

JAVA .jar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 署名確認作業の補足追加	NII
V.2.0	2018/2/26	キーストアファイルの形式をJKSからPKCS12に変更、動作環境の変更に伴う修正	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。
(本マニュアルではJava SE Development Kit (以下JDKと呼ぶ) 8u161での実行例を記載しております。)
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 2. JDK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>

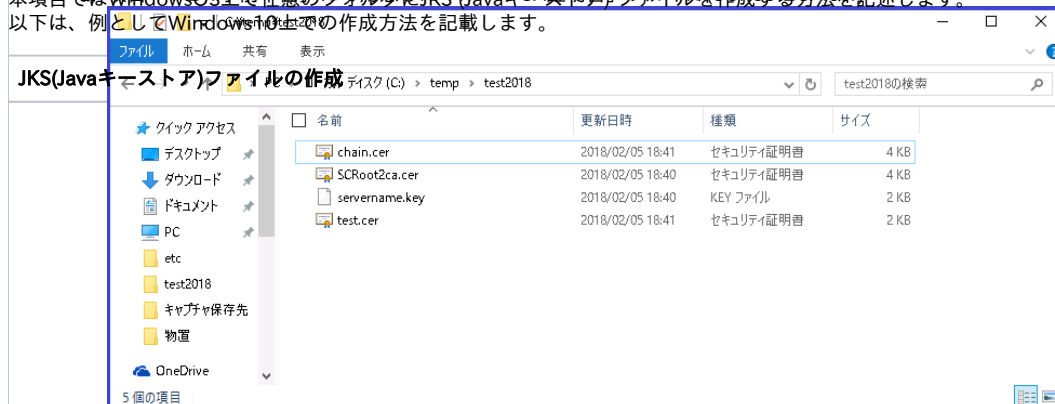
Security Communication RootCA3 証明書：
<https://repository.secomtrust.net/SC-Root3/SCRoot3ca.cer>

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。

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「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)
 - c. 項目「1-2-1事前準備」にて用意した「JavaキーCA証明書」ファイル(SCRoot2ca.cer)
- 本項目ではWindowsOS上での作成方法を記載します。

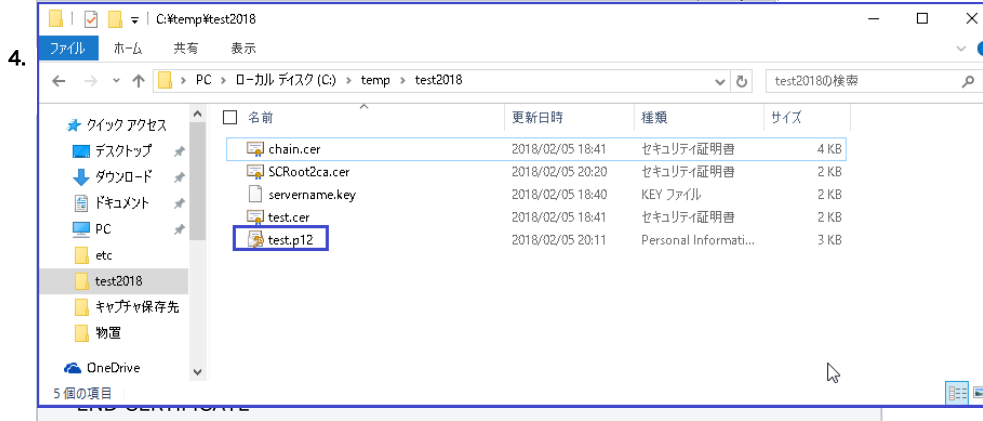


2. CAfile に指定する証明書をDER形式からPEM形式に変換します。

・ 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. 任意のサブディレクトリにはC上にて取得した18とともCA証明書下の中間CA証明書を含むディレクトリにより、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。
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)



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\test2018)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索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、以下のコマンドを実行しPKCS#12ファイルを生成してください。

