Adobe AIR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.2.0	2018/2/28	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	2018/10/24	タイムスタンプ付与に関して記載を追加	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AIR SDK 16.0、JDK 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. Adobe AIR SDKもしくはAdobe Flex SDKがインストールされていること 3. 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. 「鍵ペアの生成とCSRの作成～ 証明書の申請から取得まで」において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

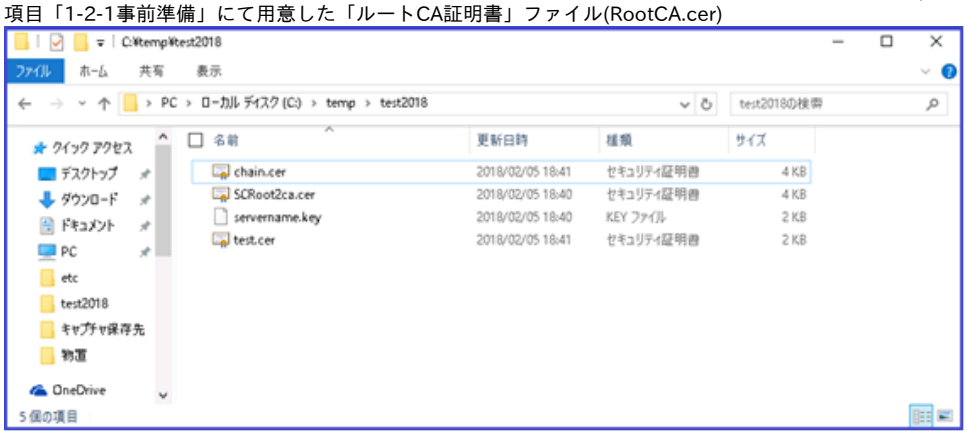
Security Communication RootCA3 証明書：
https://repository.secomtrust.net/SC-Root3/SCRoot3ca.cer

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」にて作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

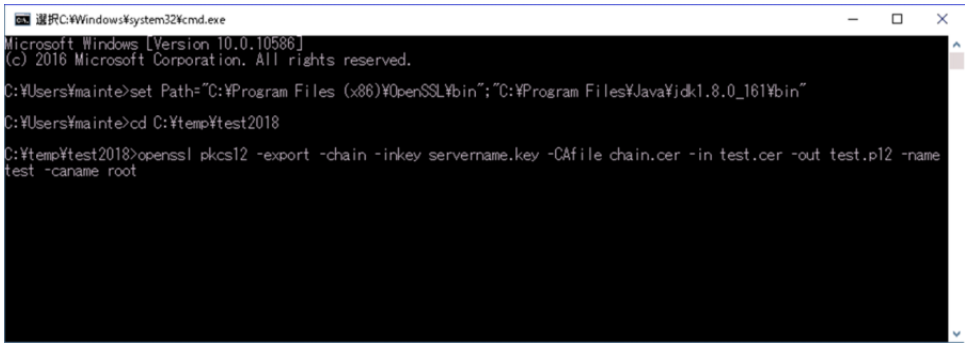

```
----BEGIN CERTIFICATE----
MIIEcTCCA1mgAwIBAgIIasWHLdnQB2owDQYJKoZIhvcNAQELBQAwbzELMAkGA1UE
BhMCSlAxFDASBgNVBAcMC0FjYWRlbWUtb3BzMSowKAYDVQQKDCFOYXRpb25hbCB
J
bnN0aXR1dGUgb2YgSW5mb3JtYXRpY3MxHjAcBgNVBAMMFU5JSSBPcGVyYXRpbmcg
Q0EgLSBHMjAeFw0xNTAzMTIwMTA4MDJaFw0xNzA0MTEwMTA4MDJaMHAxCzAJBgN
V
（中略）
LmeW0e/xkkxwdmKv5y5txLlFcp53AZl/vjn3BHp42PFkkTISEmAUiCtQ2A25QDRR
RG33laC8E8Tl/SnOA8h95XQtGWm47PrljXyYtle0rFousbpIoW8MZw4gDXVQ3485
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\test2018)へ移動します。

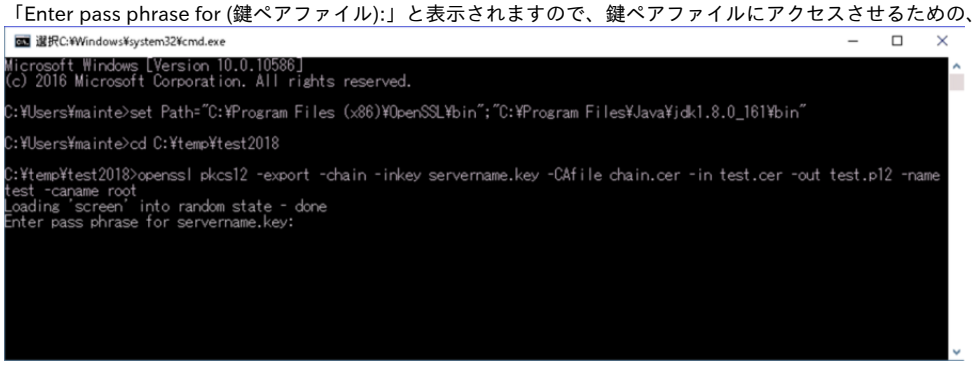
```
＞ set Path=(OpenSSLインストールディレクトリ)\bin
※OpenSSLインストールディレクトリをプログラムを探すディレクトリに指定します

