

# Generate Fingerprint for File

---

## OPERATING INSTRUCTIONS

**Copyright**

© Ericsson AB 2014–2017. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

**Disclaimer**

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

**Trademark List**

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



# Contents

<b>1</b>	<b>Description</b>	<b>1</b>
<b>2</b>	<b>Procedure</b>	<b>1</b>
2.1	Generate Fingerprint for A File	1



Generate Fingerprint for File



# 1 Description

This instruction describes how to generate a fingerprint for a file.

Fingerprints, also known as digests, are calculated on the entire provided Certificate Management file.

## 2 Procedure

### 2.1 Generate Fingerprint for A File

#### Prerequisites

- No documents are required.
- No tools are required.
- The following conditions must apply:
  - The user has the System Security Administrator role.
  - An Ericsson Command-Line Interface (ECLI) session in Exec mode is in progress.

#### Steps

1. Navigate to the *CertMCapabilities* managed object:

```
>dn ManagedElement=NODE06ST,SystemFunctions=1,SecM=1,CertM=1,CertMCapabilities=1
```

2. Show attribute `fingerprintSupport`:

```
(CertMCapabilities=1)>show fingerprintSupport
```

The output shows the algorithm used for calculating the fingerprint, for example:

```
fingerprintSupport=SHA_224
```

Here, `SHA_224` denotes the algorithm in use.



**Note:** SHA\_224 corresponds to the `openssl` command-line option `-sha224` in the next step.

3. Using a command shell, manually generate the fingerprint for a file using the command `openssl` with the algorithm specified in attribute `fingerprintSupport` as an input option. For example:

```
shell$ openssl dgst -c -hex -sha224 node06stNodeCredential1.p12
```

Here, the algorithm used is `-sha224`.

The output reflects the algorithm used, for example:

```
SHA224 (node06stNodeCredential1.p12)= ba:41:ac:4f:b3:00:10:98:28:47:36:b1:eb:d9:66:33:69:05:7d:c2
```

**Note:** The extra space character between `=` and fingerprint `ba:41:ac:4f:b3:00:10:98:28:47:36:b1:eb:d9:66:33:69:05:7d:c2` does not belong to the fingerprint itself.