

# Android用 .apk形式編

| 改版履歴  |            |                           |     |
|-------|------------|---------------------------|-----|
| 版数    | 日付         | 内容                        | 担当  |
| V.1.0 | 2015/4/1   | 初版                        | NII |
| V.1.1 | 2017/2/28  | JKSコマンドの修正<br>署名確認作業の補足追加 | NII |
| V.2.0 | 2018/2/26  | 動作環境を以下に変更<br>Windows10   | NII |
| V.2.1 | 2018/7/9   | タイムスタンプ利用手順の追加            | NII |
| V.2.2 | 2020/6/4   | 中間CA証明書のURLとリポジトリのURLの変更  | NII |
| V.2.3 | 2020/12/22 | 中間CA証明書のURL変更             | NII |
| V.2.4 | 2020/12/24 | 鍵長の変更                     | NII |
| V.2.5 | 2021/5/31  | コード署名用証明書の中間CA証明書を修正      | NII |
| V.2.6 | 2023/5/19  | コード署名用証明書のCSR作成に関する手順を削除  | NII |

## 目次

### 1. コード署名用証明書の利用

#### 1-1. 前提条件

#### 1-2. JKS (Javaキーストア) ファイルの作成

##### 1-2-1. 事前準備

##### 1-2-2. PKCS#12ファイルの作成

##### 1-2-3. JKS (Javaキーストア) ファイルの作成

#### 1-3. 署名

#### 1-4. コード署名確認作業

## 1. コード署名用証明書の利用

### 1-1. 前提条件

OpenSSLコード署名用証明書を使用する場合の前提条件について記載します。適宜、コード署名用証明書をインストールする利用管理者様の環境により、読み替えをお願いします。

コマンドプロンプト上で実行するコマンドは、「>」にて示しています。

| 前提条件                                                                           |
|--------------------------------------------------------------------------------|
| 1. OpenSSLがインストールされていること<br>2. Java SE Development Kit（以下JDKと呼ぶ）がインストールされていること |

### 1-2. JKS (Javaキーストア) ファイルの作成

本章ではJKS(Javaキーストア)ファイルの作成方法について記述します。

#### 1-2-1. 事前準備

事前準備として、「ルートCA証明書」、「中間CA証明書」、「コード署名用証明書」を取得してください。

| 事前準備 |
|------|
|------|

1. 「証明書の申請から取得まで」で受領したコード署名用証明書を任意の名前で任意の場所に保存してください。
2. 「ルートCA証明書」と「中間CA証明書」を準備し、この2つを連結させます。下記URLより、リポジトリへアクセスしてください。

「中間CA証明書」を下記リポジトリより取得してください。  
セコムパスポート for Member 2.0 PUB リポジトリ：  
<https://repo1.secomtrust.net/spcpp/pfm20pub/index.html>

【2021年5月31日00:00以前の発行証明書が対象】  
リポジトリ内にある「証明書の種類」より中間CA証明書を取得してください。  
  
<https://repo1.secomtrust.net/spcpp/pfm20pub/codecag2/CODECAG2.cer>

次に、「ルートCA証明書」を下記リポジトリより取得してください。  
Security Communication RootCA2 リポジトリ：  
<https://repository.secomtrust.net/SC-Root2/index.html>

Security Communication RootCA2 証明書：  
<https://repository.secomtrust.net/SC-Root2/SCRoot2ca.cer>

【2021年5月31日00:00以後の発行証明書が対象】  
リポジトリ内にある「証明書の種類」より中間CA証明書を取得してください。  
  
<https://repo1.secomtrust.net/spcpp/pfm20pub/codecag2/CODECAG2SCROOTCA3.cer>

次に、「ルートCA証明書」を下記リポジトリより取得してください。  
Security Communication RootCA3 リポジトリ：  
<https://repository.secomtrust.net/SC-Root3/index.html>

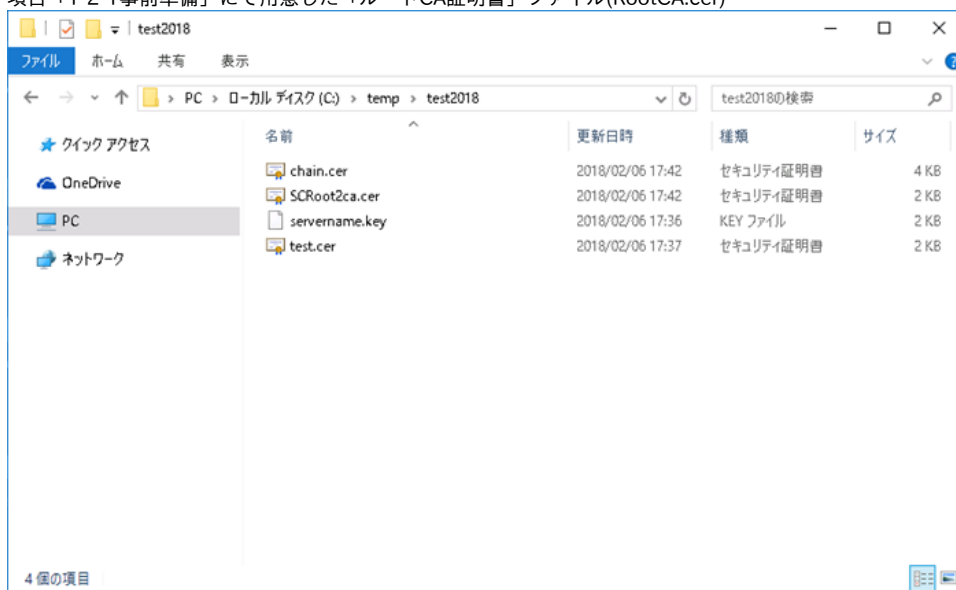
Security Communication RootCA3 証明書：  
<https://repository.secomtrust.net/SC-Root3/SCRoot3ca.cer>

## 1-2-2. PKCS#12ファイルの作成

本項目ではWindowsOS上で任意のフォルダにPKCS#12ファイルを生成する方法を記述します。  
以下は、例としてWindows10上での作成方法を記載します。

### PKCS#12ファイルの作成

1. 任意のフォルダ(ここではC:\temp\test2018とします)にて以下の3つのファイルを用意してください。
  - a. 項目「鍵ペアの生成」にて生成した鍵ペアのファイル(servername.key)
  - b. 項目「証明書の申請から取得まで」にて取得したコード署名用証明書(ここではtest.cerとします)
  - c. 項目「1-2-1事前準備」にて用意した「ルートCA証明書」と「中間CA証明書」を連結させたファイル(ここではchain.cerとします)
  - d. 項目「1-2-1事前準備」にて用意した「ルートCA証明書」ファイル(RootCA.cer)



2. CAfile に指定する証明書をDER形式からPEM形式に変換します。

```
・ Security Communication RootCA2の場合
openssl x509 -inform der -in SCRoot2ca.cer -outform pem -out SCRoot2ca.cer

・ 中間CA証明書（2021年5月31日00:00以前の発行証明書が対象）の場合
openssl x509 -inform der -in CODECAG2.cer -outform pem -out CODECAG2.cer
```

3. コマンドプロンプト上にて上記で取得した「ルートCA証明書」と「中間CA証明書」を下記のコマンドにより、連結させてください。中間CA証明書の下部にルートCA証明書が併記されるファイルとなります。

```
> type (中間CA証明書のパス) (ルートCA証明書のパス) > (出力するファイル名)
```

