คู่มือการใช้งานสำหรับนักพัฒนาระบบ

XAdES Signer (Java)

โครงการ จ้างที่ปรึกษาเพื่อบริหารโครงการปรับเปลี่ยนบริการภาครัฐที่เกี่ยวกับการออกใบอนุญาต หรือหลักฐานสำคัญ ให้เป็นดิจิทัล ด้วยมาตรฐานที่จำเป็น

สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์

23 เมษายน 2564

Strategy Consulting & Digital Transformation

FRONTIS

บริษัท ฟรอนทิส จำกัด

สารบัญ

การกำหนดค่าสำหรับ Library

1. Environment และ Software ที่เกี่ยวข้อง

Library นี้พัฒนาด้วยภาษา Java ซึ่งมี environment และ software ที่จำเป็นในการใช้พัฒนา ดังนี้

1. Java JRE 8 (ติดตั้งทั้ง 32-bit และ 64-bit)
2. Eclipse (Editor สำหรับใช้การพัฒนา)

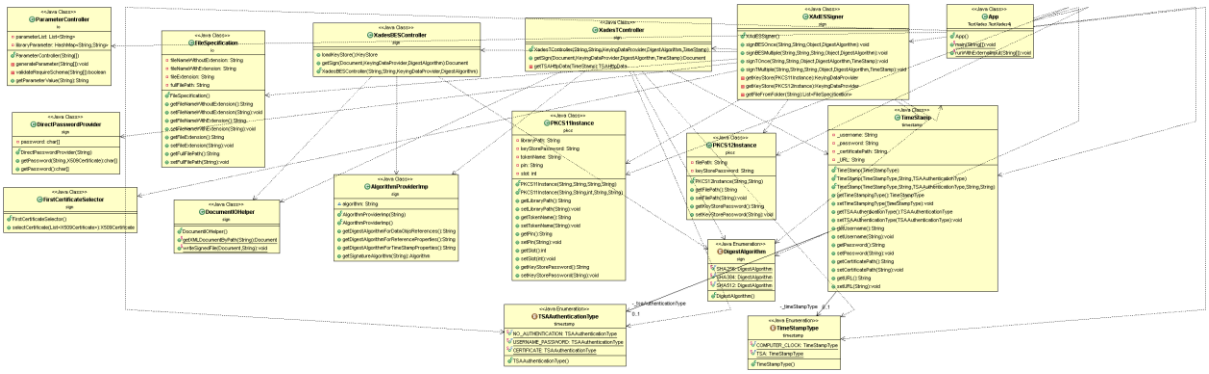
2. การกำหนด Dependencies

Library มีการใช้งาน maven library อื่น ๆ เพิ่มเติมประกอบในการพัฒนา เพื่อให้ **Library** สามารถทำงานได้อย่างสมบูรณ์ จึงจำเป็นต้องติดตั้ง **Dependency** ที่เกี่ยวข้องทั้งหมด (ประกาศในไฟล์ **pom.xml**) ดังนี้

#	groupId	artifactId	version
1	org.slf4j	slf4j-api	2.0.0-alpha1
2	org.slf4j	slf4j-simple	2.0.0-alpha1
3	com.google.inject	guice	5.0.0-BETA-1
4	com.google.inject.extensions	guice-multibindings	2.0
5	org.bouncycastle	bcprov-jdk15on	1.68
6	org.bouncycastle	bcpkix-jdk15on	1.68
7	commons-io	commons-io	2.5
8	org.apache.santuario	xmlsec	2.2.0
9	org.apache.logging.log4j	log4j-api	2.13.3
10	org.apache.logging.log4j	log4j-core	2.13.3
11	org.apache.logging.log4j	log4j-slf4j-impl	2.13.3

