



*Full credit is given to the above companies including the OS that this PDF file was generated!*

## **Red Hat Enterprise Linux Release 9.2 Manual Pages on 'tpm2\_readpublic.1' command**

### **\$ man tpm2\_readpublic.1**

tpm2\_readpublic(1)      General Commands Manual      tpm2\_readpublic(1)

#### NAME

tpm2\_readpublic(1) - Read the public area of a loaded object.

#### SYNOPSIS

tpm2\_readpublic [OPTIONS]

#### DESCRIPTION

tpm2\_readpublic(1) - Reads the public area of a loaded object.

#### OPTIONS

? -c, --object-context=OBJECT:

Context object for the object to read.

? -n, --name=FILE:

An optional file to save the name structure of the object.

? -f, --format:

Format selection for the public key output file. `tss' (the default)

will output a binary blob according to the TPM 2.0 Specification.

`pem' will output an OpenSSL compatible PEM encoded public key.

`der' will output an OpenSSL compatible DER encoded public key.

`tpmt' will output a binary blob of the TPMT\_PUBLIC struct referenced by TPM 2.0 specs.

Public key format.

? -o, --output=FILE:

The output file path, recording the public portion of the object.

? -t, --serialized-handle=HANDLE:

If the object to be read is a persistent object specified by a raw handle, optionally save the serialized handle for use later. This routine does NOT verify the name of the object being read. Callers should ensure that the contents of name match the expected objects name.

? -q, --qualified-name=FILE:

Saves the qualified name of the object to FILE. The qualified name of the object is the name algorithm hash of the parents qualified and the objects name. Thus the qualified name of the object serves as proof of the objects parents.

## References

### Context Object Format

The type of a context object, whether it is a handle or file name, is determined according to the following logic in-order:

? If the argument is a file path, then the file is loaded as a restored TPM transient object.

? If the argument is a prefix match on one of:

? owner: the owner hierarchy

? platform: the platform hierarchy

? endorsement: the endorsement hierarchy

? lockout: the lockout control persistent object

? If the argument can be loaded as a number it will be treated as a handle, e.g. 0x81010013 and used directly.\_OBJECT\_.

### COMMON OPTIONS

This collection of options are common to many programs and provide information that many users may expect.

? -h, --help=[man|no-man]: Display the tools manpage. By default, it attempts to invoke the manpager for the tool, however, on failure will output a short tool summary. This is the same behavior if the ?man? option argument is specified, however if explicit ?man? is requested, the tool will provide errors from man on stderr. If the ?no-man? option is specified, or the manpager fails, the short options will be output to stdout.

To successfully use the manpages feature requires the manpages to be installed or on MANPATH, See man(1) for more details.

? -v, --version: Display version information for this tool, supported tctis and exit.

? -V, --verbose: Increase the information that the tool prints to the console during its execution. When using this option the file and line number are printed.

? -Q, --quiet: Silence normal tool output to stdout.

? -Z, --enable-errata: Enable the application of errata fixups. Useful if an errata fixup needs to be applied to commands sent to the TPM. Defining the environment TPM2TOOLS\_ENABLE\_ERRATA is equivalent. Information many users may expect.

## TCTI Configuration

The TCTI or ?Transmission Interface? is the communication mechanism with the TPM. TCTIs can be changed for communication with TPMs across different mediums.

To control the TCTI, the tools respect:

1. The command line option -T or --tcti
2. The environment variable: TPM2TOOLS\_TCTI.

Note: The command line option always overrides the environment variable.

The current known TCTIs are:

? tabrmd - The resource manager, called tabrmd (<https://github.com/tpm2-software/tpm2-abrmd>). Note that tabrmd and abrmd as a tcti name are synonymous.

? mssim - Typically used for communicating to the TPM software simulator.

? device - Used when talking directly to a TPM device file.

? none - Do not initialize a connection with the TPM. Some tools allow for off-tpm options and thus support not using a TCTI. Tools that do not support it will error when attempted to be used without a TCTI connection. Does not support ANY options and MUST BE presented as the exact text of ?none?.