＞ 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PKCS#12ファイルを作成してください。

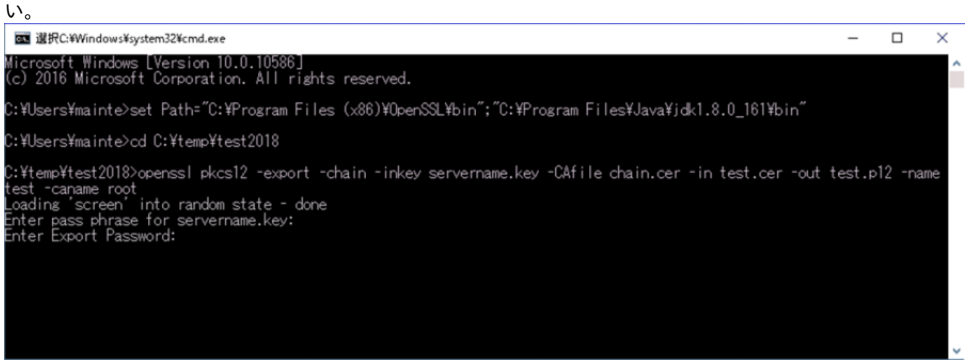
```
＞ openssl pkcs12 -export -chain -inkey (鍵ペアのファイル名) -CAfile (ルートCA証明書と中間CA証明書を連結させたファイル) -in (コード署名用の証明書ファイル名) -out (PKCS#12形式で出力するファイル名) -name (コード署名用証明書のエイリアス名) -caname (ルートCA証明書と中間CA証明書のエイリアス名)
```



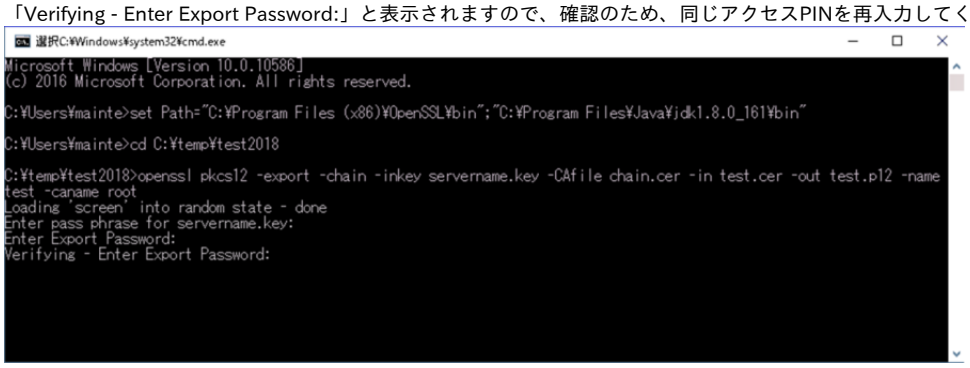
6. 「Enter pass phrase for (鍵ペアファイル):」と表示されますので、鍵ペアファイルにアクセスさせるための、パスフレーズを入力してください。