4. 連結したファイルがPEM形式になっていることを確認してください。  
例) PEM形式の証明書

```
-----BEGIN CERTIFICATE-----

MIIEcTCCA1mgAwIBAgIIasWHLdnQB2owDQYJKoZIhvcNAQELBQAwbzELMAkGA1UE
BhMCSIAxFAQSABgNVBAcMC0FjYWRIbWUtb3BzMSowKAYDVQQKDCCFOYXRpb25hbCBJ
bnN0aXR1dGUgb2YgSW5mb3JtYXRpY3MxHjAcBgNVBAMMFU5JSSBPcGVyYXRpbmcg
Q0EgLSBHMjAeFw0xNTAzMTIwMTA4MDJaFw0xNzA0MTEwMTA4MDJaMHAAcCzAIBgNV
(中略)

LmeW0e/xkkxwdmKv5y5txLIFcp53AZI/vjn3BHp42PFkkTISEmAUiCtQ2A25QDRR
RG33laC8E8TI/SnOA8h95XQtGWm47PrIjXytleOrFousbploW8MZw4gDXVQ3485
XEftqwwIMcLNxttJ6i6f9XVyPMRhHy9rdDPseHiXayxcBxJMuw==

-----END CERTIFICATE-----
```

5. コマンドプロンプトを開き、ファイルのある任意のフォルダ(ここではC:\temp\test2015)へ移動します。

```
> set Path=(OpenSSLインストールディレクトリ)\bin
※OpenSSLインストールディレクトリをプログラムを探すディレクトリに指定します
> cd (作業ディレクトリ) ←作業ディレクトリ
```

6. 移動後、下記のコマンドを実行しPKCS#12ファイルを生成してください。

> openssl pkcs12 -export -chain -inkey (鍵ペアのファイル名) -CAfile (ルートCA証明書と中間CA証明書を連結させたファイル) -in (コード署名用の証明書ファイル名) -out (PKCS#12形式で出力するファイル名) -name (コード署名用証明書のエイリアス名) -caname (ルートCA証明書と中間CA証明書のエイリアス名)

```
選択C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\main>set Path="C:\Program Files (x86)\OpenSSL\bin";"C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\main>cd C:\temp\test2018
C:\temp\test2018>openssl pkcs12 -export -chain -inkey servername.key -CAfile chain.cer -in test.cer -out test.p12 -name test -caname root
```

7. 「Enter pass phrase for (鍵ペアファイル):」と表示されますので、鍵ペアファイルにアクセスさせるための、パスフレーズを入力してください。

```
選択C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\main>set Path="C:\Program Files (x86)\OpenSSL\bin";"C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\main>cd C:\temp\test2018
C:\temp\test2018>openssl pkcs12 -export -chain -inkey servername.key -CAfile chain.cer -in test.cer -out test.p12 -name test -caname root
Loading 'screen' into random state - done
Enter pass phrase for servername.key:
```

8. 「Enter Export Password:」と表示されますので、PKCS#12形式のファイルを保護するためのアクセスPINとして任意の文字列を入力してください。

```
選択C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

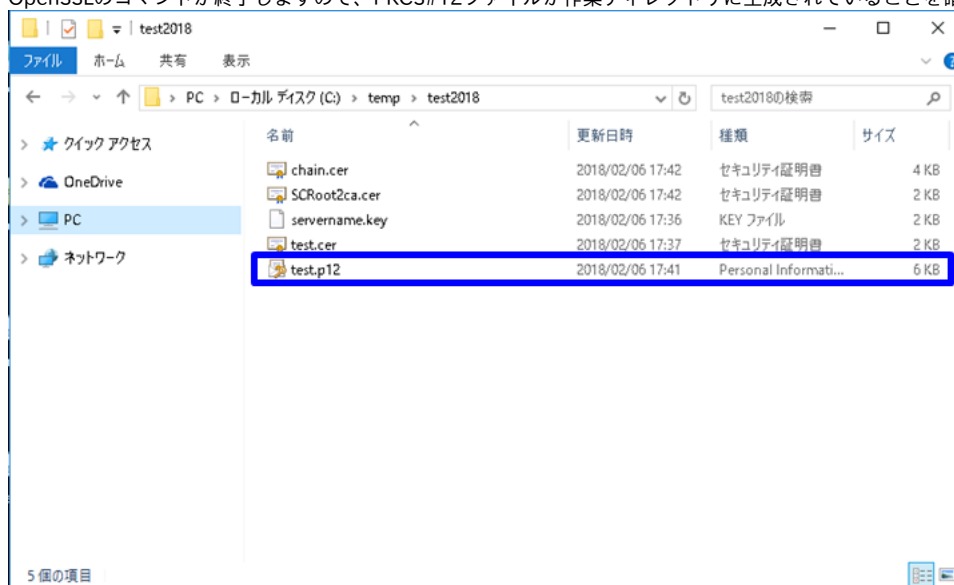
C:\Users\main>set Path="C:\Program Files (x86)\OpenSSL\bin";"C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\main>cd C:\temp\test2018
C:\temp\test2018>openssl pkcs12 -export -chain -inkey servername.key -CAfile chain.cer -in test.cer -out test.p12 -name test -caname root
Loading 'screen' into random state - done
Enter pass phrase for servername.key:
Enter Export Password:
```

9. 「Verifying - Enter Export Password:」と表示されますので、確認のため、同じアクセスPINを再入力してください。

```
選択C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\main>set Path="C:\Program Files (x86)\OpenSSL\bin";"C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\main>cd C:\temp\test2018
C:\temp\test2018>openssl pkcs12 -export -chain -inkey servername.key -CAfile chain.cer -in test.cer -out test.p12 -name test -caname root
Loading 'screen' into random state - done
Enter pass phrase for servername.key:
Enter Export Password:
Verifying - Enter Export Password:
```

10. OpenSSLのコマンドが終了しますので、PKCS#12ファイルが作業ディレクトリに生成されていることを確認してください。

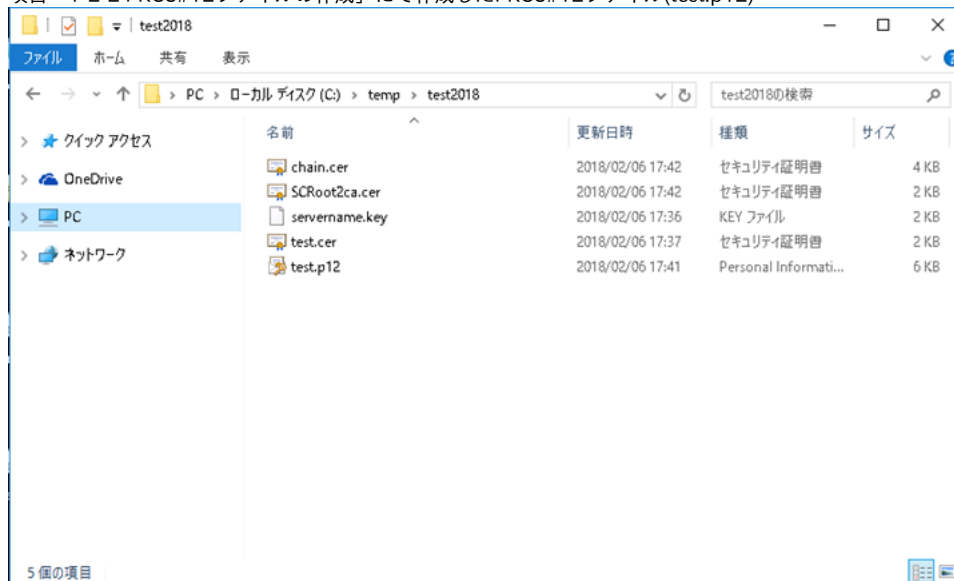


### 1-2-3. JKS (Javaキーストア) ファイルの作成

本項目ではWindowsOS上で任意のフォルダにJKS (Javaキーストア) ファイルを作成する方法を記述します。以下は、例としてWindows10上での作成方法を記載します。

