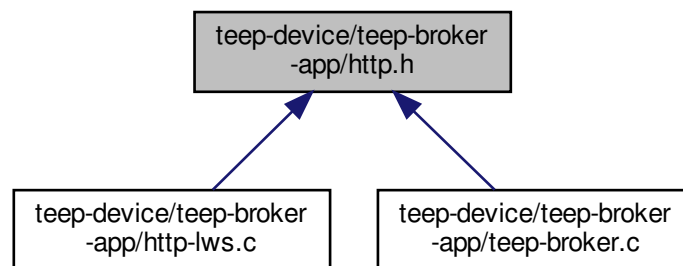
7.18.4.1 protocols `const struct lws_protocols protocols[] [static]`

Initial value:

```
= {
    { "teep", callback_teep, 0, 4096, },
    { "downloadta_image", callback_downloadta_image, 0, 4096, },
    { NULL, NULL, 0, 0 }
}
```

7.19 teep-device/teep-broker-app/http.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int [http_get](#) (const char *uri, void *out, size_t *out_len)
- int [http_post](#) (const char *uri, const void *in, size_t in_len, void *out, size_t *out_len)

7.19.1 Function Documentation

7.19.1.1 http_get() `int http_get (`
`const char * uri,`
`void * out,`
`size_t * out_len)`

7.19.1.2 http_post() `int http_post (`
`const char * uri,`
`const void * in,`
`size_t in_len,`
`void * out,`
`size_t * out_len)`

7.20 teep-device/teep-broker-app/teec-keystone.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/time.h>
#include <sys/random.h>
#include <fcntl.h>
#include <unistd.h>
#include <cstdio>
#include <string>
#include <cstring>
#include <thread>
#include <mutex>
#include <condition_variable>
#include "edger/Enclave_u.h"
#include "tee_client_api.h"
#include <libteep.h>
#include <libwebsockets.h>
#include "ta-interface.h"
```

Include dependency graph for teec-keystone.cpp:



Classes

- struct [Command](#)
- class [CommandQueue](#)

Functions

- TEEC_Result [TEEC_InitializeContext](#) (const char *name, TEEC_Context *context)
- void [TEEC_FinalizeContext](#) (TEEC_Context *context)
- TEEC_Result [TEEC_OpenSession](#) (TEEC_Context *context, TEEC_Session *session, const TEEC_UUID *destination, uint32_t connectionMethod, const void *connectionData, TEEC_Operation *operation, uint32_t *returnOrigin)
- void [TEEC_CloseSession](#) (TEEC_Session *session)
- TEEC_Result [TEEC_InvokeCommand](#) (TEEC_Session *session, uint32_t commandID, TEEC_Operation *operation, uint32_t *returnOrigin)
- EDGE_EXTERN BEGIN invoke_command_t [ocall_pull_invoke_command](#) ()
- param_buffer_t [ocall_read_invoke_param](#) (int index, size_t offset)
- void [ocall_write_invoke_param](#) (int index, size_t offset, size_t size, const char *buf)
- void [ocall_put_invoke_command_result](#) (invoke_command_t cmd, unsigned int result)

Variables

- const char * [enc_path](#) = "teep-agent-ta"
- const char * [runtime_path](#) = "eyrie-rt"
- static [CommandQueue](#) queue
- static Keystone [enclave](#)
- static std::thread [enclave_thread](#)

7.20.1 Function Documentation

7.20.1.1 ocall_pull_invoke_command() `EDGE_EXTERNC_BEGIN invoke_command_t ocall_pull_invoke_command ()`

7.20.1.2 ocall_put_invoke_command_result() `void ocall_put_invoke_command_result (invoke_command_t cmd, unsigned int result)`

7.20.1.3 ocall_read_invoke_param() `param_buffer_t ocall_read_invoke_param (int index, size_t offset)`

7.20.1.4 ocall_write_invoke_param() `void ocall_write_invoke_param (int index, size_t offset, size_t size, const char * buf)`

7.20.1.5 TEEC_CloseSession() `void TEEC_CloseSession (TEEC_Session * session)`

7.20.1.6 TEEC_FinalizeContext() `void TEEC_FinalizeContext (TEEC_Context * context)`

7.20.1.7 TEEC_InitializeContext() `TEEC_Result TEEC_InitializeContext (const char * name, TEEC_Context * context)`

7.20.1.8 TEEC_InvokeCommand() `TEEC_Result TEEC_InvokeCommand (`
 `TEEC_Session * session,`
 `uint32_t commandID,`
 `TEEC_Operation * operation,`
 `uint32_t * returnOrigin)`

7.20.1.9 TEEC_OpenSession() `TEEC_Result TEEC_OpenSession (`
 `TEEC_Context * context,`
 `TEEC_Session * session,`
 `const TEEC_UUID * destination,`
 `uint32_t connectionMethod,`
 `const void * connectionData,`
 `TEEC_Operation * operation,`
 `uint32_t * returnOrigin)`

7.20.2 Variable Documentation

7.20.2.1 enc_path `const char* enc_path = "teep-agent-ta"`

7.20.2.2 enclave `Keystone enclave [static]`

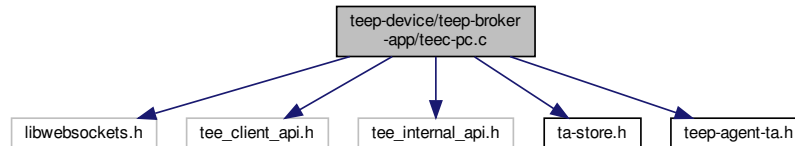
7.20.2.3 enclave_thread `std::thread enclave_thread [static]`

7.20.2.4 queue `CommandQueue queue [static]`

7.20.2.5 runtime_path `const char* runtime_path = "eyrie-rt"`

7.21 teep-device/teep-broker-app/teec-pc.c File Reference

```
#include <libwebsockets.h>
#include <tee_client_api.h>
#include <tee_internal_api.h>
#include "ta-store.h"
#include "teep-agent-ta.h"
Include dependency graph for teec-pc.c:
```



Functions

- static void [prepare_params](#) (TEEC.Operation *operation, uint32_t *types, TEE_Param params[4])
- static void [writeback_params](#) (TEEC.Operation *operation, uint32_t types, TEE_Param params[4])
- TEEC.Result [TEEC_InitializeContext](#) (const char *name, TEEC_Context *context)
- void [TEEC_FinalizeContext](#) (TEEC_Context *context)
- TEEC.Result [TEEC_OpenSession](#) (TEEC_Context *context, TEEC.Session *session, const TEEC.UUID *destination, uint32_t connectionMethod, const void *connectionData, TEEC.Operation *operation, uint32_t *returnOrigin)