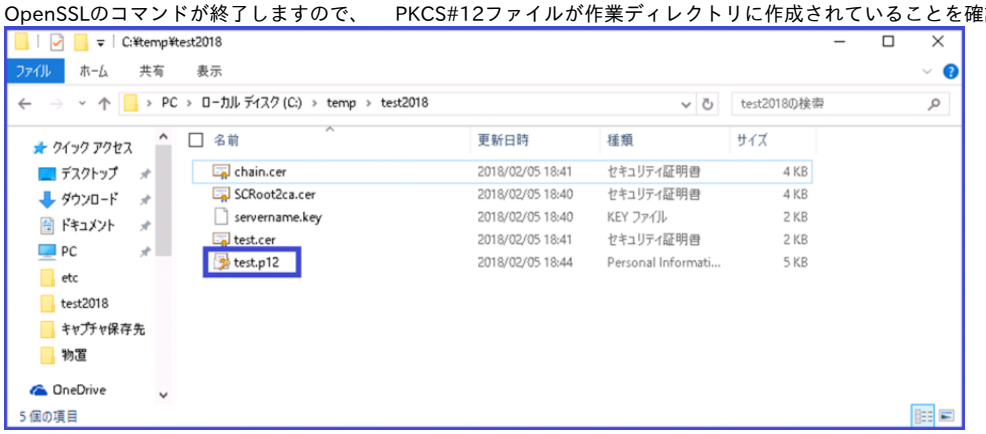
7. 「Enter Export Password:」と表示されますので、PKCS#12形式のファイルを保護するためのアクセスPINとして任意の文字列を入力してください。



8. 「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。



9. 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作成されていることを確認してください。

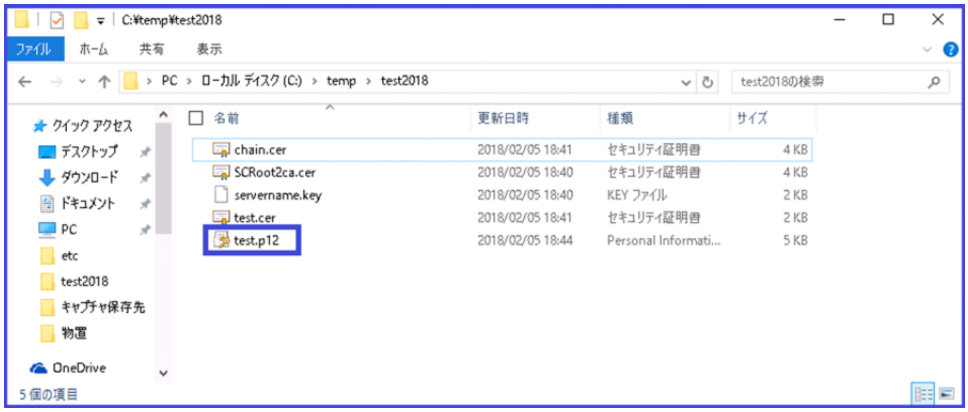


1-2-3. JKS (Javaキー・ストア) ファイルの作成

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)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\test2018)へ移動します。

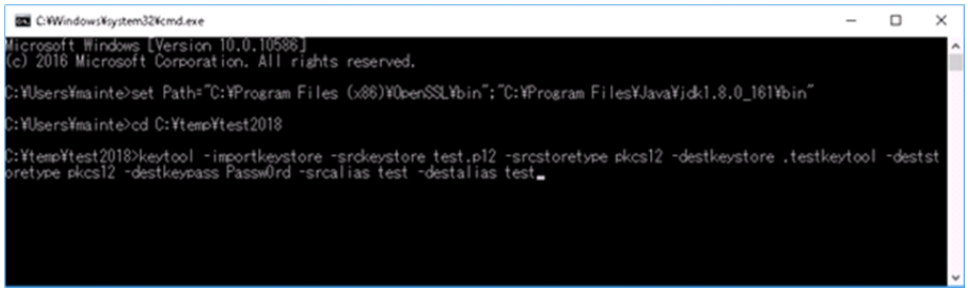
```
> set Path=(JDKインストールディレクトリ)\bin
※JDKインストールディレクトリをプログラムを探すディレクトリに指定します
> cd (作業ディレクトリ) ←作業ディレクトリ
```