移動後、以下のコマンドを実行し、keytool (鍵ペアのファイル名)を作成し、(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) -keystore (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 -srcaalias (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) -destkeypass (キーストアに設定したいパスワード)

```
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

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

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

C:\temp\test2018>openssl pkcs12 -export -chain -inkey servname.key -CAfile chain.cer -in test.cer -out test.p12 -name test -caname root
```

7.

```
コマンドプロンプト
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

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

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

C:\temp\test2018>keytool -importkeystore -srckeystore test.p12 -destkeystore .testkeystore -srcstoretype pkcs12 -deststoretype pkcs12 -srcaalias test -destalias test -destkeypass [redacted]
```

、パスフレーズを入力してください。

3.

```
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

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

keypass」に設定した文字列を入力して

```
コマンドプロンプト - keytool -importkeystore -srckeystore test.p12 -destkeystore .testkeystore -srcstoretype pkcs12 -deststoretype pkcs12 -srcaalias test -des...
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

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

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

C:\temp\test2018>keytool -importkeystore -srckeystore test.p12 -destkeystore .testkeystore -srcstoretype pkcs12 -deststoretype pkcs12 -srcaalias test -destalias test -destkeypass [redacted]
出力先キーストアのパスワードを入力してください: [redacted]
```

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。

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。

```
コマンドプロンプト - keytool -importkeystore -srckeystore test.p12 -destkeystore .testkeystore -srcstoretype pkcs12 -deststoretype pkcs12 -salias test -des...  
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 -destkeystore .testkeystore -srcstoretype pkcs12 -deststoretype pkcs12 -salias test -destalias test -destkeypass  
キーストアtest.p12を.testkeystore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: -
```

5. [パスワードを入力してください:]と表示されますので、パスワードを入力してください。

```
コマンドプロンプト - keytool -importkeystore -srckeystore test.p12 -destkeystore .testkeystore -srcstoretype pkcs12 -deststoretype pkcs12 -salias test -des...  
(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 -destkeystore .testkeystore -srcstoretype pkcs12 -deststoretype pkcs12 -salias test -destalias test -destkeypass  
キーストアtest.p12を.testkeystore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. [パスワードを入力してください:]と表示されますので、パスワードを入力してください。

```
コマンドプロンプト - keytool -importkeystore -srckeystore test.p12 -destkeystore .testkeystore -srcstoretype pkcs12 -deststoretype pkcs12 -salias test -des...  
(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 -destkeystore .testkeystore -srcstoretype pkcs12 -deststoretype pkcs12 -salias test -destalias test -destkeypass  
キーストアtest.p12を.testkeystore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: -
```

10. [パスワードを入力してください:]と表示されますので、パスワードを入力してください。



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。

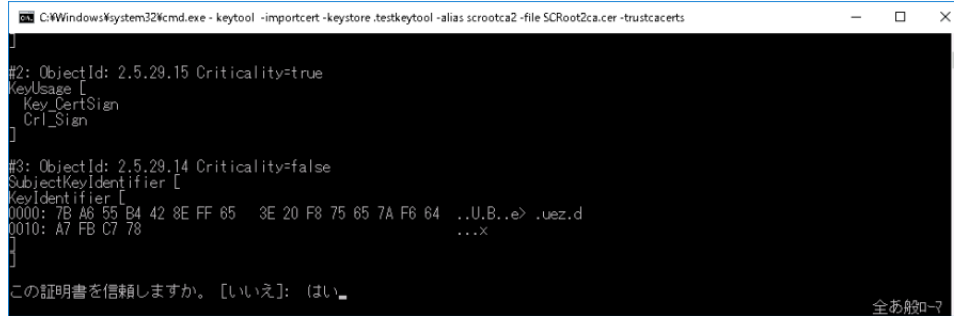
```
> keytool -importcert -keystore (キーストアファイル名) -alias (設定したいルートCA証明書のエイリアス名) -file (ルートCA証明書のファイル名) -trustcacerts
```

```
コマンドプロンプト  
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 -destkeystore .testkeystore -srcstoretype pkcs12 -deststoretype pkcs12 -salias test -destalias test -destkeypass  
キーストアtest.p12を.testkeystore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
```