- void [TEEC_CloseSession](#) (TEEC.Session *session)
- TEEC.Result [TEEC_InvokeCommand](#) (TEEC.Session *session, uint32_t commandID, TEEC.Operation *operation, uint32_t *returnOrigin)
- TEEC.Result [TEEC_RegisterSharedMemory](#) (TEEC.Context *context, TEEC.SharedMemory *sharedMem)
- TEEC.Result [TEEC_AllocateSharedMemory](#) (TEEC.Context *context, TEEC.SharedMemory *sharedMem)
- void [TEEC_ReleaseSharedMemory](#) (TEEC.SharedMemory *sharedMemory)
- void [TEEC_RequestCancellation](#) (TEEC.Operation *operation)

Variables

- static void * [session_ctx](#)

7.21.1 Function Documentation

7.21.1.1 prepare_params() static void prepare_params (

```
TEEC.Operation * operation,
uint32_t * types,
TEE_Param params[4] ) [static]
```

7.21.1.2 TEEC_AllocateSharedMemory() `TEEC_Result TEEC_AllocateSharedMemory (`
 `TEEC_Context * context,`
 `TEEC_SharedMemory * sharedMem)`

7.21.1.3 TEEC_CloseSession() `void TEEC_CloseSession (`
 `TEEC_Session * session)`

7.21.1.4 TEEC_FinalizeContext() `void TEEC_FinalizeContext (`
 `TEEC_Context * context)`

7.21.1.5 TEEC_InitializeContext() `TEEC_Result TEEC_InitializeContext (`
 `const char * name,`
 `TEEC_Context * context)`

7.21.1.6 TEEC_InvokeCommand() `TEEC_Result TEEC_InvokeCommand (`
 `TEEC_Session * session,`
 `uint32_t commandID,`
 `TEEC_Operation * operation,`
 `uint32_t * returnOrigin)`

7.21.1.7 TEEC_OpenSession() `TEEC_Result TEEC_OpenSession (`
 `TEEC_Context * context,`
 `TEEC_Session * session,`
 `const TEEC_UUID * destination,`
 `uint32_t connectionMethod,`
 `const void * connectionData,`
 `TEEC_Operation * operation,`
 `uint32_t * returnOrigin)`

7.21.1.8 TEEC_RegisterSharedMemory() `TEEC_Result TEEC_RegisterSharedMemory (`
 `TEEC_Context * context,`
 `TEEC_SharedMemory * sharedMem)`

7.21.1.9 TEEC_ReleaseSharedMemory() `void TEEC_ReleaseSharedMemory (`
 `TEEC_SharedMemory * sharedMemory)`

7.21.1.10 TEEC.RequestCancellation() void TEEC.RequestCancellation (
 TEEC.Operation * operation)

7.21.1.11 writeback_params() static void writeback_params (
 TEEC.Operation * operation,
 uint32_t types,
 TEE.Param params[4]) [static]

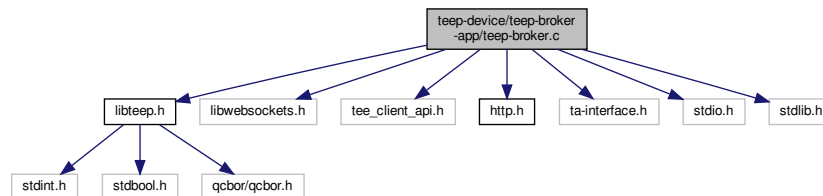
7.21.2 Variable Documentation

7.21.2.1 session_ctx void* session_ctx [static]

7.22 teep-device/teep-broker-app/teep-broker.c File Reference

```
#include <libteep.h>
#include <libwebsockets.h>
#include <tee_client_api.h>
#include "http.h"
#include "ta-interface.h"
#include <stdio.h>
#include <stdlib.h>
```

Include dependency graph for teep-broker.c:



Classes

- struct [broker_ctx](#)

Functions

- static int [broker_ctx_init](#) (struct [broker_ctx](#) *ctx)
- static void [broker_ctx_destroy](#) (struct [broker_ctx](#) *ctx)
- static int [set_agent_dev_option](#) (struct [broker_ctx](#) *ctx, enum agent_dev_option option, const char *value)
- static int [broker_task_done](#) (struct [broker_ctx](#) *ctx, const void *in, size_t in_len)
- static int [agent_query_next_broker_task](#) (struct [broker_ctx](#) *ctx, struct broker_task *task)
- static void [usage](#) (void)
- static void [cmdline_parse](#) (int argc, const char *argv[])
- static int [broker_http_post](#) (struct [broker_ctx](#) *ctx, const struct broker_task *task)
- static int [broker_http_get](#) (struct [broker_ctx](#) *ctx, const struct broker_task *task)
- static int [loop_teep](#) (struct [broker_ctx](#) *ctx)
- int [broker_main](#) ()
- int [main](#) (int argc, const char **argv)

Variables

- static const TEEC_UUID `uuid_aist_otrp_ta`
- static uint8_t `http_res_buf` [6 *1024 *1024]
- const char * `uri` = "http://127.0.0.1:3000/api/tam"
- const char * `alist` = ""
- bool `cose` = false

7.22.1 Function Documentation

7.22.1.1 agent_query_next_broker_task() static int agent_query_next_broker_task (struct `broker_ctx` * `ctx`, struct `broker_task` * `task`) [static]

7.22.1.2 broker_ctx_destroy() static void broker_ctx.destroy (struct `broker_ctx` * `ctx`) [static]

7.22.1.3 broker_ctx_init() static int broker_ctx.init (struct `broker_ctx` * `ctx`) [static]

7.22.1.4 broker_http_get() static int broker_http_get (struct `broker_ctx` * `ctx`, const struct `broker_task` * `task`) [static]

7.22.1.5 broker_http_post() static int broker_http_post (struct `broker_ctx` * `ctx`, const struct `broker_task` * `task`) [static]

7.22.1.6 broker_main() int broker_main ()

7.22.1.7 broker_task_done() static int broker_task_done (struct `broker_ctx` * `ctx`, const void * `in`, size_t `in_len`) [static]

7.22.1.8 cmdline_parse() static void cmdline_parse (
int argc,
const char * argv[]) [static]

7.22.1.9 loop_teep() static int loop_teep (
struct broker_ctx * ctx) [static]

loop_teep - The teep message request.