#### JKS(Javaキーストア)の作成

1. 任意のフォルダ(ここではC:\temp\test2015とします)にて以下のファイルを用意してください。  
項目「1-2-2 PKCS#12ファイルの作成」にて作成したPKCS#12ファイル(test.p12)



2. コマンドプロンプトを開き、ファイルのある任意のフォルダ(ここではC:\temp\test2015)へ移動します。

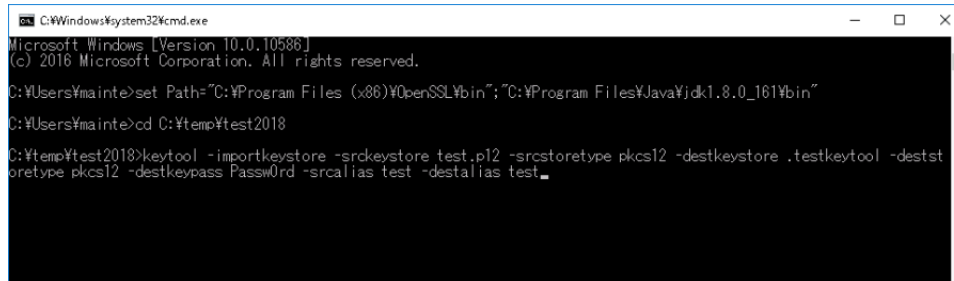
```
> set Path=(JDKインストールディレクトリ)\bin
※JDKインストールディレクトリをプログラムを探すディレクトリに指定します
> cd (作業ディレクトリ) ←作業ディレクトリ
```

移動後、下記のコマンドを入力しJKS (Javaキーストア) ファイルを作成してください。

```
> keytool -importkeystore -srckeystore (PKCS#12ファイル名) -destkeystore (作成したいキーストアファイル名) -srcstoretype pkcs12 -deststoretype pkcs12 -destkeypass " (キーストアに設定したいパスワード) " -srcalias "( PKCS#12ファイルで利用されているエイリアス名) " -destalias " (登録したいエイリアス名) "
```

※ PKCS#12ファイルで利用されているエイリアス名(別名)は以下コマンドでご参照ください。

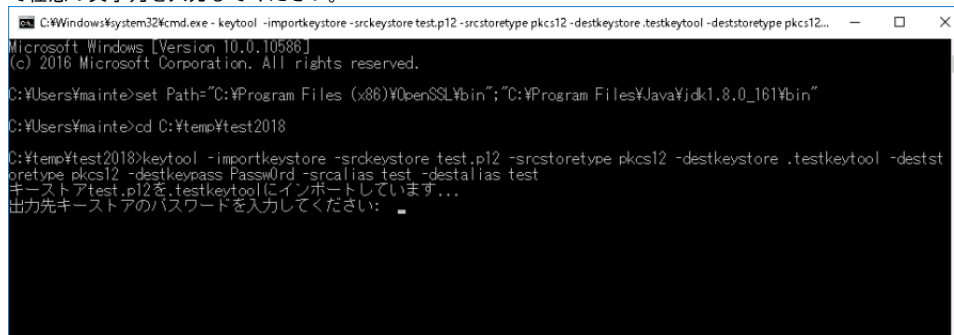
```
keytool -v -list -keystore (PKCS#12ファイル名)
```



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ymainte>set Path="C:\Program Files (x86)\OpenSSL\bin";"C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\ymainte>cd C:\temp\test2018
C:\temp\test2018>keytool -importkeystore -srckeystore test.p12 -srcstoretype pkcs12 -destkeystore .testkeytool -deststoretype pkcs12 -destkeypass Passw0rd -srcalias test -destalias test_
```

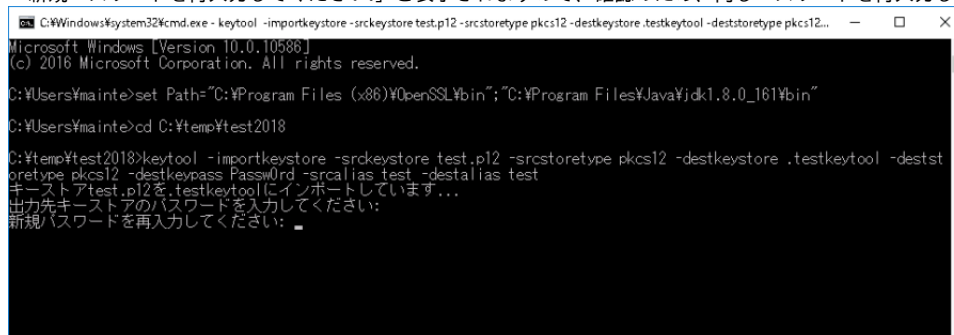
3. 「出力先キーストアのパスワードを入力してください:」と表示されますので、JKS (Javaキーストア) ファイルを保護するためのパスワードとして任意の文字列を入力してください。



```
C:\Windows\system32\cmd.exe - keytool -importkeystore -srckeystore test.p12 -srcstoretype pkcs12 -destkeystore .testkeytool -deststoretype pkcs12...
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ymainte>set Path="C:\Program Files (x86)\OpenSSL\bin";"C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\ymainte>cd C:\temp\test2018
C:\temp\test2018>keytool -importkeystore -srckeystore test.p12 -srcstoretype pkcs12 -destkeystore .testkeytool -deststoretype pkcs12 -destkeypass Passw0rd -srcalias test -destalias test
キーストアtest.p12を.testkeytoolにインポートしています...
出力先キーストアのパスワードを入力してください: _
```

4. 「新規パスワードを再入力してください:」と表示されますので、確認のため、同じパスワードを再入力してください。



```
C:\Windows\system32\cmd.exe - keytool -importkeystore -srckeystore test.p12 -srcstoretype pkcs12 -destkeystore .testkeytool -deststoretype pkcs12...
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ymainte>set Path="C:\Program Files (x86)\OpenSSL\bin";"C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\ymainte>cd C:\temp\test2018
C:\temp\test2018>keytool -importkeystore -srckeystore test.p12 -srcstoretype pkcs12 -destkeystore .testkeytool -deststoretype pkcs12 -destkeypass Passw0rd -srcalias test -destalias test
キーストアtest.p12を.testkeytoolにインポートしています...
出力先キーストアのパスワードを入力してください:
新規パスワードを再入力してください: _
```

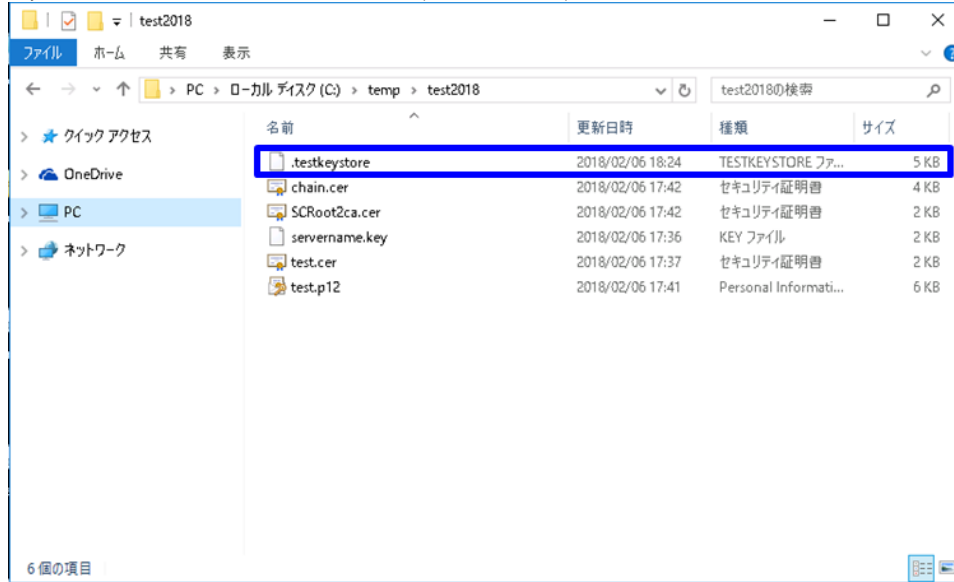
5. 「ソース・キーストアのパスワードを入力してください」と表示されますので、PKCS#12ファイルのアクセスPINを入力してください。