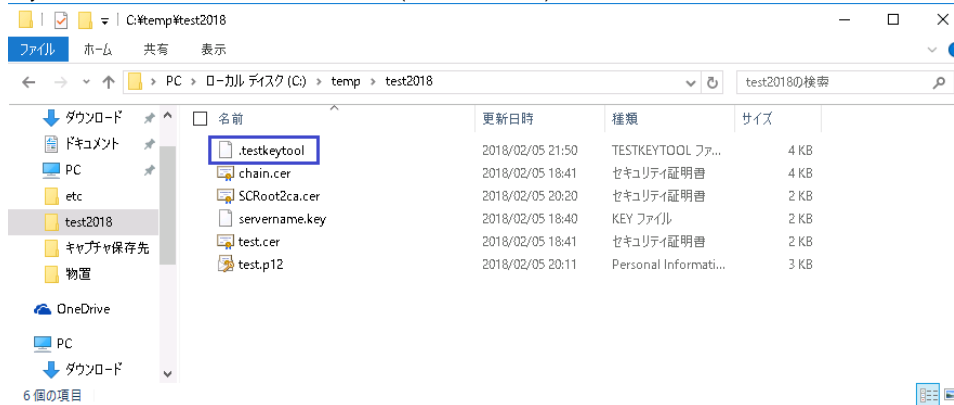
8. 「キーストアのパスワードを入力してください:」と表示されますので、JKS (Javaキーストア) ファイルを保護するパスワードを入力してください。

```
コマンドプロンプト - keytool -importcert -keystore .testkeytool -alias scrootca2 -file SCRoot2ca.cer -trustcacerts  
C:\Users\ymainte>cd C:\temp\test2018  
  
C:\temp\test2018>keytool -importkeystore -srckeystore test.p12 -destkeystore .testkeystore -srcstoretype pkcs12 -deststoretype pkcs12 -salias test -destalias test -destkeypass  
キーストアtest.p12を.testkeystore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  
キーストアのパスワードを入力してください: -
```

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。



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。

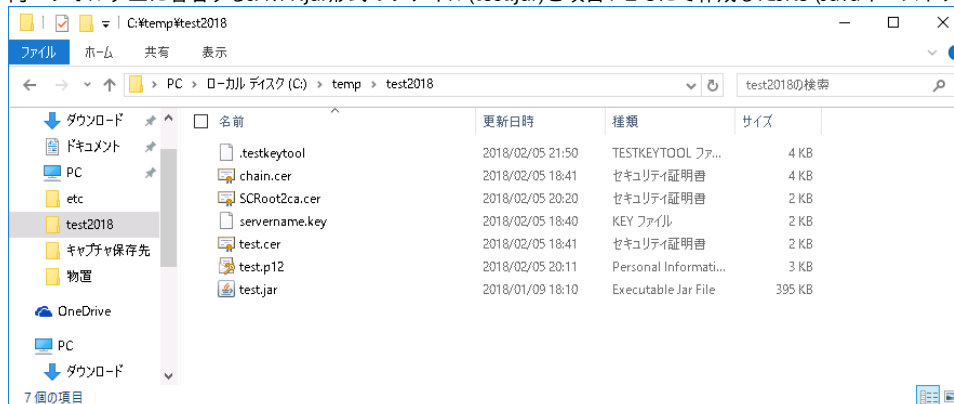


1-3. 署名

本章ではJAVA .jar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JAVA .jar形式のファイル(test.jar)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。