ข้อมูลรายละเอียด Library

1. Class and method



Class XAdESSigner

Method name	Parameter in	Return	Remark
XAdESSigner	-	-	Class Constructor
signBESOnce	**String** inputFilePath, **String** outputFilePath, **IPKCSInstance** pkcs, **DigestAlgorithm** digestAlgorithm	-	Sign XAdES-BES ไฟล์เดียว
signBESMultiple	**String** inputFolderPath, **String** outputFolderPath, **String** outputSuffix, **IPKCSInstance** pkcs, **DigestAlgorithm** digestAlgorithm	-	Sign XAdES-BES หลายไฟล์
signTOnce	**String** inputFilePath, **String** outputFilePath, **IPKCSInstance** pkcs, **DigestAlgorithm** digestAlgorithm, **TimeStamp** timeStamp	-	Sign XAdES-T ไฟล์เดียว
signTMultiple	**String** inputFolderPath, **String** outputFolderPath,	-	Sign XAdES-T หลายไฟล์

Method name	Parameter in	Return	Remark
	String outputSuffix, **IPKCSInstance** pkcs, **DigestAlgorithm** digestAlgorithm, **TimeStamp** timeStamp		
getKeyStore	**PKCS11Instance** pkcs11Instance	**KeyingDataProvider**	อ่านค่าจาก PKCS11 แล้วส่งแปลงเป็น instance ของ class **KeyingDataProvider**
getKeyStore	**PKCS12Instance** pkcs12Instanc	**KeyingDataProvider**	อ่านค่าจาก PKCS12แล้วส่งแปลงเป็น instance ของ class **KeyingDataProvider**
getFileFromFolder	**String** folderPath	**List<FileSpecification>**	แสดงรายชื่อไฟล์ทั้งหมดในโฟล์เดอร์ที่เลือก

Class ParameterController

Method name	Parameter in	Return	Remark
ParameterController	**String[]** args	-	Class constructor
generateParameter	**String[]** args	-	ประมวลผล external จาก Main method
validateRequireSchema	**String[]** args	**boolean**	ตรวจสอบความครบถ้วนของ parameter ที่จำเป็น
getParameterValue	**String** key	**String**	คืนค่าของ parameter ตาม key ที่ส่งค่าเข้ามา

Class PKCS11Instance

Method name	Parameter in	Return	Remark
PKCS11Instance	**String** tokenName, **String** libraryPath, **String** pin, **String** keyStorePassword, **String** searchPhase		Class constructor
PKCS11Instance	**String** tokenName, **String** libraryPath, **int** slot, **String** pin,		Class constructor

Method name	Parameter in	Return	Remark
	String keyStorePassword, **String** searchPhase		
getLibraryPath	-	**String**	คืนค่าตำแหน่งของไฟล์ .dll ของ PKCS11
setLibraryPath	**String** libraryPath,	**-**	กำหนดค่าตำแหน่งของไฟล์ .dll ของ PKCS11
getTokenName	-	**String**	คืนค่า Token name
setTokenName	**String** tokenName	**-**	กำหนดค่า Token name
getPin	-	**String**	คืนค่า Token Pin
setPin	**String** pin	**-**	กำหนดค่า Token Pin
getSlot	-	**int**	คืนค่า PKCS11 Slot
setSlot	**int** slot	**-**	กำหนดค่า PKCS11 Slot
getKeyStorePassword	-	**String**	คืนค่ารหัสผ่านของ KeyStore
setKeyStorePassword	**String** keyStorePassword	**-**	กำหนดค่ารหัสผ่านของ KeyStore
getSearchPhase	**-**	**String**	คืนค่าคำค้นหา Certificate
setSearchPhase	**String** searchPhase	**-**	กำหนดค่าคำค้นหา Certificate

Class PKCS12Instance

Method name	Parameter in	Return	Remark
PKCS12Instance	**String** filePath, **String** keyStorePassword		Class constructor
getFilePath		**String**	คืนค่าตำแหน่งของไฟล์ PFX, P12
setFilePath	**String** filePath		กำหนดค่าตำแหน่งของไฟล์ ไฟล์ PFX, P12
getKeyStorePassword		**String**	คืนค่ารหัสผ่านของ KeyStore
setKeyStorePassword	**String** keyStorePassword		กำหนดค่ารหัสผ่านของ KeyStore