```
C:\Windows\system32\cmd.exe - keytool -importkeystore -srckeystore test.p12 -srcstoretype pkcs12 -destkeystore .testkeytool -deststoretype pkcs12...
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\main>set Path="C:\Program Files (x86)\OpenSSL\bin";"C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\main>cd C:\temp\test2018

C:\temp\test2018>keytool -importkeystore -srckeystore test.p12 -srcstoretype pkcs12 -destkeystore .testkeytool -deststoretype pkcs12 -destkeypass Passw0rd -srcalias test -destalias test
キーストアtest.p12を.testkeytoolにインポートしています...
出力先キーストアのパスワードを入力してください:
新規パスワードを再入力してください:
ソース・キーストアのパスワードを入力してください: 
```

6. keytoolのコマンドが終了しますので、 JKS (Javaキーストア) ファイルが作業ディレクトリに作成されていることを確認してください。



7. 署名検証時に必要となるため下記のコマンドを入力しJKS (Javaキーストア) ファイルに「ルートCA証明書」をインポートしてください。

```
> keytool -importcert -keystore (キーストアファイル名) -alias (設定したいルートCA証明書のエイリアス名) -file (ルートCA証明書のファイル名) -trustcacerts
```

```
C:\Windows\system32\cmd.exe - keytool -importkeystore -srckeystore test.p12 -srcstoretype pkcs12 -destkeystore .testkeytool -deststoretype pkcs12...
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\main>set Path="C:\Program Files (x86)\OpenSSL\bin";"C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\main>cd C:\temp\test2018

C:\temp\test2018>keytool -importkeystore -srckeystore test.p12 -srcstoretype pkcs12 -destkeystore .testkeytool -deststoretype pkcs12 -destkeypass Passw0rd -srcalias test -destalias test
キーストアtest.p12を.testkeytoolにインポートしています...
出力先キーストアのパスワードを入力してください:
新規パスワードを再入力してください:
ソース・キーストアのパスワードを入力してください:

C:\temp\test2018>keytool -importcert -keystore .testkeytool -alias scrootca2 -file SCRoot2ca.cer -trustcacerts
```

8. 「キーストアのパスワードを入力してください:」と表示されますので、JKS (Javaキーストア) ファイルを保護するパスワードを入力してください。

```
C:\Windows\system32\cmd.exe - keytool -importkeystore -srckeystore test.p12 -srcstoretype pkcs12 -destkeystore .testkeytool -deststoretype pkcs12...
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ymainte>set Path="C:\Program Files (x86)\OpenSSL\bin";"C:\Program Files\Java\jdk1.8.0_161\bin"

C:\Users\ymainte>cd C:\temp\test2018

C:\temp\test2018>keytool -importkeystore -srckeystore test.p12 -srcstoretype pkcs12 -destkeystore .testkeytool -deststoretype pkcs12 -destkeypass Passw0rd -srcalias test -destalias test
キーストアtest.p12を.testkeytoolにインポートしています...
出力先キーストアのパスワードを入力してください:
新規パスワードを再入力してください:
ソース・キーストアのパスワードを入力してください:

C:\temp\test2018>keytool -importcert -keystore .testkeytool -alias sctestca2 -file SCRoot2ca.cer -trustcacerts
キーストアのパスワードを入力してください: ー
```

9. 「この証明書を信頼しますか。[いいえ:]」と表示されますので、「はい」と日本語入力で入力してください。

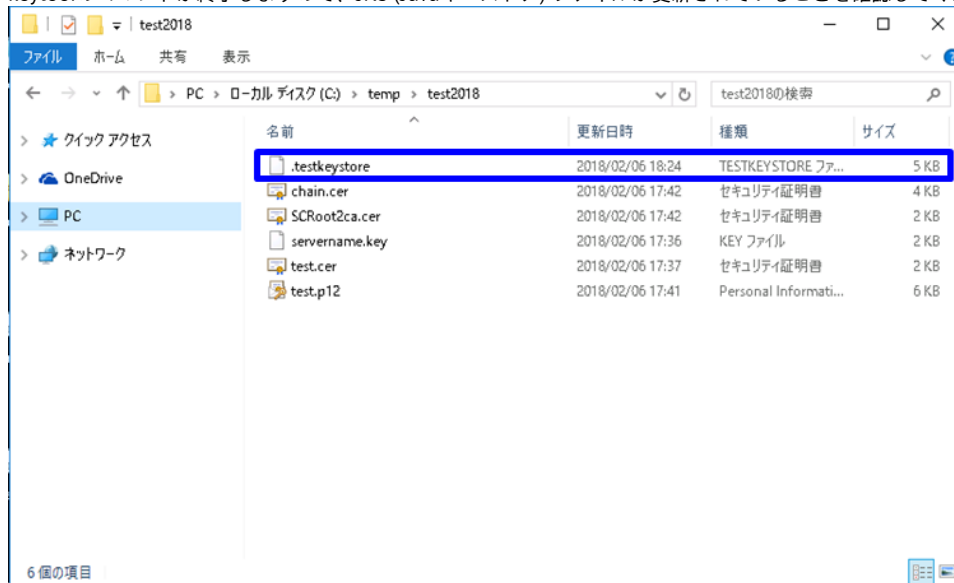
```
C:\Windows\system32\cmd.exe - keytool -importcert -keystore .testkeytool -alias sctestca2 -file SCRoot2ca.cer -trustcacerts

#2: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
  KeyIdentifier [
    0000: 78 A6 55 B4 42 8E FF 65 3E 20 F8 75 65 7A F6 64 ..U.B...e> .uez.d
    0010: A7 FB C7 78 ....x
  ]
]