3. 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。

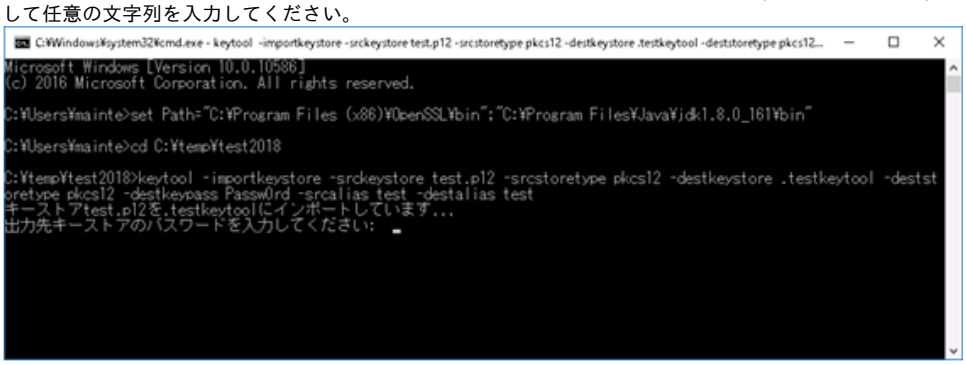
```
> keytool -importkeystore -srckeystore (PKCS#12ファイル名) -srcstoretype pkcs12 -destkeystore （作成したいキーストアファイル名） -deststoretype pkcs12 -destkeypass （キーストアに設定したいパスワード） -srcalias ( PKCS#12ファイルで利用されているエイリアス名) -destalias （登録したいエイリアス名）
```

※ 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。

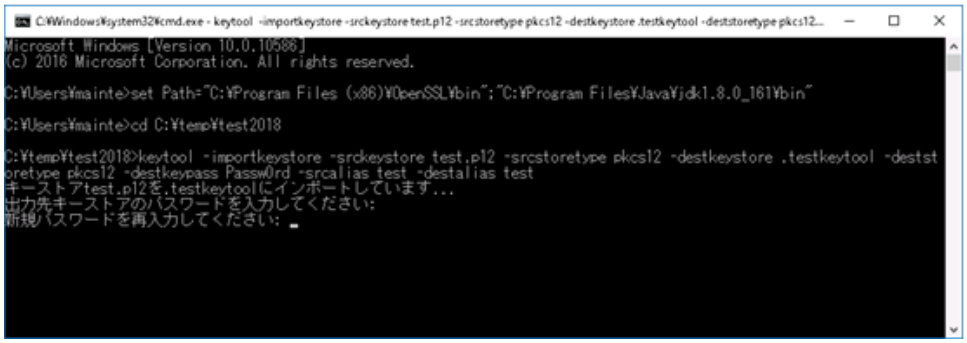
```
keytool -v -list -keystore (PKCS#12ファイル名)
```



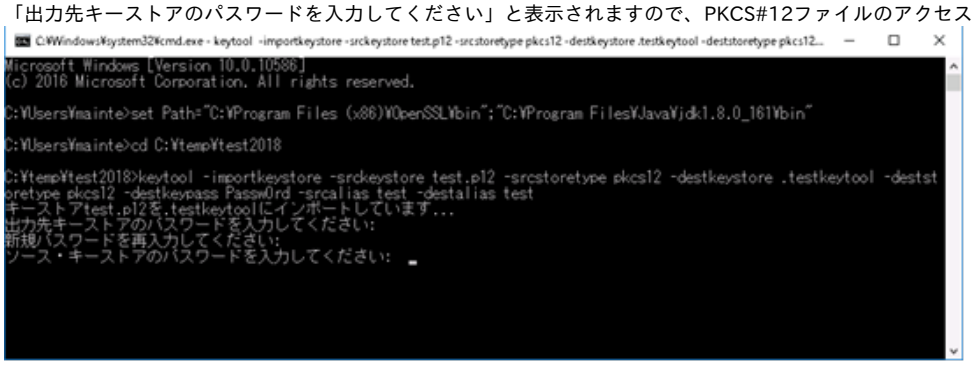
4. 「出力先キーストアのパスワードを入力してください：」と表示されますので、JKS (Javaキー・ストア) ファイルを保護するためのパスワードとして任意の文字列を入力してください。