Class XAdESBESController

Method name	Parameter in	Return	Remark
XadesBESController	**String** inputFileName, **String** outputFileName, **KeyingDataProvider**	-	Class constructor

	keyingDataProvider, **DigestAlgorithm** digestAlgorithm		
getSign	**Document** document, **KeyingDataProvider** keyingDataProvider, **DigestAlgorithm** digestAlgorithm	**Document**	เรียกใช้ Third-party เพื่อทำการ sign XML
loadKeyStore	-	**KeyStore**	อ่านค่า certificate และ private key จาก keystore

Class XAdESTController

Method name	Parameter in	Return	Remark
XadesTController	**String** inputFileName, **String** outputFileName, **KeyingDataProvider** keyingDataProvider, **DigestAlgorithm** digestAlgorithm, **TimeStamp** timeStamp	-	Class constructor
getSign	**Document** document, **KeyingDataProvider** keyingDataProvider, **DigestAlgorithm** digestAlgorithm, **TimeStamp** timeStamp	**Document**	เรียกใช้ Third-party เพื่อทำการ sign XML
getTSAHttpData	**TimeStamp** timeStamp	**TSAHttpData**	อ่านค่าการเชื่อมต่อ TSA

Class FileSpecification

Method name	Parameter in	Return	Remark
getFileNameWithoutExtension	-	**String**	คืนค่าชื่อไฟล์แบบไม่มีนามสกุล
setFileNameWithoutExtension	**String** fileNameWithoutExtension	-	กำหนดค่าชื่อไฟล์แบบไม่มีนามสกุล
getFileNameWithExtension	-	**String**	คืนค่าชื่อไฟล์พร้อมนามสกุล
setFileNameWithExtension	**String** fileNameWithExtension	**-**	กำหนดค่าชื่อไฟล์พร้อมนามสกุล
getFileExtension	-	**String**	คืนค่านามสกุลไฟล์
setFileExtension	**String** fileExtension	**-**	กำหนดค่านามสกุลไฟล์
getFullFilePath	-	**String**	คืนค่าตำแหน่งเต็มของไฟล์
setFullFilePath	**String** fullFilePath	**-**	กำหนดค่าตำแหน่งเต็มของไฟล์

Class DocumentIOHelper

Method name	Parameter in	Return	Remark
getXMLDocumentByPath	String path	**Document**	Load และ parse ไฟล์ XML
writeSignedFile	**Document** doc, **String** path	-	Write XML ที่ sign แล้วออกไปยัง path ที่กำหนด

Class TimeStamp

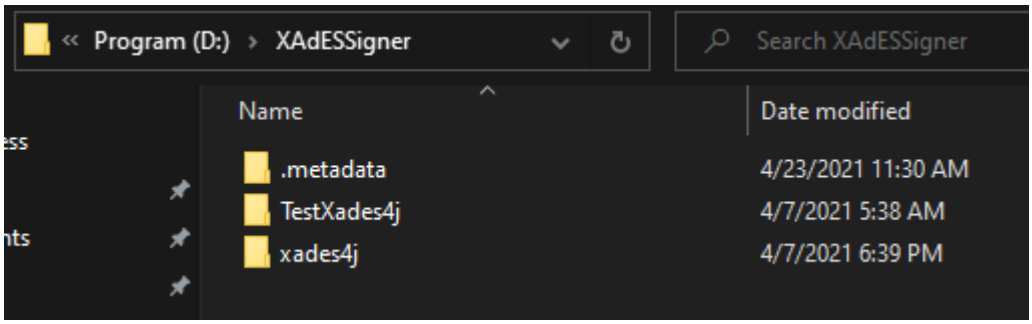
Method name	Parameter in	Return	Remark
TimeStamp	**TimeStampType** timeStampType	**-**	Class constructor
TimeStamp	**TimeStampType** timeStampType, **String** url, **TSAAuthenticationType** tsaAuthenticationType	-	Class constructor
TimeStamp	**TimeStampType** timeStampingType, **String** url, **TSAAuthenticationType** tsaAuthenticationType, **String** username, **String** password	**-**	Class constructor
TimeStamp	**TimeStampType** timeStampingType**,**		