The arguments to either the command line option or the environment variable are in the form:

<tcti-name>:<tcti-option-config>

Specifying an empty string for either the <tcti-name> or <tcti-option-config> results in the default being used for that portion respectively.

## TCTI Defaults

When a TCTI is not specified, the default TCTI is searched for using `dlopen(3)` semantics. The tools will search for `tabrmd`, `device` and `mssim` TCTIs IN THAT ORDER and USE THE FIRST ONE FOUND. You can query what TCTI will be chosen as the default by using the `-v` option to print the version information. The `?default-tcti?` key-value pair will indicate which of the aforementioned TCTIs is the default.

## Custom TCTIs

Any TCTI that implements the dynamic TCTI interface can be loaded. The tools internally use `dlopen(3)`, and the raw `tcti-name` value is used for the lookup. Thus, this could be a path to the shared library, or a library name as understood by `dlopen(3)` semantics.

## TCTI OPTIONS

This collection of options are used to configure the various known TCTI modules available:

? `device`: For the `device` TCTI, the TPM character device file for use by the `device` TCTI can be specified. The default is `/dev/tpm0`.

Example: `-T device:/dev/tpm0` or `export TPM2TOOLS_TCTI=?device:/dev/tpm0?`

? `mssim`: For the `mssim` TCTI, the domain name or IP address and port number used by the simulator can be specified. The default are `127.0.0.1` and `2321`.

Example: `-T mssim:host=localhost,port=2321` or `export TPM2TOOLS_TCTI=?mssim:host=localhost,port=2321?`

? `abrmd`: For the `abrmd` TCTI, the configuration string format is a series of simple key value pairs separated by a ``,'` character. Each key and value string are separated by a ``='` character.

? TCTI abrmd supports two keys:

1. `bus\_name` : The name of the tabrmd service on the bus (a string).
2. `bus\_type` : The type of the dbus instance (a string) limited to `session` and `system`.

Specify the tabrmd tcti name and a config string of bus\_name=com.ex?

ample.FooBar:

```
\--tcti=tabrmd:bus_name=com.example.FooBar
```

Specify the default (abrmd) tcti and a config string of bus\_type=ses?

sion:

```
\--tcti:bus_type=session
```

NOTE: abrmd and tabrmd are synonymous. the various known TCTI mod?

ules. # EXAMPLES

Create a primary object and read the public structure in an openssl compli?

ant format

```
tpm2_createprimary -c primary.ctx
```

```
tpm2_readpublic -c primary.ctx -o output.dat -f pem
```

Serialize an existing persistent object handle to disk for later use

This work-flow is primarily intended for existing persistent TPM ob?

jects. This work-flow does not verify that the name of the serialized

object matches the expected, and thus the serialized handle could be

pointing to an attacker controlled object if no verification is done.

If you are creating an object from scratch, save the serialized handle

when making the object persistent.

We assume that an object has already been persisted, for example via:

```
# We assume that an object has already been persisted, for example
```

```
tpm2_createprimary -c primary.ctx
```

```
# context files have all the information for the TPM to verify the object
```

```
tpm2_evictcontrol -c primary.ctx
```

```
persistent-handle: 0x81000001
```

```
action: persisted
```

Next use the persistent handle to get a serialized handle:

```
# The persistent handle output could be at an attacker controlled object,
```

```
# best practice is to use the option "-o:" for tpm2_evictcontrol to get a
# serialized handle instead.

tpm2_readpublic -c 0x81000001 -o output.dat -f pem -t primary.handle

# use this verified handle in an encrypted session with the tpm

tpm2_startauthsession --policy-session -S session.ctx -c primary.handle
```

For new objects, its best to use all serialized handles.

## Returns

Tools can return any of the following codes:

- ? 0 - Success.
- ? 1 - General non-specific error.
- ? 2 - Options handling error.
- ? 3 - Authentication error.
- ? 4 - TCTI related error.
- ? 5 - Non supported scheme. Applicable to tpm2\_testparams.

## BUGS

Github Issues (<https://github.com/tpm2-software/tpm2-tools/issues>)

## HELP

See the Mailing List (<https://lists.01.org/mailman/listinfo/tpm2>)

tpm2-tools

tpm2\_readpublic(1)