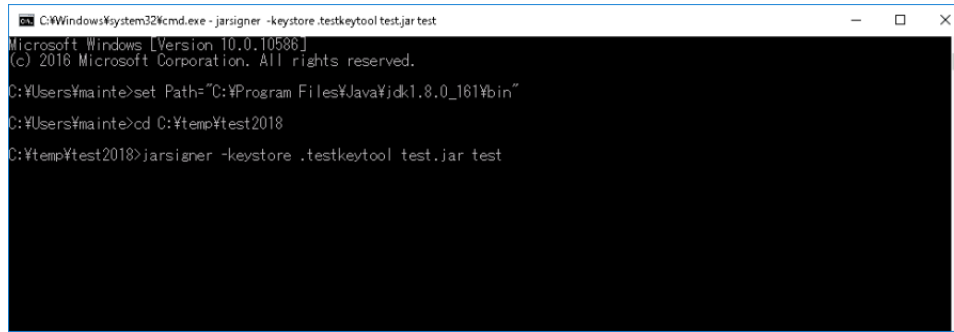


2. コマンドプロンプトを実行し、署名対象ファイルのあるフォルダへ移動します。

```
> 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JAVA .jar形式のファイル(ここではtest.jar)に対して以下のコマンドにて署名を実行してください。

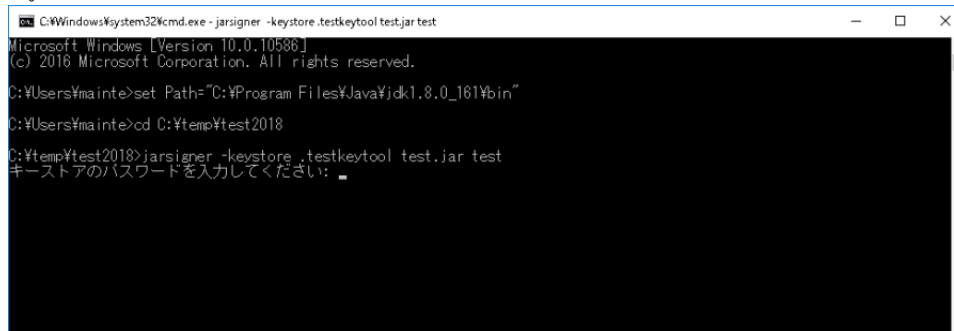
jarsigner -keystore (キーストアファイル名) (署名したいJAVA .jar形式のファイル名) (キーストア内の証明書のエイリアス名)



```
C:\Windows\system32\cmd.exe - jarsigner -keystore .testkeytool test.jar test
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\mainie>set Path="C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\mainie>cd C:\temp\test2018
C:\temp\test2018>jarsigner -keystore .testkeytool test.jar test
```

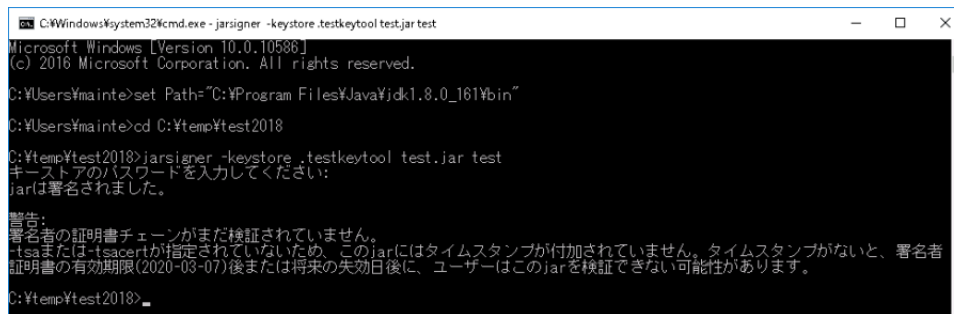
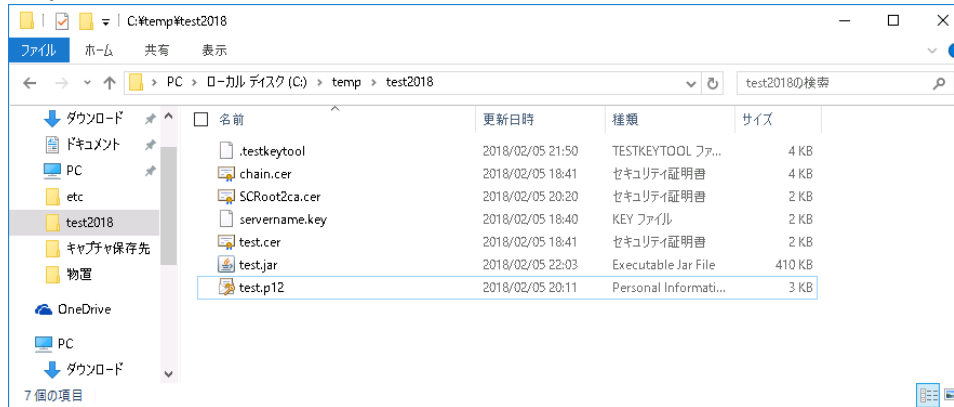
4. 「キーストアのパスワードを入力してください:」と表示されますので、JKS (Javaキーストア) ファイルを保護するパスフレーズを入力してください。