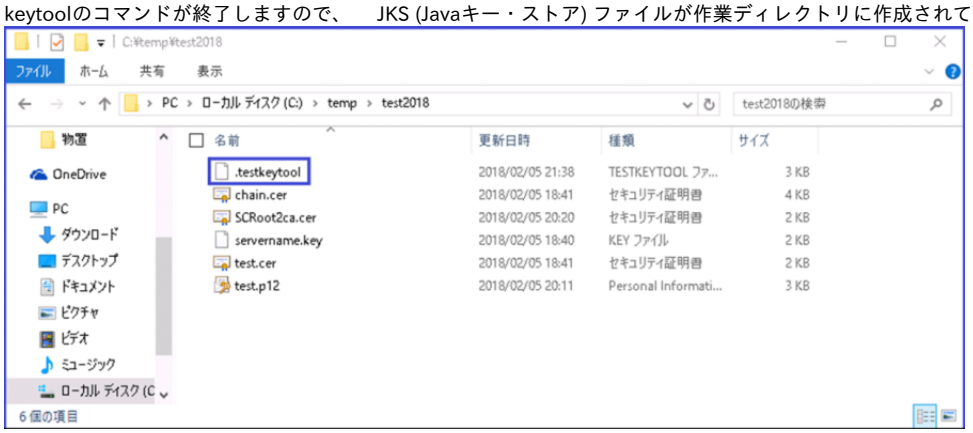
5. 「新規パスワードを再入力してください:」と表示されますので、確認のため、同じパスワードを再入力してください。



6. 「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、PKCS#12ファイルのアクセスPINを入力してください。



7. 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。



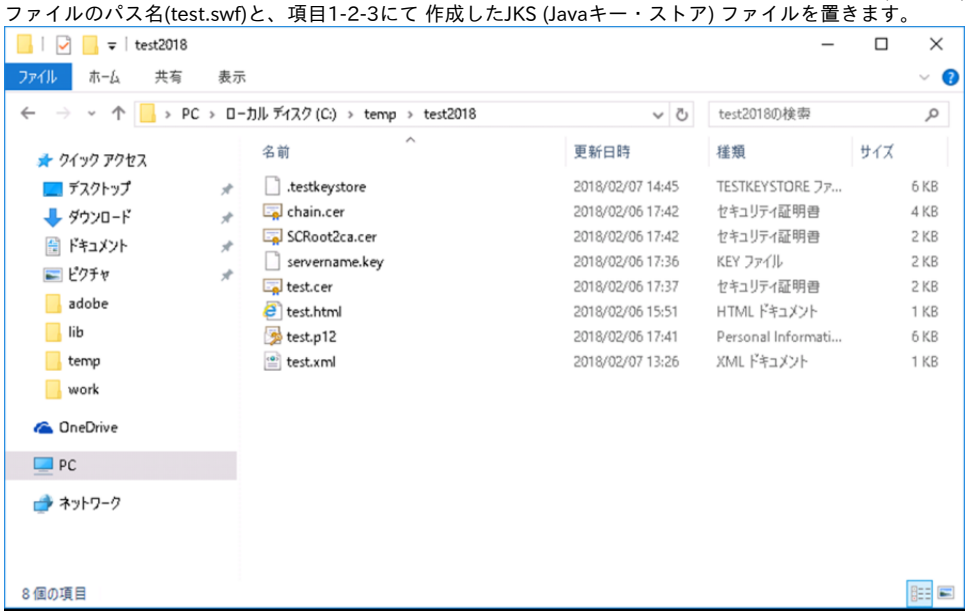
1-3. 署名

本章では、Adobe AIR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。
Adobe AIR形式のファイルへの署名はファイル作成時にAdobe AIR SDK、もしくはAdobe Flex SDK内のadtコマンドを利用して署名します。

【adtコマンドを用いたタイムスタンプの付与について】
当サービスで提供しているタイムスタンプ局はadtコマンドを用いたタイムスタンプの付与に関しては非対応となっております。