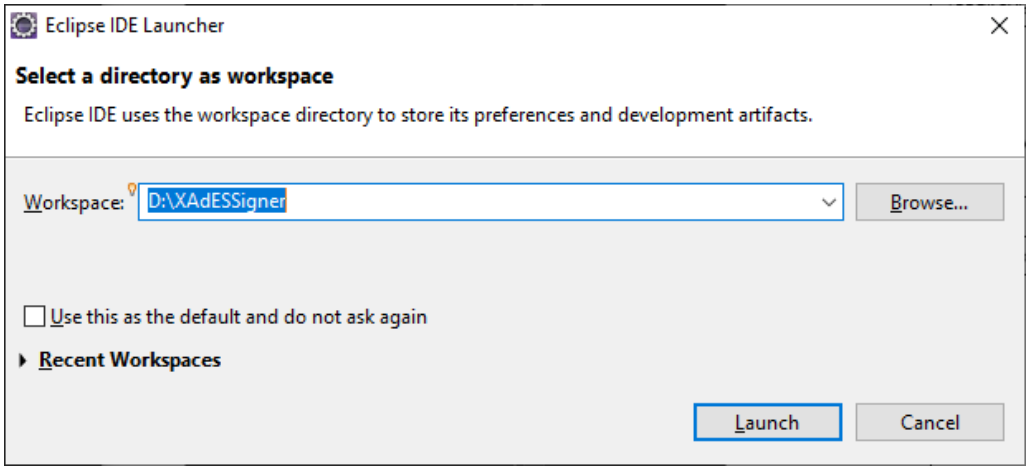
	String url, **TSAAuthenticationType** tsaAuthenticationType, **String** certificatePath, **char[]** certificatePassword		
getTimeStampingType	-	**TimeStampType**	คืนค่ารูปแบบ TimeStamp
setTimeStampingType	**TimeStampType** timeStampingType	**-**	กำหนดรูปแบบ TimeStamp
getTSAAuthenticationType	-	**TSAAuthenticationType**	คืนค่ารูปแบบ TSA Authentication
setTSAAuthenticationType	**TSAAuthenticationType** tsaAuthenticationType	**-**	กำหนดรูปแบบ TSA Authentication
getUsername	-	**String**	คืนค่าชื่อผู้ใช้
setUsername	**String** username	**-**	กำหนดชื่อผู้ใช้
getPassword	-	**String**	คืนค่ารหัสผ่าน
setPassword	**String** password	**-**	กำหนดรหัสผ่าน
getCertificatePath	-	**String**	คืนค่าตำแหน่ง certificate
setCertificatePath	**String** certificatePath	**-**	กำหนดตำแหน่ง certificate
getURL	-	**String**	คืนค่า URL
setURL	**String** uRL	**-**	กำหนด URL
getCertificatePassword	**-**	**char[]**	คืนค่า Password ของ Keystore
setCertificatePassword	**char[]** certificatePassword	**-**	กำหนดค่า Password ของ Keystore

