Commissioning Matter Devices

Last edited by Becker Lennart 3 weeks ago

How to install

Install using pip

esp-matter-mfg-tool can be installed using package installer for Python

```
python3 -m pip install esp-matter-mfg-tool
```

Optional dependency

If you want to use the esp-matter-mfg-tool for signing DAC by providing PAI or PAA certificate and key, then chip-cert is required.

CHIP Certificate Tool, chip-cert, provides command line interface (CLI) utility used for generating and manipulating CHIP certificates and CHIP private keys.

If you have already setup the esp-matter and ran install.sh and sourced export.sh then you can skip building and adding tools to path steps. Build chip-cert

```
cd $ESP_MATTER_PATH/connectedhomeip/connectedhomeip
source scripts/activate.sh
gn gen out/host
ninja -C out/host chip-cert
```

Above commands will generate chip-cert at esp-matter/connectedhomeip/connectedhomeip/out/host. Add the tools path to $PATH

```
export PATH="$PATH:$ESP_MATTER_PATH/connectedhomeip/connectedhomeip/out/host"
```

Configure your app

Open the project configuration menu using -

```
cd <your_app>
idf.py menuconfig
```

In the configuration menu, set the following additional configuration to use custom factory partition and different values for Data and Device Info Providers.

`Enable ESP32 Factory Data Provider [Component config → CHIP Device Layer → Commissioning options → Use ESP32 Factory Data Provider]`

Enable config option CONFIG_ENABLE_ESP32_FACTORY_DATA_PROVIDER to use ESP32 specific implementation of CommissionableDataProvider and DeviceAttestationCredentialsProvider.

`Enable ESP32 Device Instance Info Provider [Component config → CHIP Device Layer → Commissioning options → Use ESP32 Device Instance Info Provider]`

Enable config option ENABLE_ESP32_DEVICE_INSTANCE_INFO_PROVIDER to get device instance info from factory partition.

`Enable Attestation - Factory [ Component config → ESP Matter → DAC Provider options → Attestation - Factory]`

Enable config option CONFIG_FACTORY_PARTITION_DAC_PROVIDER to use DAC certificates from the factory partition during

`Set chip-factory namespace partition label [Component config → CHIP Device Layer → Matter Manufacturing Options → chip-factory namespace partition label]`

Set config option CHIP_FACTORY_NAMESPACE_PARTITION_LABEL to choose the label of the partition to store key-values in the "chip-factory" namespace. The default chosen partition label is nvs.

`Change Vendor ID & Device Product ID [Component config → CHIP Device Layer → Device Identification Options]`

- Vendor ID 0xFFF2
- Device ID 0x8001

NOTE that those values have to match the values used for generating the factory partition with `esp-matter-mfg-tool`

Usage examples

Export the Matter SDK path to simplify the certificate and key paths in `~/.bashrc`.

```
export MATTER_SDK_PATH=$ESP_MATTER_PATH/connectedhomeip/connectedhomeip
```

Generate a factory partition

```
esp-matter-mfg-tool -cn "My bulb" -v 0xFFF2 -p 0x8001 --pai \
    -k $MATTER_SDK_PATH/credentials/test/attestation/Chip-Test-PAI-FFF2-8001-Key.pem \
    -c $MATTER_SDK_PATH/credentials/test/attestation/Chip-Test-PAI-FFF2-8001-Cert.pem \
    -cd $MATTER_SDK_PATH/credentials/test/certification-declaration/Chip-Test-CD-FFF2-8001.der
```

Flashing the manufacturing binary

NOTE: Before flashing the manufacturing binary images, flash the erase the flash and flash the firmware

1. `idf.py erase-flash`
2. `idf.py flash -p /dev/ttyACM0`

Flashing a binary image to the device

Please note that esp-matter-mfg-tool only generates manufacturing binary images which need to be flashed onto device using esptool.py.

```
esptool.py -p <serial_port> write_flash <address> path/to/<uuid>-partition.bin
```

NOTE: First flash your app firmware and then followed by the custom partition binary on the device. Please flash the manufacturing binary at the corresponding address of the configured factory partition set by CHIP_FACTORY_NAMESPACE_PARTITION_LABEL which by default is nvs.

- 'serial_port' is most likely /dev/ttyACM0 or /dev/TTYUSB0
- 'address' is most likely 0x10000 if the partition label was not changed. I don't know how to get this value though so best is you just leave the partition label as is
- 'path/to/uuid-partition.bin' - The partition bin is stored somewhere in the out folder. just fuzzy find it.