This function has a loop. The loop condition is based on the tam message and for each iteration it will go through the every switch case and if the switch statement matches with the type it will invoke the respective function; if it does not match, then it executes the default case.

Parameters

<i>lao_ctx</i>	It is an object of structure libteep context.
----------------	---

Returns

0 If success, else error occurred.

7.22.1.10 main() int main (
int argc,
const char ** argv)

7.22.1.11 set_agent_dev_option() static int set_agent_dev_option (
struct broker_ctx * ctx,
enum agent_dev_option option,
const char * value) [static]

7.22.1.12 usage() static void usage (
void) [static]

7.22.2 Variable Documentation

7.22.2.1 cose `bool cose = false`

7.22.2.2 http_res_buf `uint8_t http_res_buf[6 * 1024 * 1024] [static]`

7.22.2.3 talist `const char* talist = ""`

7.22.2.4 uri `const char* uri = "http://127.0.0.1:3000/api/tam"`

7.22.2.5 uuid_aist_otrp_ta `const TEEC_UUID uuid_aist_otrp_ta [static]`

Initial value:

```
=  
    { 0x68373894, 0x5bb3, 0x403c,  
      { 0x9e, 0xec, 0x31, 0x14, 0xa1, 0xf5, 0xd3, 0xfc } }
```


Index

- [_atoi](#)
 - [vsnprintf.c, 61](#)
 - [_ftoa](#)
 - [vsnprintf.c, 62](#)
 - [_is_digit](#)
 - [vsnprintf.c, 62](#)
 - [_ntoa_format](#)
 - [vsnprintf.c, 63](#)
 - [_ntoa_long](#)
 - [vsnprintf.c, 63](#)
 - [_ntoa_long_long](#)
 - [vsnprintf.c, 64](#)
 - [_out_buffer](#)
 - [vsnprintf.c, 65](#)
 - [_out_char](#)
 - [vsnprintf.c, 65](#)
 - [_out_fct](#)
 - [vsnprintf.c, 66](#)
 - [_out_null](#)
 - [vsnprintf.c, 66](#)
 - [_putchar](#)
 - [vsnprintf.c, 59](#)
 - [_strlen](#)
 - [tools.c, 56](#)
 - [vsnprintf.c, 66](#)
 - [_vsnprintf](#)
 - [vsnprintf.c, 67](#)
- [AGENT_DOWNLOAD_TA](#)
 - [teep-agent-ta.c, 50](#)
- [AGENT_FINISH](#)
 - [teep-agent-ta.c, 50](#)
- [AGENT_INIT](#)
 - [teep-agent-ta.c, 50](#)
- [AGENT_POSTING_ERROR](#)
 - [teep-agent-ta.c, 50](#)
- [AGENT_POSTING_INITIAL_REQUEST](#)
 - [teep-agent-ta.c, 50](#)
- [AGENT_POSTING_QUERY_RESPONSE](#)
 - [teep-agent-ta.c, 50](#)
- [AGENT_POSTING_SUCCESS](#)
 - [teep-agent-ta.c, 50](#)
- [agent_query_next_broker_task](#)
 - [teep-broker.c, 79](#)
- [agent_state](#)
 - [teep-agent-ta.c, 50](#)
- [arg](#)
 - [out_fct_wrap_type, 14](#)
- [array](#)
 - [teep_buffer_array, 16](#)
 - [teep_tc_info_array, 23](#)
 - [teep_uint32_array, 24](#)
- [atoi](#)
 - [tools.c, 56](#)
- [broker_ctx, 10](#)
 - [tee_context, 10](#)
 - [tee_session, 10](#)
- [broker_ctx_destroy](#)
 - [teep-broker.c, 79](#)
- [broker_ctx_init](#)
 - [teep-broker.c, 79](#)
- [broker_http_get](#)
 - [teep-broker.c, 79](#)
- [broker_http_post](#)
 - [teep-broker.c, 79](#)
- [broker_main](#)
 - [teep-broker.c, 79](#)
- [broker_task_done](#)
 - [teep-agent-ta.c, 51](#)
 - [teep-broker.c, 79](#)
- [build_error](#)
 - [teep-agent-ta.c, 51](#)
- [build_query_response](#)
 - [teep-agent-ta.c, 51](#)
- [build_success](#)
 - [teep-agent-ta.c, 51](#)
- [callback_download_ta_image](#)
 - [http-lws.c, 71](#)
- [callback_teep](#)
 - [http-lws.c, 71](#)
- [challenge](#)
 - [teep_message, 18](#)
- [cmdline_parse](#)
 - [teep-broker.c, 79](#)
- [Command, 10](#)
 - [command, 10](#)
 - [command_result, 10](#)
 - [operation, 11](#)
- [command](#)
 - [Command, 10](#)
- [command_result](#)
 - [Command, 10](#)
- [CommandQueue, 11](#)
 - [pull_invoke_command, 11](#)
 - [pull_invoke_command_result, 11](#)
 - [put_invoke_command, 11](#)
 - [put_invoke_command_result, 11](#)
 - [read_invoke_param, 11](#)
 - [write_invoke_param, 12](#)
- [component_id](#)
 - [teep_tc_info, 22](#)
- [context](#)
 - [optee-main.c, 32](#)
- [copyout_param](#)
 - [teep-agent-ta.c, 51](#)
- [cose](#)
 - [teep-broker.c, 80](#)
- [data_item_requested](#)
 - [teep_agent_session, 15](#)

- teep_message, 18
- download_ta_index
 - teep_agent_session, 15
- EC
 - teep_message_encoder, 21
- enc_path
 - teec-keystone.cpp, 75
- enclave
 - teec-keystone.cpp, 75