この証明書を信頼しますか。[いいえ:] (はい)
```

10. keytoolのコマンドが終了しますので、JKS (Javaキーストア) ファイルが更新されていることを確認してください。

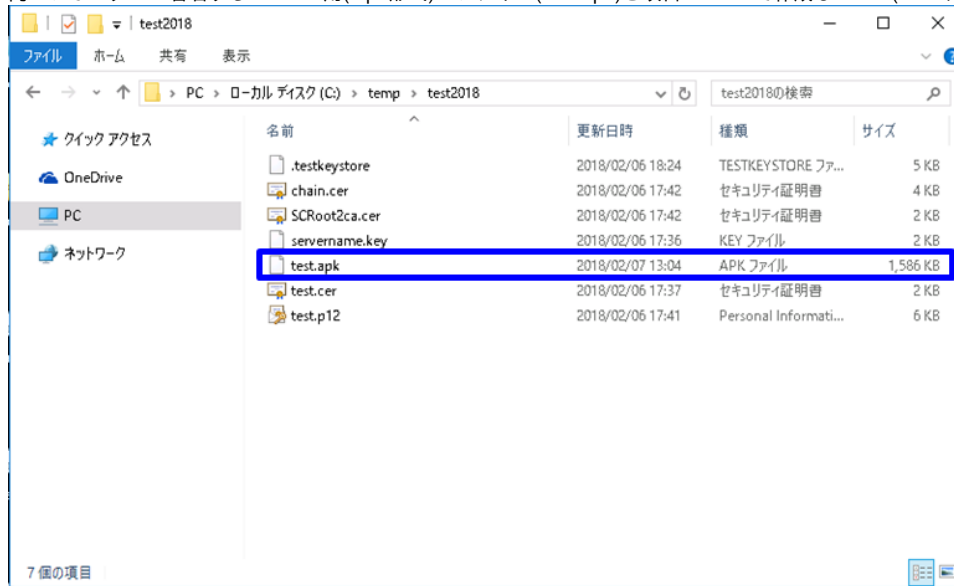


## 1-3. 署名

本章ではAndroid用(.apk形式)のファイルにWindowsOS上にて、デジタル署名をする方法について記述します。併せてタイムスタンプを付与する場合と2通りの手順がありますので適した方をご選択ください。

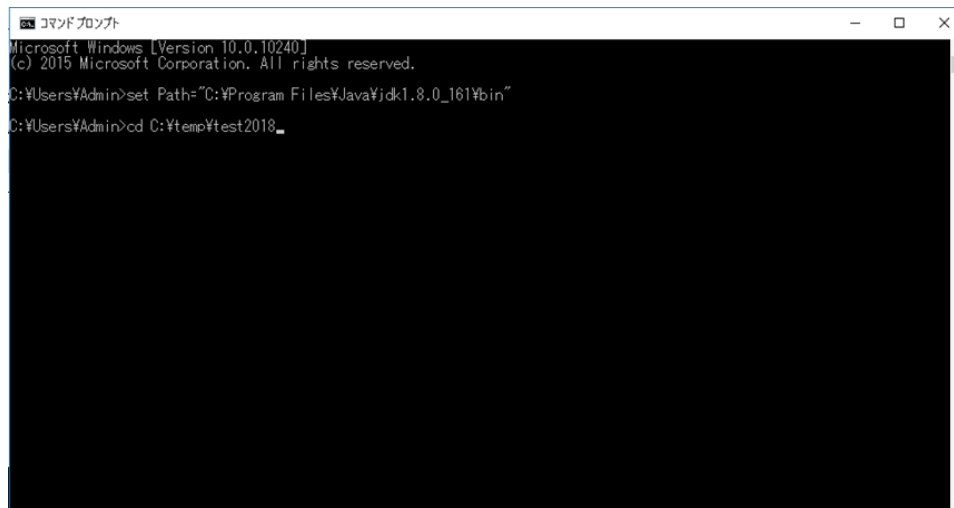
### 署名作業（併せてタイムスタンプを付与しない場合）

1. 同一フォルダ上に署名するAndroid用(.apk形式)のファイル(test.apk)と項目1-2-3にて作成したJKS (Javaキーストア) ファイルを置きます。



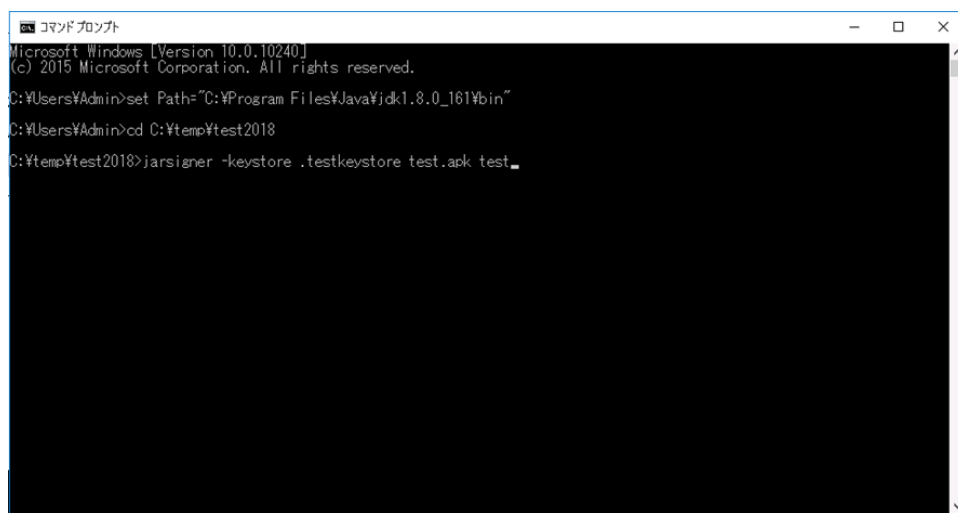
2. コマンドプロンプトを実行し、署名対象ファイルのあるフォルダへ移動します。

```
> set Path=(JDKインストールディレクトリ)\bin
※JDKインストールディレクトリをプログラムを探すディレクトリに指定します
> cd (作業ディレクトリ) ←作業ディレクトリ
```



3. フォルダ移動後、署名したいAndroid用(.apk形式)のファイル(ここではtest.apk)に対して下記のコマンドにて署名を実行してください。

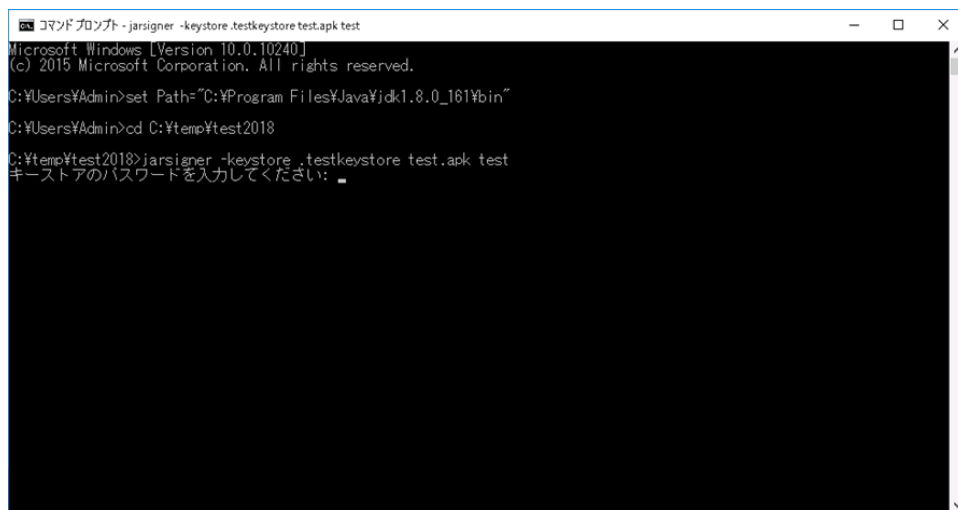
jarsigner -keystore (キーストアファイル名) (署名したいAndroid用(.apk形式)のファイル名) (キーストア内の証明書のエイリアス名)