การใช้งานและการ Deploy library

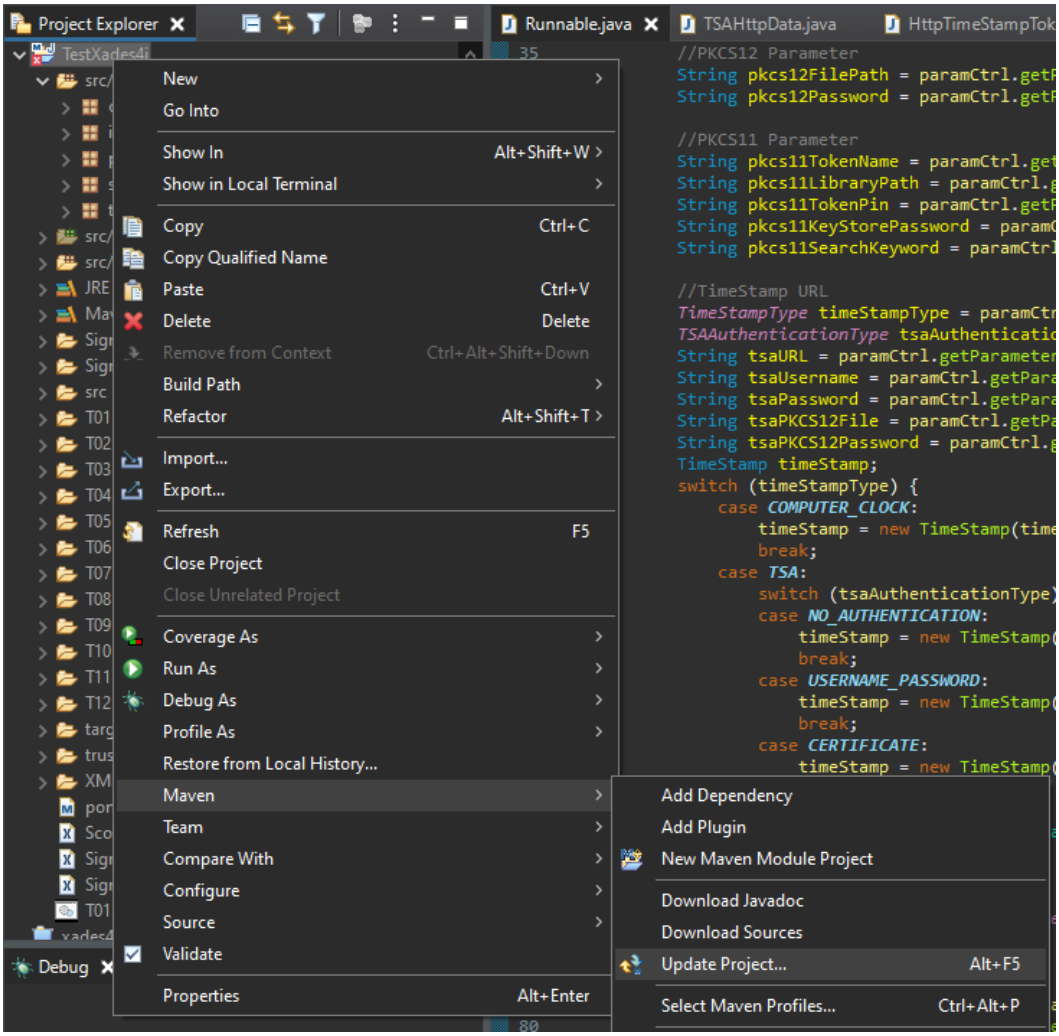
1. การเตรียม Project

1. เปิดโปรแกรม Eclipse และเลือกไปยังที่ตั้งของโฟลเดอร์ project (วิธีการสังเกตคือต้องมี folder .metadata ด้วยเสมอ)