- enclave_thread
 - teec-keystone.cpp, 75
- err_code
 - teep_message, 19
- err_msg
 - teep_message, 19
- evidence
 - teep_message, 19
- evidence_format
 - teep_message, 19
- ext_list
 - teep_message, 19
- fct
 - out_fct_wrap_type, 14
- fctprintf
 - vsnprintf.c, 67
- file_length
 - optee-main.c, 32
- filecontents
 - optee-main.c, 32
- FLAGS.CHAR
 - vsnprintf.c, 60
- FLAGS.HASH
 - vsnprintf.c, 60
- FLAGS.LEFT
 - vsnprintf.c, 60
- FLAGS.LONG
 - vsnprintf.c, 60
- FLAGS.LONG_LONG
 - vsnprintf.c, 60
- FLAGS.PLUS
 - vsnprintf.c, 60
- FLAGS.PRECISION
 - vsnprintf.c, 60
- FLAGS.SHORT
 - vsnprintf.c, 60
- FLAGS.SPACE
 - vsnprintf.c, 60
- FLAGS.UPPERCASE
 - vsnprintf.c, 60
- FLAGS.ZEROPAD
 - vsnprintf.c, 60
- free_parsed_teep_message
 - libteep.c, 37
 - libteep.h, 43
- get_lws_context
 - ta-store.c, 45
- gettimeofday
 - tools.c, 56
- handle_ta_download
 - teep-agent-ta.c, 51
- handle_tam_message
 - teep-agent-ta.c, 51
- handle_TEEP_AGENT_BROKER_TASK_DONE
 - teep-agent-ta.c, 52
- handle_TEEP_AGENT_QUERY_NEXT_BROKER_TASK
 - teep-agent-ta.c, 52
- handle_TEEP_AGENT_SET_DEV_OPTION
 - teep-agent-ta.c, 52
- have_binary
 - teep_tc.info, 22
- have_value
 - teep_buffer_array, 17
 - teep_tc.info_array, 23
 - teep_uint32_array, 24
 - teep_uint32_option, 24
- hello-ta.c
 - STR_TRACE_USER_TA, 33
 - TA_CloseSessionEntryPoint, 33
 - TA_CreateEntryPoint, 33
 - TA_DestroyEntryPoint, 33
 - TA_InvokeCommandEntryPoint, 34
 - TA_OpenSessionEntryPoint, 34
- hex
 - ta-store.c, 46
- http-lws.c
 - callback_download_ta_image, 71
 - callback_teep, 71
 - http_get, 71
 - http_post, 71
 - protocols, 71
 - tam_result, 70
 - TR_FAIL_CLOSED, 71
 - TR_FAIL_CONN_ERR, 71
 - TR_FAIL_OVERSIZE, 71
 - TR_FAIL_REFUSED, 71
 - TR_FAIL_START, 71
 - TR_OKAY, 71
 - TR_ONGOING, 71
- http.h
 - http_get, 72
 - http_post, 72
- http_get
 - http-lws.c, 71
 - http.h, 72
- http_post
 - http-lws.c, 71
 - http.h, 72
- http_res_buf
 - teep-broker.c, 81
- http_resp
 - libteep_async, 13
- id
 - ta_manifest, 14

- in
 - lao_rpc_io, 12
- in_len
 - lao_rpc_io, 12
- io
 - libteep_async, 13
- keystone-main.cpp
 - main, 26
 - ocall_pull_invoke_command, 26
 - ocall_put_invoke_command_result, 26
 - ocall_read_invoke_param, 26
 - ocall_write_invoke_param, 28
 - TEEC_CloseSession, 28
 - TEEC_FinalizeContext, 28
 - TEEC_InitializeContext, 29
 - TEEC_InvokeCommand, 29
 - TEEC_OpenSession, 30
- lao_rpc_io, 12
 - in, 12
 - in_len, 12
 - out, 12
 - out_len, 12
- len
 - teep_buffer_array, 17
 - teep_tc_info_array, 23
 - teep_uint32_array, 24
- libteep.c
 - free_parsed_teeep_message, 37
 - parse_buffer_array, 38
 - parse_option, 38
 - parse_options, 38
 - parse_tc_info_array, 38
 - parse_teeep_message, 38
 - parse_uint32_array, 38
 - parse_uint32_option, 38
 - teep_message_encoder_add_err_code, 39
 - teep_message_encoder_add_header, 39
 - teep_message_encoder_add_ta_to_ta_list, 39
 - teep_message_encoder_close_options, 39
 - teep_message_encoder_close_ta_list, 39
 - teep_message_encoder_finish, 39
 - teep_message_encoder_init, 39
 - teep_message_encoder_open_options, 39
 - teep_message_encoder_open_ta_list, 40
- libteep.h
 - free_parsed_teeep_message, 43
 - parse_teeep_message, 43
 - TEEP_AES_CCM_16_64_128_HMAC256_256_P_256_ES256, 43
 - TEEP_AES_CCM_16_64_128_HMAC256_256_X25519_FFDSEA, 43
 - TEEP_DATA_ATTESTATION, 41
 - TEEP_DATA_EXTENSIONS, 41
 - teep_data_item, 41
 - TEEP_DATA_SUIT_COMMANDS, 41
 - TEEP_DATA_TRUSTED_COMPONENTS, 41
 - TEEP_DELETE, 42
 - TEEP_ERR_BAD_CERTIFICATE, 42
 - TEEP_ERR_CERTIFICATE_EXPIRED, 42
 - TEEP_ERR_CERTIFICATE_REVOKED, 42
 - TEEP_ERR_ILLEGAL_PARAMETER, 42
 - TEEP_ERR_INTERNAL_ERROR, 42
 - TEEP_ERR_MANIFEST_PROCESSING_FAILED, 42
 - TEEP_ERR_REQUEST_SIGNATURE_FAILED, 42
 - TEEP_ERR_TC_NOT_FOUND, 42
 - TEEP_ERR_UNSUPPORTED_CERTIFICATE, 42
 - TEEP_ERR_UNSUPPORTED_CRYPT_ALG, 42
 - TEEP_ERR_UNSUPPORTED_EXTENSION, 42
 - TEEP_ERR_UNSUPPORTED_MSG_VERSION, 42
 - TEEP_ERROR, 42
 - teep_error, 42