```
コマンド プロンプト
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Admin>set Path="C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\Admin>cd C:\temp\test2018
C:\temp\test2018>jarsigner -keystore .testkeystore test.apk test_
```

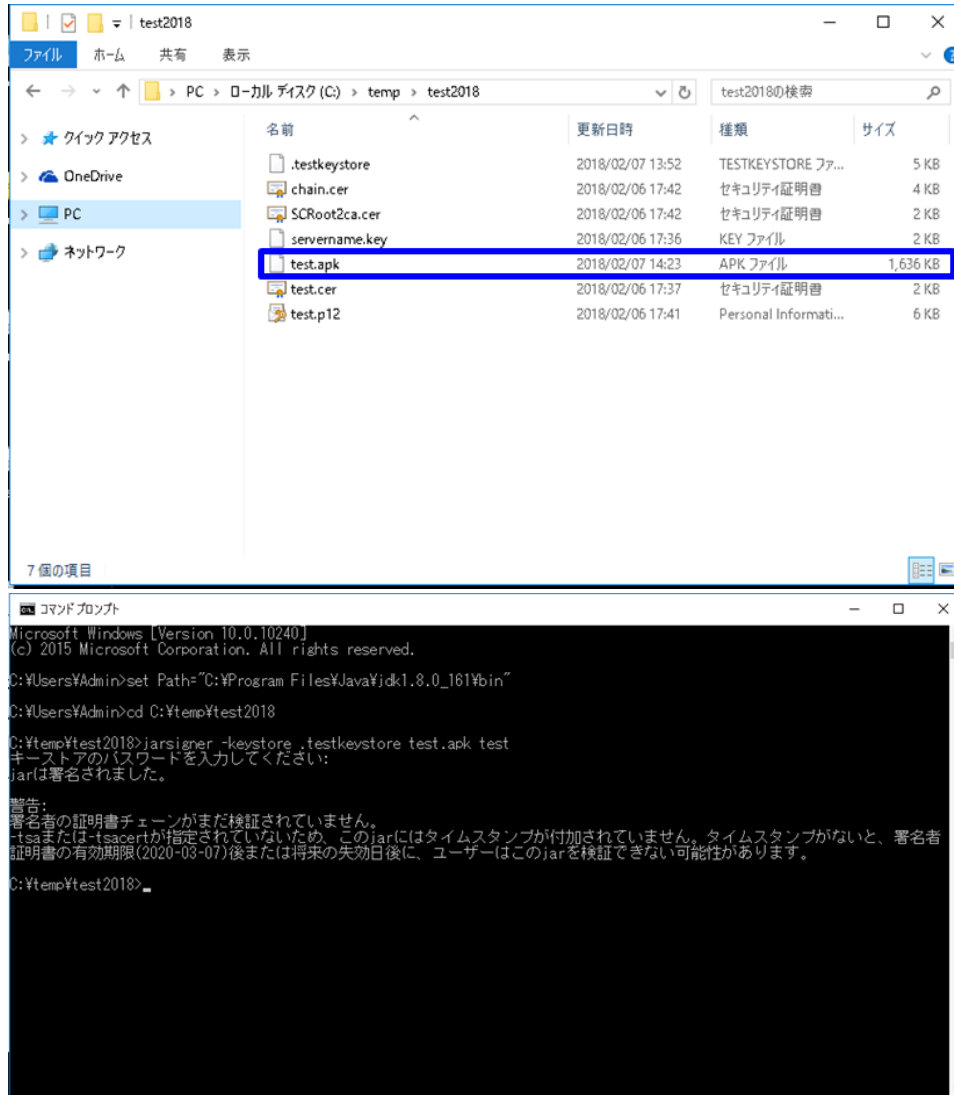
4. 「キーストアのパスワードを入力してください:」と表示されますので、JKS (Javaキーストア) ファイルを保護するパスフレーズを入力してください。



```
コマンド プロンプト - jarsigner -keystore .testkeystore test.apk test
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Admin>set Path="C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\Admin>cd C:\temp\test2018
C:\temp\test2018>jarsigner -keystore .testkeystore test.apk test
キーストアのパスワードを入力してください: _
```

5. 「jarは署名されました。」表示され、keytoolのコマンドが終了しますので、対象のAndroid用(.apk形式)ファイルが更新されていることを確認してください。



※以下警告はタイムスタンプが付与されていないため、表示されているものです。

署名状況には問題ありません。警告内の署名書証明書の有効期限は証明書、証明書毎に異なります。

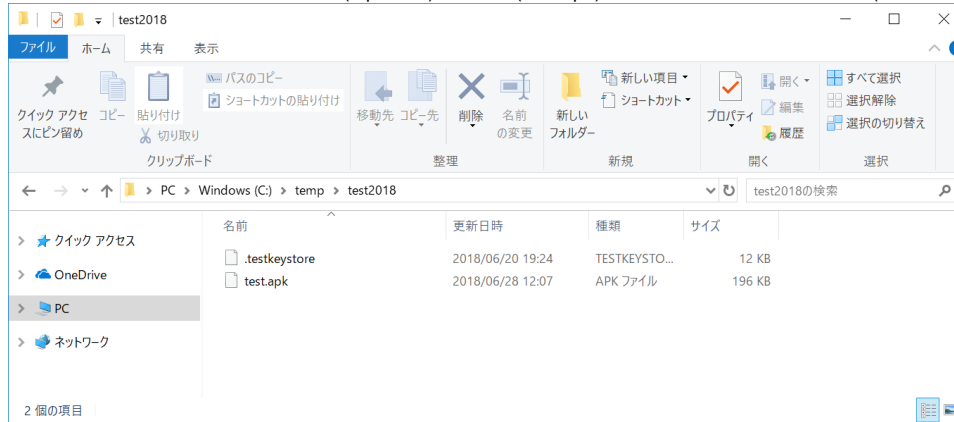
警告：

-tsaまたは-tsacertが指定されていないため、このjarにはタイムスタンプが付加されていません。

タイムスタンプがないと、署名者証明書の有効期限(20YY-MM-)後または将来の失効日後に、ユーザーはこのjarを検証できない可能性があります。

#### 署名作業（併せてタイムスタンプを付与する場合）

1. 同一フォルダ上に署名するAndroid用(.apk形式)ファイル(test.apk)と項目1-2-3にて作成したJKS (Javaキーストア) ファイルを置きます。



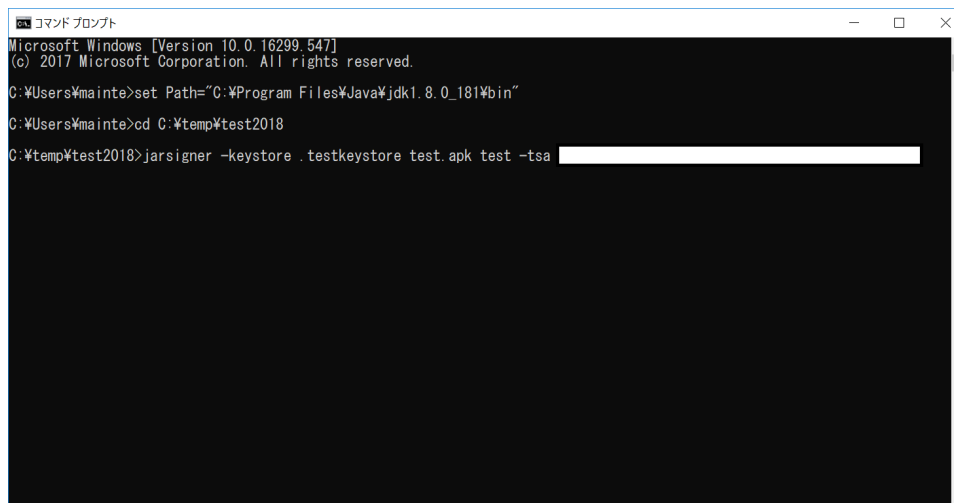
2. コマンドプロンプトを実行し、署名対象ファイルのあるフォルダへ移動します。