署名作業

1. 同一フォルダ上にAdobe AIR形式ファイル作成に必要な署名されるアプリケーション記述ファイル(test.xml)、アプリケーションの使用する任意のファイルのパス名(test.swf)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。



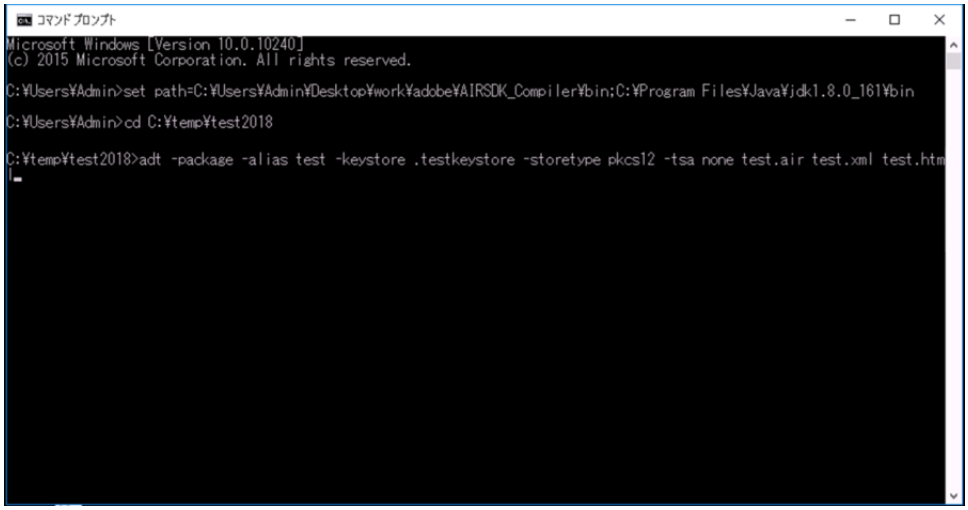
2. コマンドプロンプトを実行し、署名対象ファイルのあるフォルダへ移動します。

> set Path=(Adobe AIR SDKもしくはAdobe Flex SDK);(JDKインストールフォルダ)\bin

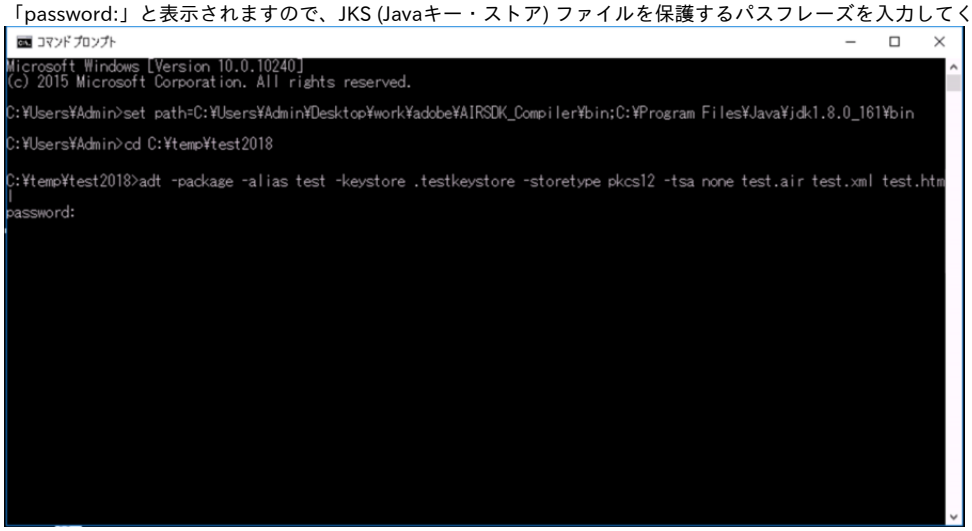
※JDKインストールフォルダをプログラムを探すフォルダに指定します
> cd (作業ディレクトリ) ←作業ディレクトリ