 - TEEP_INSTALL, 42
 - teep_message_encoder_add_err_code, 43
 - teep_message_encoder_add_header, 43
 - teep_message_encoder_add_ta_to_ta_list, 43
 - teep_message_encoder_close_options, 44
 - teep_message_encoder_close_ta_list, 44
 - teep_message_encoder_finish, 44
 - teep_message_encoder_init, 44
 - teep_message_encoder_open_options, 44
 - teep_message_encoder_open_ta_list, 44
 - teep_message_type, 42
 - TEEP_OPTION_CHALLENGE, 42
 - TEEP_OPTION_COMPONENT_ID, 43
 - TEEP_OPTION_ERR_MSG, 43
 - TEEP_OPTION_EVIDENCE, 42
 - TEEP_OPTION_EVIDENCE_FORMAT, 43
 - TEEP_OPTION_EXT_LIST, 42
 - TEEP_OPTION_HAVE_BINARY, 43
 - teep_option_key, 42
 - TEEP_OPTION_MANIFEST_LIST, 42
 - TEEP_OPTION_MSG, 43
 - TEEP_OPTION_OCSP_DATA, 42
 - TEEP_OPTION_REQUESTED_TC_LIST, 43
 - TEEP_OPTION_SELECTED_CIPHER_SUIT, 42
 - TEEP_OPTION_SELECTED_VERSION, 42
 - TEEP_OPTION_SUIT_REPORTS, 43
 - TEEP_OPTION_SUPPORTED_CIPHER_SUITS, 42
 - TEEP_OPTION_TC_LIST, 42
 - TEEP_OPTION_TC_MANIFEST_SEQUENCE_NUMBER, 43
 - TEEP_OPTION_UNNEEDED_TC_LIST, 43
 - TEEP_OPTION_VERSIONS, 42
 - TEEP_QUERY_REQUEST, 42
 - TEEP_QUERY_RESPONSE, 42
 - TEEP_SUCCESS, 42
 - teep_suite, 43
 - libteep_async, 13
 - http_resp, 13
 - io, 13
 - max_out_len, 13
 - result, 13
 - wsi, 13
 - loop_teeep

- teep-broker.c, 80
- main
 - keystone-main.cpp, 26
 - optee-main.c, 31
 - teep-broker.c, 80
- manifest_list
 - teep_message, 19
- manifests
 - teep_agent_session, 15
- manifests_len
 - teep_agent_session, 16
- max_out_len
 - libteep_async, 13
- msg
 - teep_message, 19
- ocall_pull_invoke_command
 - keystone-main.cpp, 26
 - teec-keystone.cpp, 74
- ocall_put_invoke_command_result
 - keystone-main.cpp, 26
 - teec-keystone.cpp, 74
- ocall_read_invoke_param
 - keystone-main.cpp, 26
 - teec-keystone.cpp, 74
- ocall_write_invoke_param
 - keystone-main.cpp, 28
 - teec-keystone.cpp, 74
- ocsp_data
 - teep_message, 19
- on_going_task
 - teep_agent_session, 16
- operation
 - Command, 11
- optee-main.c
 - context, 32
 - file_length, 32
 - filecontents, 32
 - main, 31
 - session, 32
 - shm, 32
 - sp_hello_app, 31
 - uuid, 32
- out
 - lao_rpc_io, 12
- out_fct_type
 - vsnprintf.c, 61
- out_fct_wrap_type, 14
 - arg, 14
 - fct, 14
- out_len
 - lao_rpc_io, 12
- parse_buffer_array
 - libteep.c, 38
- parse_option
 - libteep.c, 38
- parse_options
 - libteep.c, 38
- parse_tc_info_array
 - libteep.c, 38
- parse_teep_message
 - libteep.c, 38
 - libteep.h, 43
- parse_uint32_array
 - libteep.c, 38
- parse_uint32_option
 - libteep.c, 38
- prepare_params
 - teec-pc.c, 76
- PRINTF_FTOA_BUFFER_SIZE
 - vsnprintf.c, 61
- PRINTF_NTOA_BUFFER_SIZE
 - vsnprintf.c, 61
- PRINTF_SUPPORT_FLOAT
 - vsnprintf.c, 61
- PRINTF_SUPPORT_LONG_LONG
 - vsnprintf.c, 61
- PRINTF_SUPPORT_PTRDIFF_T
 - vsnprintf.c, 61
- protocols
 - http-lws.c, 71
- pull_invoke_command
 - CommandQueue, 11
- pull_invoke_command_result
 - CommandQueue, 11
- put_invoke_command
 - CommandQueue, 11
- put_invoke_command_result
 - CommandQueue, 11
- putchar
 - vsnprintf.c, 68
- query_next_broker_task
 - teep-agent-ta.c, 52
- query_request
 - teep_message, 19
- query_response
 - teep_message, 19
- queue
 - teec-keystone.cpp, 75
- read_invoke_param
 - CommandQueue, 11
- requested_tc_list
 - teep_message, 19
- result
 - libteep_async, 13
- runtime_path
 - teec-keystone.cpp, 75
- selected_cipher_suit
 - teep_message, 20
- selected_version
 - teep_message, 20
- session
 - optee-main.c, 32

- session_ctx
 - teec-pc.c, 78
- set_agent_dev_option
 - teep-broker.c, 80
- set_dev_option
 - teep-agent-ta.c, 52
- set_manifest_from_suit
 - teep-agent-ta.c, 52
- set_manifest_from_suit_install
 - teep-agent-ta.c, 52
- set_manifest_from_suit_manifest
 - teep-agent-ta.c, 53
- set_manifest_from_uri
 - teep-agent-ta.c, 53
- shm
 - optee-main.c, 32
- snprintf
 - vsnprintf.c, 68
- sp_hello_app
 - optee-main.c, 31
- sp_pubkey_jwk
 - ta-store.c, 47
- sprintf
 - vsnprintf.c, 68
- state
 - teep_agent_session, 16