```
C:\Windows\system32\cmd.exe - jarsigner -keystore .testkeytool test.jar test
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\mainie>set Path="C:\Program Files\Java\jdk1.8.0_161\bin"
C:\Users\mainie>cd C:\temp\test2018
C:\temp\test2018>jarsigner -keystore .testkeytool test.jar 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JAVA .jar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。

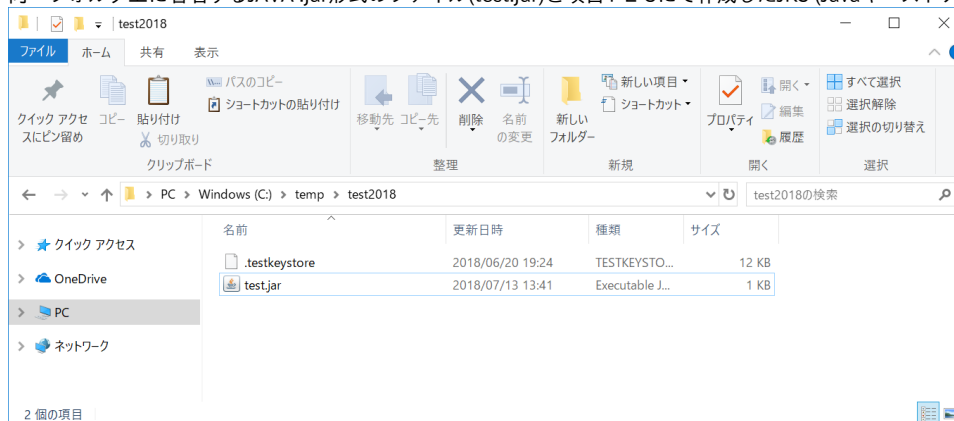
署名状況には問題ありません。警告内の署名書証明書の有効期限は証明書、証明書毎に異なります。

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JAVA .jar形式のファイル(test.jar)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JAVA .jar形式のファイル(ここではtest.jar)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JAVA .jar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)

```
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\mainte>set Path="C:\Program Files\Java\jdk1.8.0_181\bin"
C:\Users\mainte>cd C:\temp\test2018
C:\temp\test2018>jarsigner -keystore .testkeystore test.jar test -tsa
```

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\mainte>set Path="C:\Program Files\Java\jdk1.8.0_181\bin"
C:\Users\mainte>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。対象のJAVA .jar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。