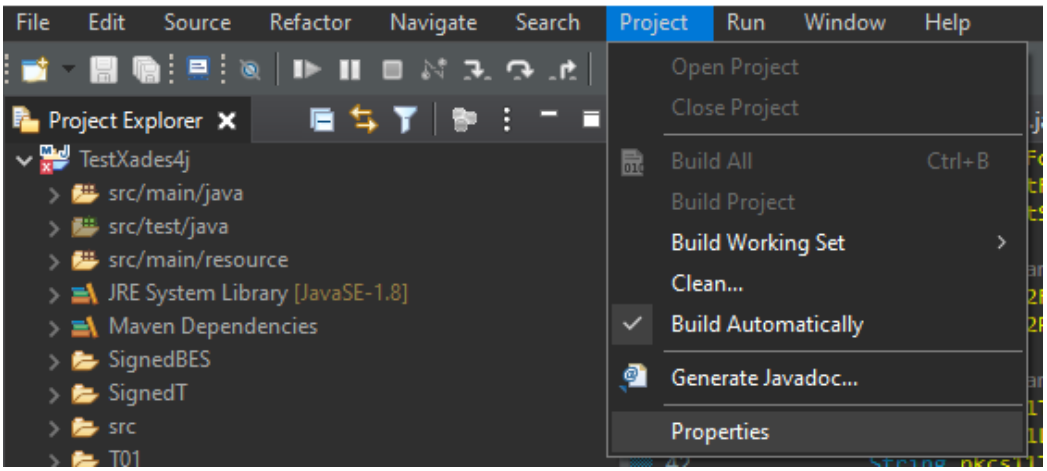




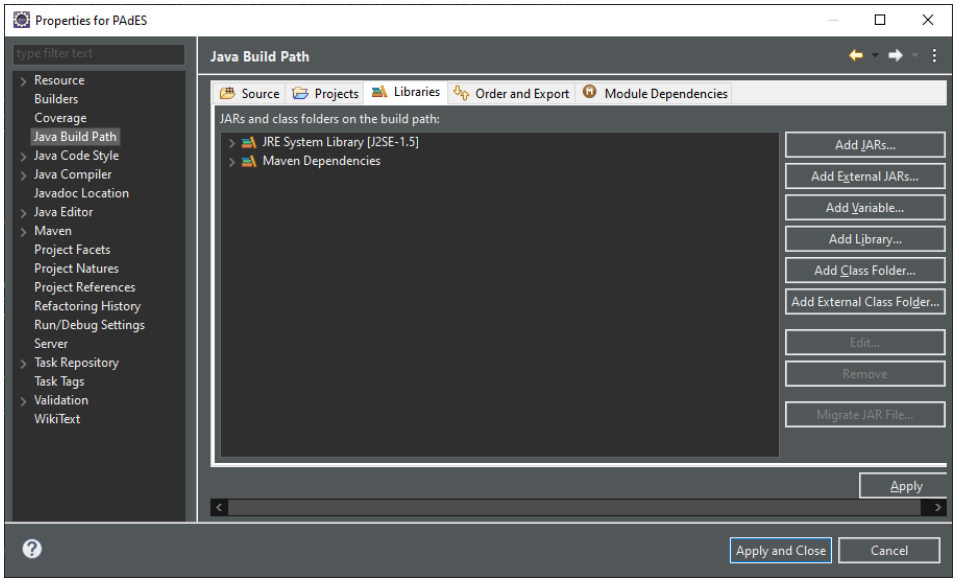
2. หลังจากเปิด project แล้ว ที่หน้าต่าง Project explorer ให้คลิกขวาที่ root folder ของ project แล้วเลือกไปที่ Maven > Update project จากนั้นรอจนกว่า project จะติดตั้ง dependency ที่จำเป็นเสร็จ