- STR_TRACE_USER_TA
 - hello-ta.c, 33
- strchr
 - tools.c, 56
- strcpy
 - tools.c, 57
- strdup
 - tools.c, 57
- string_to_uuid_octets
 - ta-store.c, 46
- strncat
 - tools.c, 57
- strncpy
 - tools.c, 57
- strrchr
 - tools.c, 58
- suit_reports
 - teep_message, 20
- supported_cipher_suits
 - teep_message, 20
- ta-store.c
 - get_lws_context, 45
 - hex, 46
 - sp_pubkey_jwk, 47
 - string_to_uuid_octets, 46
 - ta_image_buf, 47
 - ta_store_delete, 46
 - ta_store_install, 46
 - tee_id_privkey_jwk, 47
 - teep_message_unwrap_ta_image, 47
 - temp_buf, 47
 - TEMP_BUF_SIZE, 45
- ta-store.h
 - ta_store_delete, 48
 - ta_store_install, 48
- TA_CloseSessionEntryPoint
 - hello-ta.c, 33
 - teep-agent-ta.c, 53
- TA_CreateEntryPoint
 - hello-ta.c, 33
 - teep-agent-ta.c, 53
- TA_CURRENT_TA_EXT_PROPERTIES
 - user_ta_header_defines.h, 35, 36
- TA_DATA_SIZE
 - user_ta_header_defines.h, 35, 36
- TA_DestroyEntryPoint
 - hello-ta.c, 33
 - teep-agent-ta.c, 53
- TA_FLAGS
 - user_ta_header_defines.h, 35, 36
- ta_image_buf
 - ta-store.c, 47
- TA_InvokeCommandEntryPoint
 - hello-ta.c, 34
 - teep-agent-ta.c, 54
- ta_list
 - teep_message, 20
- ta_manifest, 14
 - id, 14
 - uri, 14
- TA_OpenSessionEntryPoint
 - hello-ta.c, 34
 - teep-agent-ta.c, 54
- TA_STACK_SIZE
 - user_ta_header_defines.h, 35, 36
- ta_store_delete
 - ta-store.c, 46
 - ta-store.h, 48
- ta_store_install
 - ta-store.c, 46
 - ta-store.h, 48
- TA_UUID
 - user_ta_header_defines.h, 35, 36
- talist
 - teep-broker.c, 81
- tam_result
 - http-lws.c, 70
- tam_uri
 - teep_agent_session, 16
- task_buffer
 - teep_agent_session, 16
- tc_list
 - teep_message, 20
- tc_manifest_sequence_number
 - teep_tc_info, 22
- tee_context
 - broker_ctx, 10
- tee_id_privkey_jwk
 - ta-store.c, 47
- tee_session

- broker_ctx, 10
- teec-keystone.cpp
 - enc_path, 75
 - enclave, 75
 - enclave_thread, 75
 - ocall_pull_invoke_command, 74
 - ocall_put_invoke_command_result, 74
 - ocall_read_invoke_param, 74
 - ocall_write_invoke_param, 74
 - queue, 75
 - runtime_path, 75
 - TEEC_CloseSession, 74
 - TEEC_FinalizeContext, 74
 - TEEC_InitializeContext, 74
 - TEEC_InvokeCommand, 74
 - TEEC_OpenSession, 75
- teec-pc.c
 - prepare_params, 76
 - session_ctx, 78
 - TEEC_AllocateSharedMemory, 76
 - TEEC_CloseSession, 77
 - TEEC_FinalizeContext, 77
 - TEEC_InitializeContext, 77
 - TEEC_InvokeCommand, 77
 - TEEC_OpenSession, 77
 - TEEC_RegisterSharedMemory, 77
 - TEEC_ReleaseSharedMemory, 77
 - TEEC_RequestCancellation, 77
 - writeback_params, 78
- TEEC_AllocateSharedMemory
 - teec-pc.c, 76
- TEEC_CloseSession
 - keystone-main.cpp, 28
 - teec-keystone.cpp, 74
 - teec-pc.c, 77
- TEEC_FinalizeContext
 - keystone-main.cpp, 28
 - teec-keystone.cpp, 74
 - teec-pc.c, 77
- TEEC_InitializeContext
 - keystone-main.cpp, 29
 - teec-keystone.cpp, 74
 - teec-pc.c, 77
- TEEC_InvokeCommand
 - keystone-main.cpp, 29
 - teec-keystone.cpp, 74
 - teec-pc.c, 77
- TEEC_OpenSession
 - keystone-main.cpp, 30
 - teec-keystone.cpp, 75
 - teec-pc.c, 77
- TEEC_RegisterSharedMemory
 - teec-pc.c, 77
- TEEC_ReleaseSharedMemory
 - teec-pc.c, 77
- TEEC_RequestCancellation
 - teec-pc.c, 77
- teep-agent-ta.c
 - AGENT_DOWNLOAD_TA, 50
 - AGENT_FINISH, 50
 - AGENT_INIT, 50
 - AGENT_POSTING_ERROR, 50
 - AGENT_POSTING_INITIAL_REQUEST, 50
 - AGENT_POSTING_QUERY_RESPONSE, 50
 - AGENT_POSTING_SUCCESS, 50
 - agent_state, 50
 - broker_task_done, 51
 - build_error, 51
 - build_query_response, 51
 - build_success, 51
 - copyout_param, 51
 - handle_ta_download, 51
 - handle_tam_message, 51
 - handle_TEEP_AGENT_BROKER_TASK_DONE, 52
 - handle_TEEP_AGENT_QUERY_NEXT_BROKER_TASK, 52
 - handle_TEEP_AGENT_SET_DEV_OPTION, 52
 - query_next_broker_task, 52
 - set_dev_option, 52
 - set_manifest_from_suit, 52
 - set_manifest_from_suit_install, 52
 - set_manifest_from_suit_manifest, 53
 - set_manifest_from_uri, 53
 - TA_CloseSessionEntryPoint, 53
 - TA_CreateEntryPoint, 53
 - TA_DestroyEntryPoint, 53