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

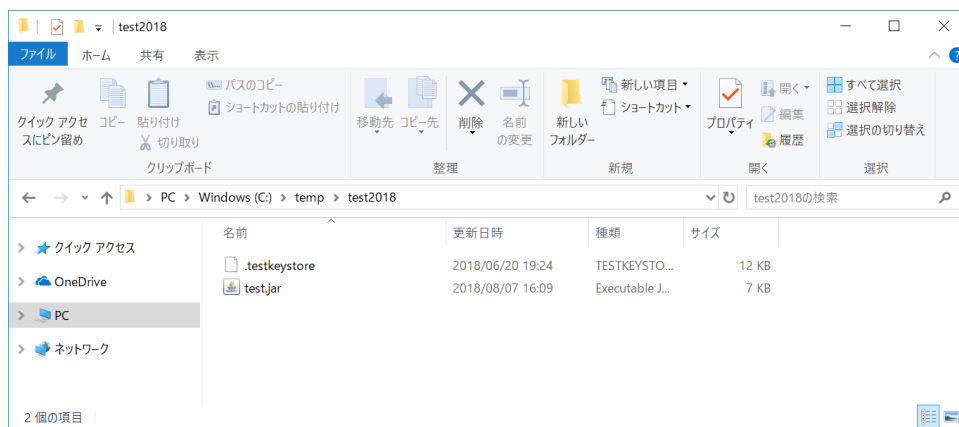
C:\Users\mainte>set Path="C:\Program Files\Java\jdk1.8.0_181\bin"

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

C:\temp\test2018>jarsigner -keystore testkeystore test.jar 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JAVA .jar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。

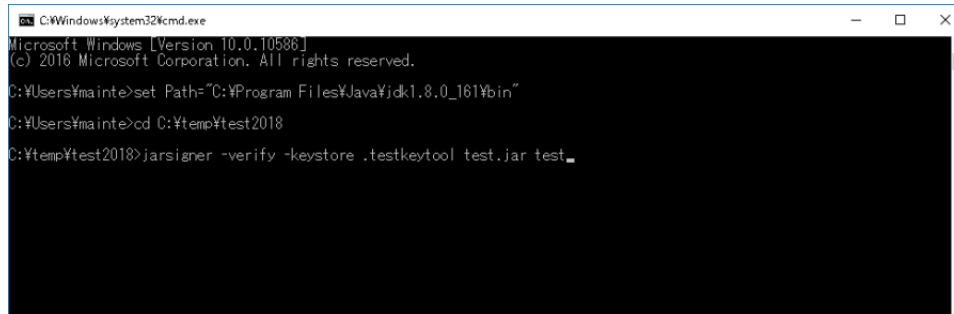
署名確認作業（併せてタイムスタンプを付与しない場合）

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
```

2. 下記のコマンドにて署名検証を実施してください。

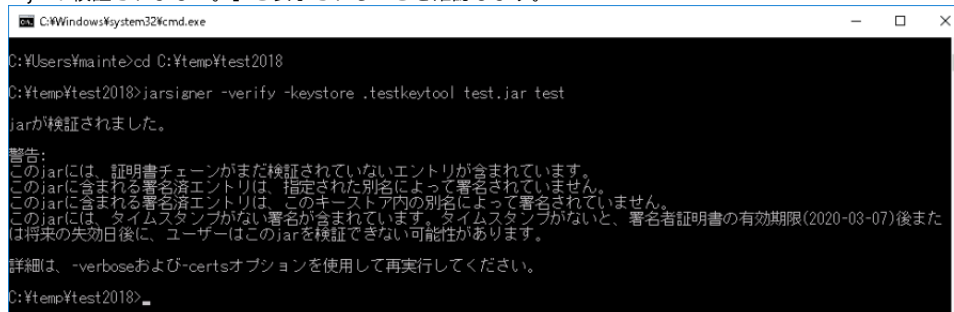
```
> jarsigner -verify -keystore (キーストアファイル名) (検証したいJAVA .jar形式のファイル名) (コード署名用証明書のエイリアス名)
```



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

C:\Users\%mainte>set Path="C:\Program Files\Java\jdk1.8.0_101\bin"
C:\Users\%mainte>cd C:\temp\test2018
C:\temp\test2018>jarsigner -verify -keystore .testkeytool test.jar 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。