3. フォルダ移動後、Adobe AIR形式のファイル作成と同時に、作成ファイルへの署名を下記のコマンドにて実施してください。

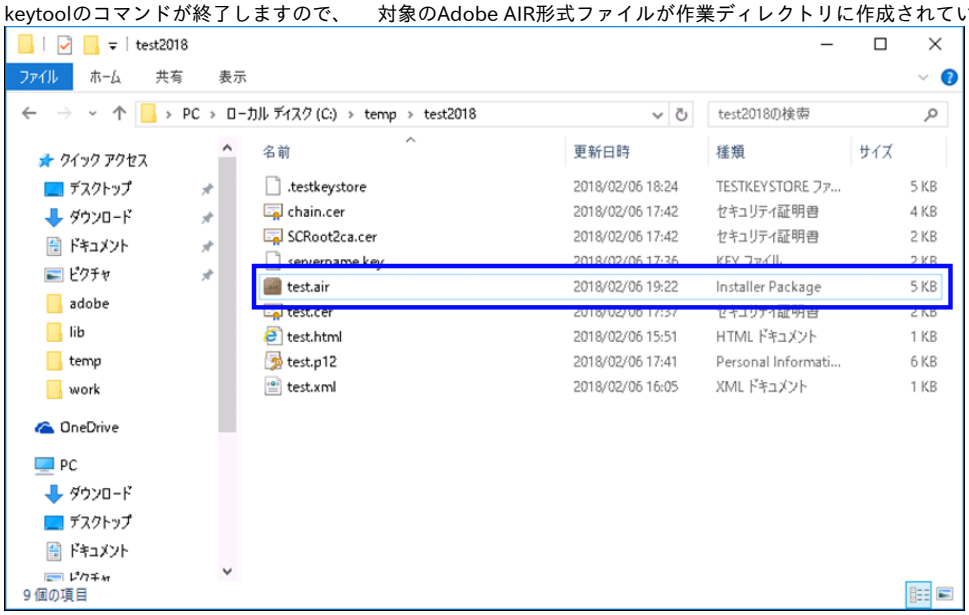
> adt -package -alias (キーストア内の証明書のエイリアス名) -keystore （キーストアファイル名） -storetype pkcs12 -tsa none (作成するAdobe AIRファイル名) (署名されるアプリケーション記述ファイルのパス名) (トップレベルHTMLファイルのパス名)



4. 「password:」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。



5. keytoolのコマンドが終了しますので、　対象のAdobe AIR形式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。