```
> set Path=(JDKインストールディレクトリ)\bin
※JDKインストールディレクトリをプログラムを探すディレクトリに指定します
> cd (作業ディレクトリ) ←作業ディレクトリ
```

3. フォルダ移動後、署名したいAndroid用(.apk形式)ファイル(ここではtest.apk)に対して以下のコマンドにて署名を実行してください。

```
jarsigner -keystore (キーストアファイル名) (署名したいAndroid用(.apk形式)のファイル名) (キーストア内の証明書のエイリアス名) -tsa (タイムスタンプURL)
```

※タイムスタンプURLに関しては登録担当者（各利用機関において、証明書発行のための審査と電子証明書自動発行支援システムの操作をする方）



4. 「キーストアのパスワードを入力してください:」と表示されますので、JKS (Javaキーストア) ファイルを保護するパスフレーズを入力してください

```
コマンド プロンプト - jarsigner -keystore .testkeystore test.apk test -tsa
Microsoft Windows [Version 10.0.16299.547]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vmainte>set Path="C:\Program Files\Java\jdk1.8.0_181\bin"

C:\Users\vmainte>cd C:\temp\test2018

C:\temp\test2018>jarsigner -keystore .testkeystore test.apk test -tsa
キーストアのパスワードを入力してください:
```

5. 「jarは署名されました。」表示され、keytoolのコマンドが終了しますので、対象のAndroid用(.apk形式)ファイルが更新されていることを確認して

```
コマンド プロンプト
Microsoft Windows [Version 10.0.16299.547]
(c) 2017 Microsoft Corporation. All rights reserved.

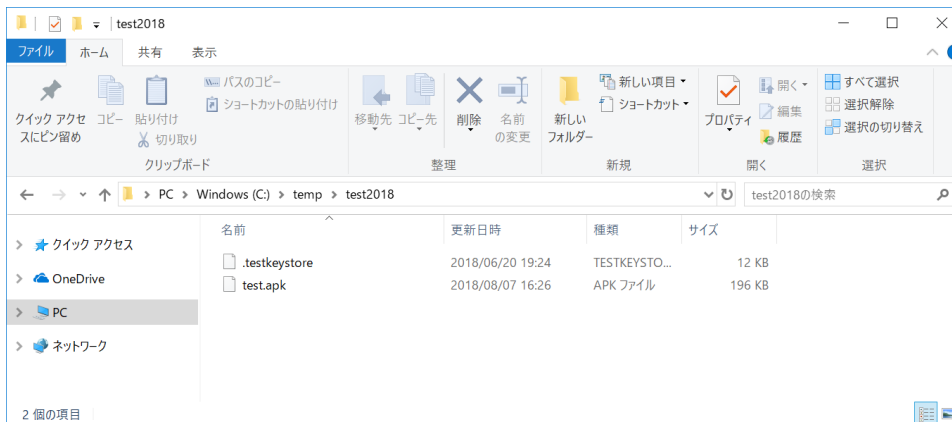
C:\Users\vmainte>set Path="C:\Program Files\Java\jdk1.8.0_181\bin"

C:\Users\vmainte>cd C:\temp\test2018

C:\temp\test2018>jarsigner -keystore .testkeystore test.apk test -tsa
キーストアのパスワードを入力してください:
jarは署名されました。

警告:
署名者の証明書チェーンがまだ検証されていません。

C:\temp\test2018>
```



※以下警告は署名状況には問題ありません。

警告：  
署名者の証明書チェーンがまだ検証されていません。

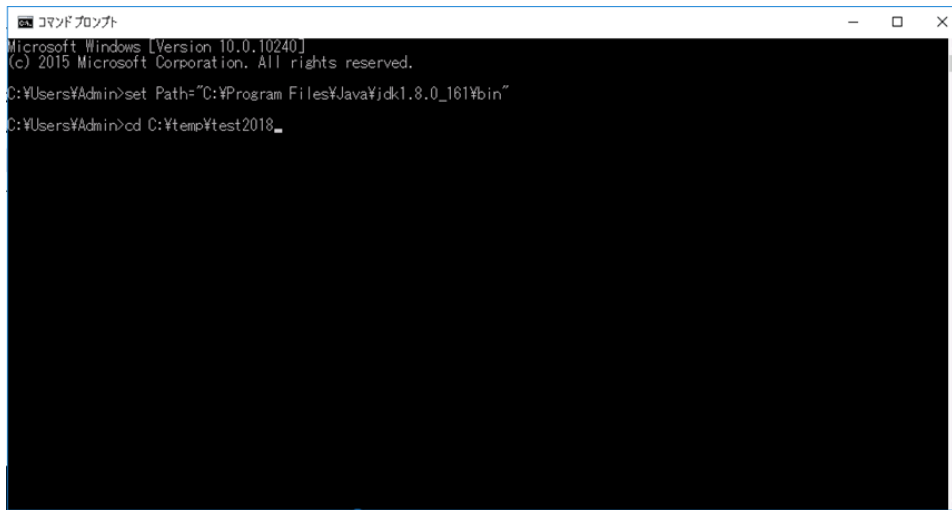
## 1-4. コード署名確認作業

本章では、デジタル署名したAndroid用(.apk形式)のファイルのコード署名確認作業について記述します。

#### 署名確認作業（併せてタイムスタンプを付与していない場合）

1. コマンドプロンプトを実行し、署名対象ファイルのあるフォルダへ移動します。

```
> set Path=(JDKインストールディレクトリ)\bin
※JDKインストールディレクトリをプログラムを探すディレクトリに指定します
> cd (作業ディレクトリ) ←作業ディレクトリ
```

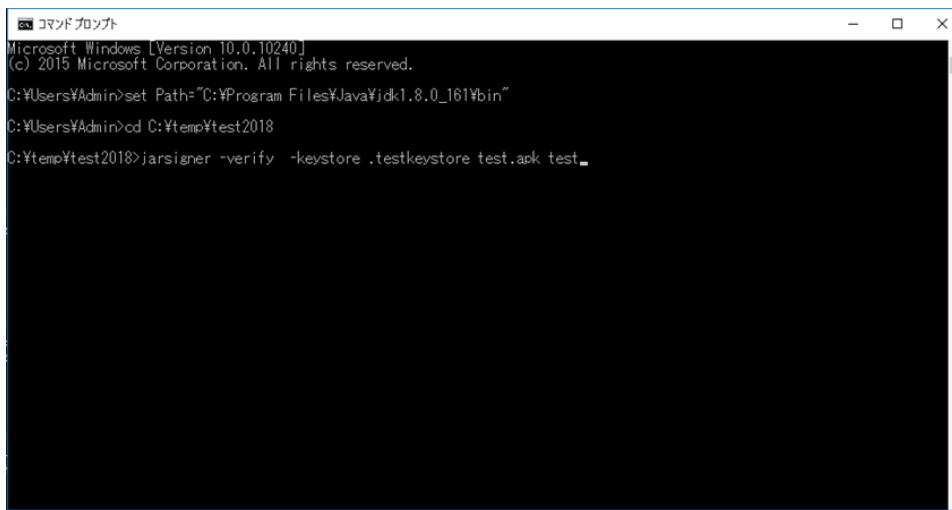