```
C:\Windows\system32\cmd.exe
C:\Users\%mainte>cd C:\temp\test2018
C:\temp\test2018>jarsigner -verify -keystore .testkeytool test.jar 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。
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(2020-03-07)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>_
```

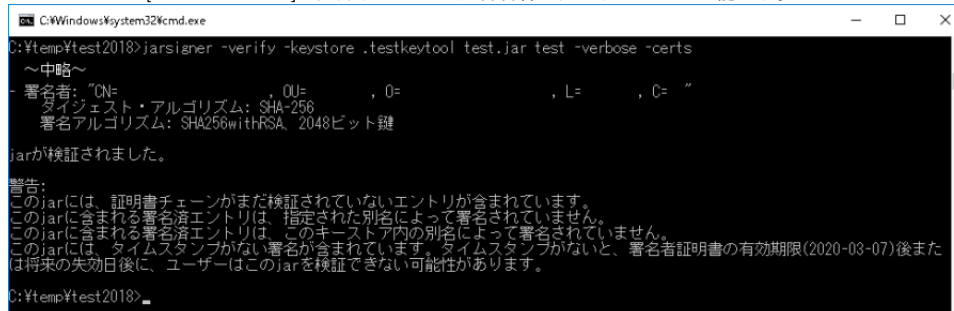
※1.以下警告はタイムスタンプが付与されていないため、表示されているものです。

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：

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。



```
C:\Windows\system32\cmd.exe
C:\temp\test2018>jarsigner -verify -keystore .testkeytool test.jar test -verbose -certs
~中略~
- 署名者: "CN=, OU=, O=, L=, C="
  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。
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(2020-03-07)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>_
```

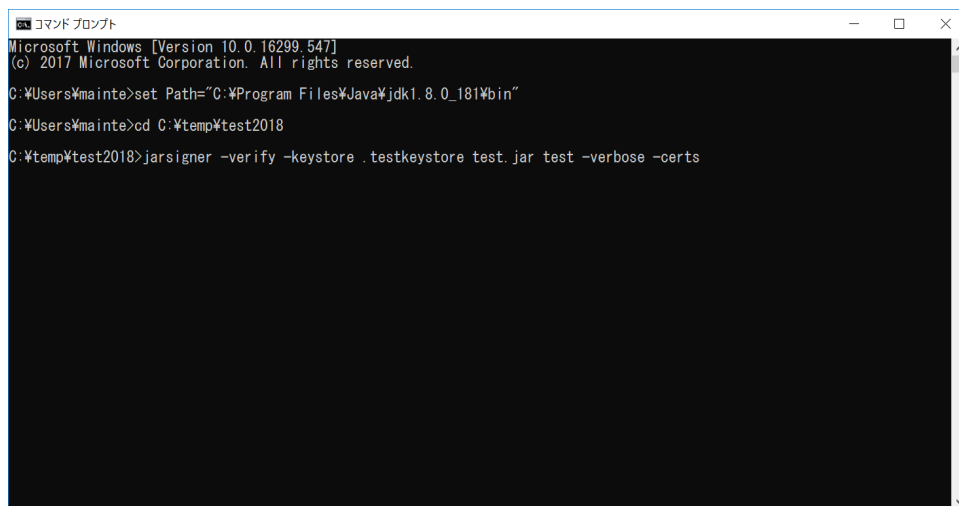
署名確認作業（併せてタイムスタンプを付与した場合）

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) (検証したいJAVA .jar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\mainte>set Path="C:\Program Files\Java\jdk1.8.0_181\bin"
C:\Users\mainte>cd C:\temp\test2018
C:\temp\test2018>jarsigner -verify -keystore .testkeystore test.jar test -verbose -certs
```

「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 07:09:57 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.jar 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
l=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= , OU= , OU= , OU= , O= , L= , 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= , OU= , OU= , OU= , OU= , O= , L= , C= "
日時: 火 8 07 07:09:57 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。