 - TA_InvokeCommandEntryPoint, 54
 - TA_OpenSessionEntryPoint, 54
 - teep_agent_session_create, 55
 - teep_agent_session_destroy, 55
 - teep_error, 55
- teep-broker.c
 - agent_query_next_broker_task, 79
 - broker_ctx_destroy, 79
 - broker_ctx_init, 79
 - broker_http_get, 79
 - broker_http_post, 79
 - broker_main, 79
 - broker_task_done, 79
 - cmdline_parse, 79
 - cose, 80
 - http_res_buf, 81
 - loop_teep, 80
 - main, 80
 - set_agent_dev_option, 80
 - talist, 81
 - uri, 81
 - usage, 80
 - uuid_aist_otrp_ta, 81
- teep-device/docs/cloning_and_building.md, 25
- teep-device/docs/overview_of_teep-device.md, 25
- teep-device/docs/teep-device_operations.md, 25
- teep-device/hello-app/keystone-main.cpp, 25
- teep-device/hello-app/optee-main.c, 30
- teep-device/hello-ta/hello-ta.c, 32
- teep-device/hello-ta/user_ta_header_defines.h, 35

- teep-device/libteep/lib/libteep.c, 37
- teep-device/libteep/lib/libteep.h, 40
- teep-device/teep-agent-ta/sys/time.h, 44
- teep-device/teep-agent-ta/ta-store.c, 45
- teep-device/teep-agent-ta/ta-store.h, 48
- teep-device/teep-agent-ta/teep-agent-ta.c, 49
- teep-device/teep-agent-ta/teep-agent-ta.h, 55
- teep-device/teep-agent-ta/tools.c, 55
- teep-device/teep-agent-ta/user_ta_header_defines.h, 36
- teep-device/teep-agent-ta/vsnprintf.c, 58
- teep-device/teep-broker-app/http-lws.c, 69
- teep-device/teep-broker-app/http.h, 72
- teep-device/teep-broker-app/teec-keystone.cpp, 73
- teep-device/teep-broker-app/teec-pc.c, 76
- teep-device/teep-broker-app/teep-broker.c, 78
- TEEP_AES_CCM_16_64_128_HMAC256_256_P_256_ES256TEEP_AES_CCM_16_64_128_HMAC256_256_X25519_EdDSA
 - libteep.h, 43
- TEEP_AGENT_SESSION
 - data_item_requested, 15
 - download_ta_index, 15
 - manifests, 15
 - manifests_len, 16
 - on_going_task, 16
 - state, 16
 - ta.uri, 16
 - task_buffer, 16
 - token, 16
- teep_agent_session_create
 - teep-agent-ta.c, 55
- teep_agent_session_destroy
 - teep-agent-ta.c, 55
- teep_buffer_array, 16
 - array, 16
 - have_value, 17
 - len, 17
- TEEP_DATA_ATTESTATION
 - libteep.h, 41
- TEEP_DATA_EXTENSIONS
 - libteep.h, 41
- teep_data_item
 - libteep.h, 41
- TEEP_DATA_SUIT_COMMANDS
 - libteep.h, 41
- TEEP_DATA_TRUSTED_COMPONENTS
 - libteep.h, 41
- TEEP_DELETE
 - libteep.h, 42
- teep_delete
 - teep_message, 20
- TEEP_ERR_BAD_CERTIFICATE
 - libteep.h, 42
- TEEP_ERR_CERTIFICATE_EXPIRED
 - libteep.h, 42
- TEEP_ERR_CERTIFICATE_REVOKED
 - libteep.h, 42
- TEEP_ERR_ILLEGAL_PARAMETER
 - libteep.h, 42
- TEEP_ERR_INTERNAL_ERROR
 - libteep.h, 42
- TEEP_ERR_MANIFEST_PROCESSING_FAILED
 - libteep.h, 42
- TEEP_ERR_REQUEST_SIGNATURE_FAILED
 - libteep.h, 42
- TEEP_ERR_TC_NOT_FOUND
 - libteep.h, 42
- TEEP_ERR_UNSUPPORTED_CERTIFICATE
 - libteep.h, 42
- TEEP_ERR_UNSUPPORTED_CRYPT_ALG
 - libteep.h, 42
- TEEP_ERR_UNSUPPORTED_EXTENSION
 - libteep.h, 42
- TEEP_ERR_UNSUPPORTED_MSG_VERSION
 - libteep.h, 42
- TEEP_ERROR
 - libteep.h, 42
- teep_error
 - libteep.h, 42
 - teep-agent-ta.c, 55
 - teep_message, 20
- TEEP_INSTALL
 - libteep.h, 42
- teep_install
 - teep_message, 20
- teep_message, 17
 - challenge, 18
 - data_item_requested, 18
 - err_code, 19
 - err_msg, 19
 - evidence, 19
 - evidence_format, 19
 - ext_list, 19
 - manifest_list, 19
 - msg, 19
 - ocsp_data, 19
 - query_request, 19
 - query_response, 19
 - requested_tc_list, 19
 - selected_cipher_suit, 20
 - selected_version, 20
 - suit_reports, 20
 - supported_cipher_suits, 20
 - ta_list, 20
 - tc_list, 20
 - teep_delete, 20
 - teep_error, 20
 - teep_install, 20
 - teep_success, 20
 - token, 20
 - type, 21
 - unnneeded_tc_list, 21
 - versions, 21
- teep_message_encoder, 21
 - EC, 21
- teep_message_encoder_add_err_code

- libteep.c, [39](#)
- libteep.h, [43](#)
- teep_message_encoder_add_header
 - libteep.c, [39](#)
 - libteep.h, [43](#)
- teep_message_encoder_add_ta_to_ta_list
 - libteep.c, [39](#)
 - libteep.h, [43](#)
- teep_message_encoder_close_options
 - libteep.c, [39](#)
 - libteep.h, [44](#)