```
コマンド プロンプト
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Admin>set Path="C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\Admin>cd C:\temp\test2018_
```

2. 以下のコマンドにて署名検証を実行してください。

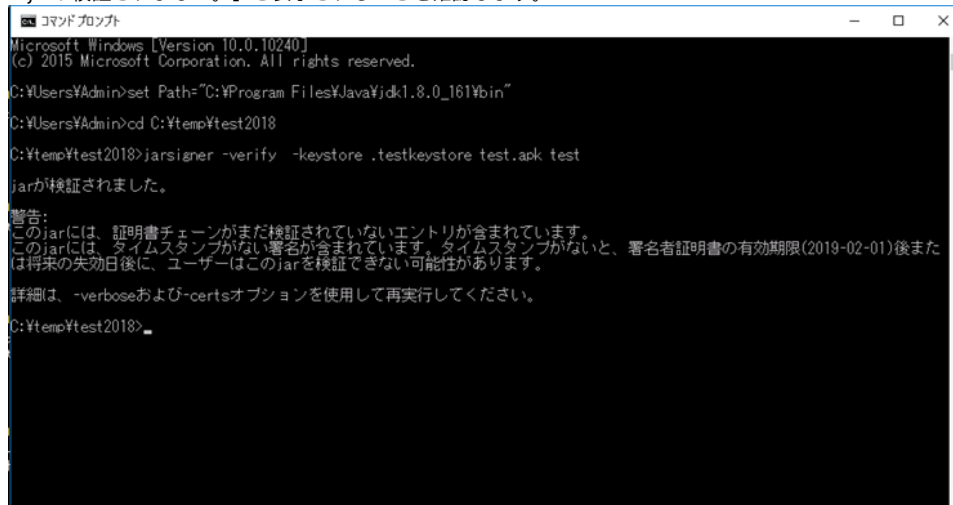
```
> jarsigner -verify -keystore (キーストアファイル名) (検証したAndroid用(.apk形式)のファイル名) (コード署名用証明書のエイリアス名)
```



```
コマンド プロンプト
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Admin>set Path="C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\Admin>cd C:\temp\test2018
C:\temp\test2018>jarsigner -verify -keystore .testkeystore test.apk test_
```

3. 「jarが検証されました。」と表示されることを確認します。



```
コマンド プロンプト
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Admin>set Path="C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\Admin>cd C:\temp\test2018
C:\temp\test2018>jarsigner -verify -keystore .testkeystore test.apk test
jarが検証されました。

警告:
このjarには、証明書チェーンがまだ検証されていないエントリが含まれています。
このjarには、タイムスタンプがない署名が含まれています。タイムスタンプがないと、署名者証明書の有効期限(2019-02-01)後または将来の失効日後に、ユーザーはこのjarを検証できない可能性があります。
詳細は、-verboseおよび-certsオプションを使用して再実行してください。
C:\temp\test2018>
```

※以下警告はタイムスタンプが付与されていないため、表示されているものです。

署名状況には問題ありません。警告内の署名者証明書の有効期限は証明書、証明書毎に異なります。

警告：

このjarには、タイムスタンプがない署名が含まれています。タイムスタンプがないと、署名者証明書の有効期限(20YY-MM-DD)後または将来の失効日後に、ユーザーはこのjarを検証できない可能性があります。

※2.のコマンド[-verbose -certs]を追加することで、署名者を表示することが可能です。



```
コマンド プロンプト
C:\temp\test2018>jarsigner -verify -keystore .testkeystore test.apk test -verbose -certs

～中略～

  署名者: "CN=          , OU=          , O=          , L=          , C=JP"
    タイジエスト・アルゴリズム: SHA-256
    署名アルゴリズム: SHA256withRSA, 2048ビット鍵
jarが検証されました。

警告:
このjarには、証明書チェーンがまだ検証されていないエントリが含まれています。
このjarには、タイムスタンプがない署名が含まれています。タイムスタンプがないと、署名者証明書の有効期限(2019-02-01)後または将来の失効日後に、ユーザーはこのjarを検証できない可能性があります。
C:\temp\test2018>
```

## 署名確認作業（併せてタイムスタンプを付与した場合）

1. コマンドプロンプトを実行し、署名対象ファイルのあるフォルダへ移動します。

```
> set Path=(JDKインストールディレクトリ)\bin
※JDKインストールディレクトリをプログラムを探すディレクトリに指定します
> cd (作業ディレクトリ) ←作業ディレクトリ
```

2. 下記のコマンドにて署名検証を実施してください。

```
> jarsigner -verify -keystore (キーストアファイル名) (検証したAndroid用(.apk形式)のファイル名) (コード署名用証明書のエイリアス名) -verbose -certs
```

```
コマンド プロンプト
Microsoft Windows [Version 10.0.16299.547]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\ymainte>set Path="C:\Program Files\Java\jdk1.8.0_181\bin"
C:\Users\ymainte>cd C:\temp\test2018
C:\temp\test2018>jarsigner -verify -keystore .testkeystore test.apk test -verbose -certs
```

3. 「jarが検証されました。」と表示されることを確認します。

また、以下の記載箇所からタイムスタンプが付与された時間を確認することができます。

タイムスタンプ付加者: "CN=\_\_\_\_\_, OU=\_\_\_\_\_, OU=\_\_\_\_\_, O=\_\_\_\_\_, L=\_\_\_\_\_, ST=\_\_\_\_\_, C=\_\_\_\_\_" 日時:  
火 8 07 07:26:40 UTC 2018

※タイムスタンプは協定世界時(UTC)で付与されています。日本標準時の現在時刻(JST)はUTCに9時間加算したものです。

```
コマンド プロンプト
Microsoft Windows [Version 10.0.16299.547]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\ymainte>set Path="C:\Program Files\Java\jdk1.8.0_181\bin"
C:\Users\ymainte>cd C:\temp\test2018
C:\temp\test2018>jarsigner -verify -keystore .testkeystore test.apk test -verbose -certs

~中略~

s=シグネチャが検証されました
m=エントリがマニフェスト内にリストされます
k=1つ以上の証明書がキーストアで検出されました
i=1つ以上の証明書がアイデンティティ・スコープで検出されました
X=指定した別名で署名されていません

署名者: "CN=_____, OU=_____, OU=_____, O=_____, L=_____, ST=_____, C=_____"
ダイジェスト・アルゴリズム: SHA-256
署名アルゴリズム: SHA256withRSA, 2048ビット鍵
タイムスタンプ付加者: "CN=_____, OU=_____, O=_____, L=_____, ST=_____, C=_____"
日時: 火 8 07 07:26:40 UTC 2018
タイムスタンプのダイジェスト・アルゴリズム: SHA-256
タイムスタンプの署名アルゴリズム: SHA256withSHA256withRSA, 2048ビット鍵

jarが検証されました。

警告:
このjarには、証明書チェーンがまだ検証されていないエントリが含まれています。
このjarに含まれる署名済エントリは、指定された別名によって署名されていません。
このjarに含まれる署名済エントリは、このキーストア内の別名によって署名されていません。

C:\temp\test2018>
```

※1.以下警告は署名状況には問題ありません。

警告:

このjarには、証明書チェーンがまだ検証されていないエントリが含まれています。

このjarに含まれる署名済エントリは、指定された別名によって署名されていません。

このjarに含まれる署名済エントリは、このキーストア内の別名によって署名されていません。