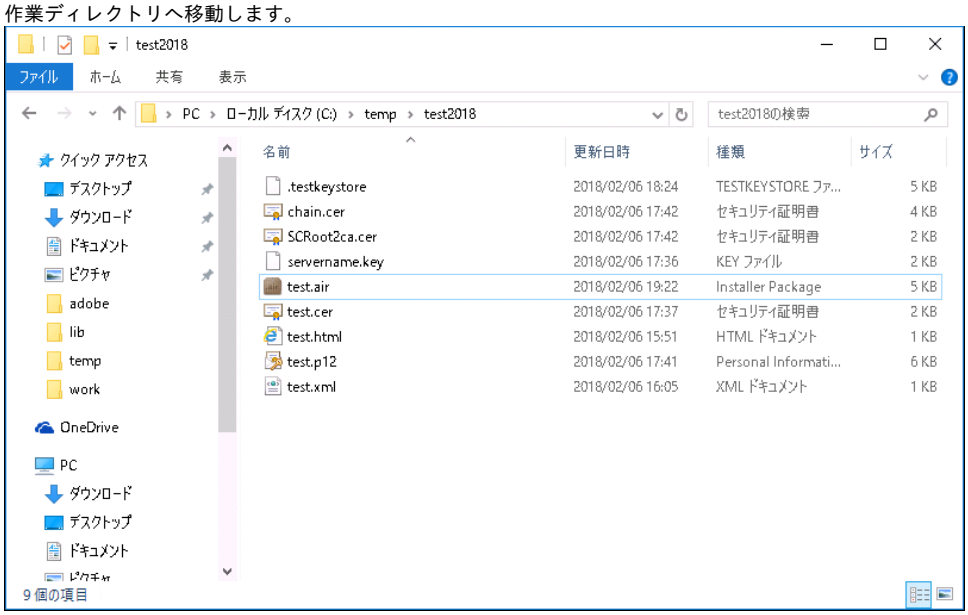


1-4. コード署名確認作業

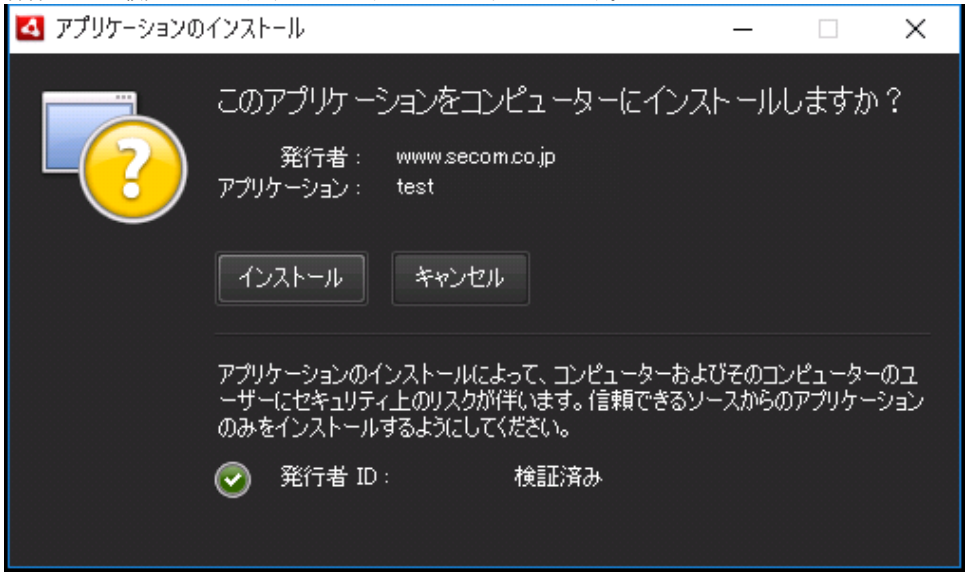
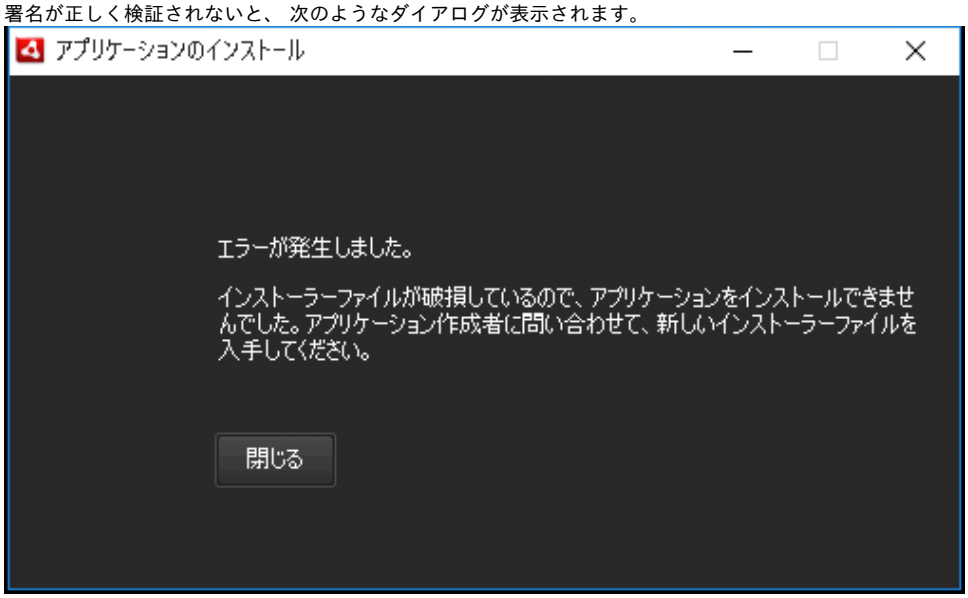
本章では、デジタル署名したAdobe AIR形式のファイルのコード署名確認作業について記述します。
タイムスタンプを付与した場合と付与していない場合で確認の手順に違いはありません。

署名確認作業

1. 作業ディレクトリへ移動します。



2. 作成されたAdobe AIR形式ファイルを実行し、以下①のダイアログログが表示されることを確認します。
署名が正しく検証されると、 次のようなダイアログが表示されます。



3. 署名が正しく検証されないと、 次のようなダイアログが表示されます。