- teep_message_encoder_close_ta_list
 - libteep.c, [39](#)
 - libteep.h, [44](#)
- teep_message_encoder_finish
 - libteep.c, [39](#)
 - libteep.h, [44](#)
- teep_message_encoder_init
 - libteep.c, [39](#)
 - libteep.h, [44](#)
- teep_message_encoder_open_options
 - libteep.c, [39](#)
 - libteep.h, [44](#)
- teep_message_encoder_open_ta_list
 - libteep.c, [40](#)
 - libteep.h, [44](#)
- teep_message_type
 - libteep.h, [42](#)
- teep_message_unwrap.ta.image
 - ta-store.c, [47](#)
- TEEP_OPTION_CHALLENGE
 - libteep.h, [42](#)
- TEEP_OPTION_COMPONENT_ID
 - libteep.h, [43](#)
- TEEP_OPTION_ERR_MSG
 - libteep.h, [43](#)
- TEEP_OPTION_EVIDENCE
 - libteep.h, [42](#)
- TEEP_OPTION_EVIDENCE_FORMAT
 - libteep.h, [43](#)
- TEEP_OPTION_EXT_LIST
 - libteep.h, [42](#)
- TEEP_OPTION_HAVE_BINARY
 - libteep.h, [43](#)
- teep_option_key
 - libteep.h, [42](#)
- TEEP_OPTION_MANIFEST_LIST
 - libteep.h, [42](#)
- TEEP_OPTION_MSG
 - libteep.h, [43](#)
- TEEP_OPTION_OCSP_DATA
 - libteep.h, [42](#)
- TEEP_OPTION_REQUESTED_TC_LIST
 - libteep.h, [43](#)
- TEEP_OPTION_SELECTED_CIPHER_SUIT
 - libteep.h, [42](#)
- TEEP_OPTION_SELECTED_VERSION
 - libteep.h, [42](#)
- TEEP_OPTION_SUIT_REPORTS
 - libteep.h, [43](#)
- TEEP_OPTION_SUPPORTED_CIPHER_SUITS
 - libteep.h, [42](#)
- TEEP_OPTION_TC_LIST
 - libteep.h, [42](#)
- TEEP_OPTION_TC_MANIFEST_SEQUENCE_NUMBER
 - libteep.h, [43](#)
- TEEP_OPTION_UNNEEDED_TC_LIST
 - libteep.h, [43](#)
- TEEP_OPTION_VERSIONS
 - libteep.h, [42](#)
- TEEP_QUERY_REQUEST
 - libteep.h, [42](#)
- TEEP_QUERY_RESPONSE
 - libteep.h, [42](#)
- TEEP_SUCCESS
 - libteep.h, [42](#)
- teep_success
 - teep_message, [20](#)
- teep_suite
 - libteep.h, [43](#)
- teep_tc_info, [22](#)
 - component_id, [22](#)
 - have_binary, [22](#)
 - tc_manifest_sequence_number, [22](#)
- teep_tc_info_array, [23](#)
 - array, [23](#)
 - have_value, [23](#)
 - len, [23](#)
- teep_uint32_array, [24](#)
 - array, [24](#)
 - have_value, [24](#)
 - len, [24](#)
- teep_uint32_option, [24](#)
 - have_value, [24](#)
 - value, [24](#)
- temp_buf
 - ta-store.c, [47](#)
- TEMP_BUF_SIZE
 - ta-store.c, [45](#)
- time
 - tools.c, [58](#)
- token
 - teep_agent_session, [16](#)
 - teep_message, [20](#)
- tools.c
 - _strlen, [56](#)
 - atoi, [56](#)
 - gettimeofday, [56](#)
 - strchr, [56](#)
 - strcpy, [57](#)
 - strdup, [57](#)
 - strncat, [57](#)
 - strncpy, [57](#)
 - strrchr, [58](#)
 - time, [58](#)
- TR_FAIL_CLOSED

- http-lws.c, 71
- TR_FAIL_CONN_ERR
 - http-lws.c, 71
- TR_FAIL_OVERSIZE
 - http-lws.c, 71
- TR_FAIL_REFUSED
 - http-lws.c, 71
- TR_FAIL_START
 - http-lws.c, 71
- TR_OKAY
 - http-lws.c, 71
- TR_ONGOING
 - http-lws.c, 71
- type
 - teep_message, 21
- unnneeded_tc_list
 - teep_message, 21
- uri
 - ta_manifest, 14
 - teep-broker.c, 81
- usage
 - teep-broker.c, 80
- user.ta_header_defines.h
 - TA_CURRENT_TA_EXT_PROPERTIES, 35, 36
 - TA_DATA_SIZE, 35, 36
 - TA_FLAGS, 35, 36
 - TA_STACK_SIZE, 35, 36
 - TA_UUID, 35, 36
- uuid
 - optee-main.c, 32
- uuid_aist_otrp.ta
 - teep-broker.c, 81
- value
 - teep_uint32_option, 24
- versions
 - teep_message, 21
- vsnprintf
 - vsnprintf.c, 69
- vsnprintf.c
 - _atoi, 61
 - _ftoa, 62
 - _is_digit, 62
 - _ntoa_format, 63
 - _ntoa_long, 63
 - _ntoa_long_long, 64
 - _out_buffer, 65
 - _out_char, 65
 - _out_fct, 66
 - _out_null, 66
 - _putchar, 59
 - _strlen, 66
 - _vsnprintf, 67
 - fctprintf, 67
 - FLAGS_CHAR, 60
 - FLAGS_HASH, 60
 - FLAGS_LEFT, 60
 - FLAGS_LONG, 60
 - FLAGS_LONG_LONG, 60
 - FLAGS_PLUS, 60
 - FLAGS_PRECISION, 60
 - FLAGS_SHORT, 60
 - FLAGS_SPACE, 60
 - FLAGS_UPPERCASE, 60
 - FLAGS_ZEROPAD, 60
 - out_fct_type, 61
 - PRINTF_FTOA_BUFFER_SIZE, 61
 - PRINTF_NTOA_BUFFER_SIZE, 61
 - PRINTF_SUPPORT_FLOAT, 61
 - PRINTF_SUPPORT_LONG_LONG, 61
 - PRINTF_SUPPORT_PTRDIFF_T, 61
 - putchar, 68
 - snprintf, 68
 - sprintf, 68
 - vsnprintf, 69
- write_invoke_param
 - CommandQueue, 12
- writeback_params
 - teec-pc.c, 78
- wsa
 - libteep_async, 13