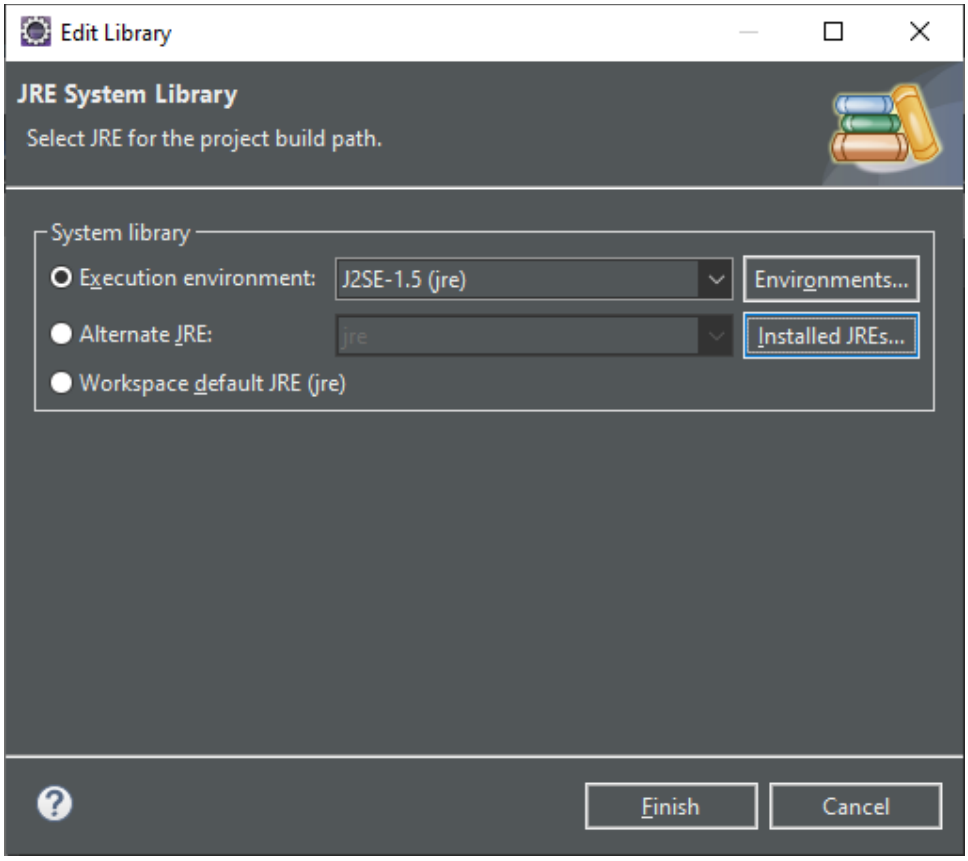
3. ที่ Menu bar เลือกที่ Project > Properties



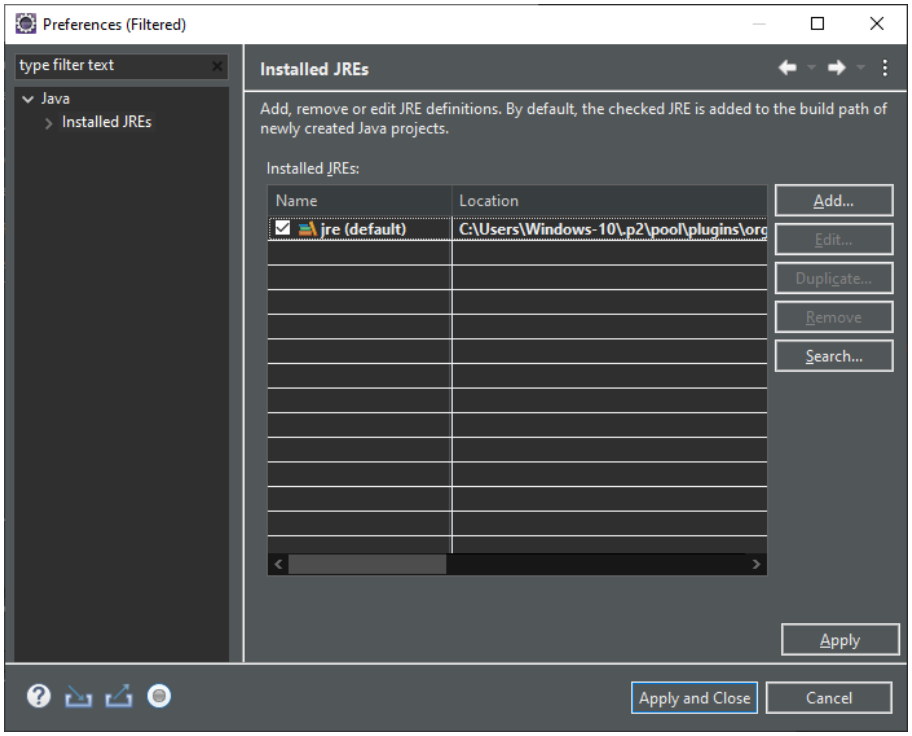
4. เลือกหัวข้อ Java build path จากนั้นเลือกที่ tab libraries แล้ว double click ที่ JRE System Library



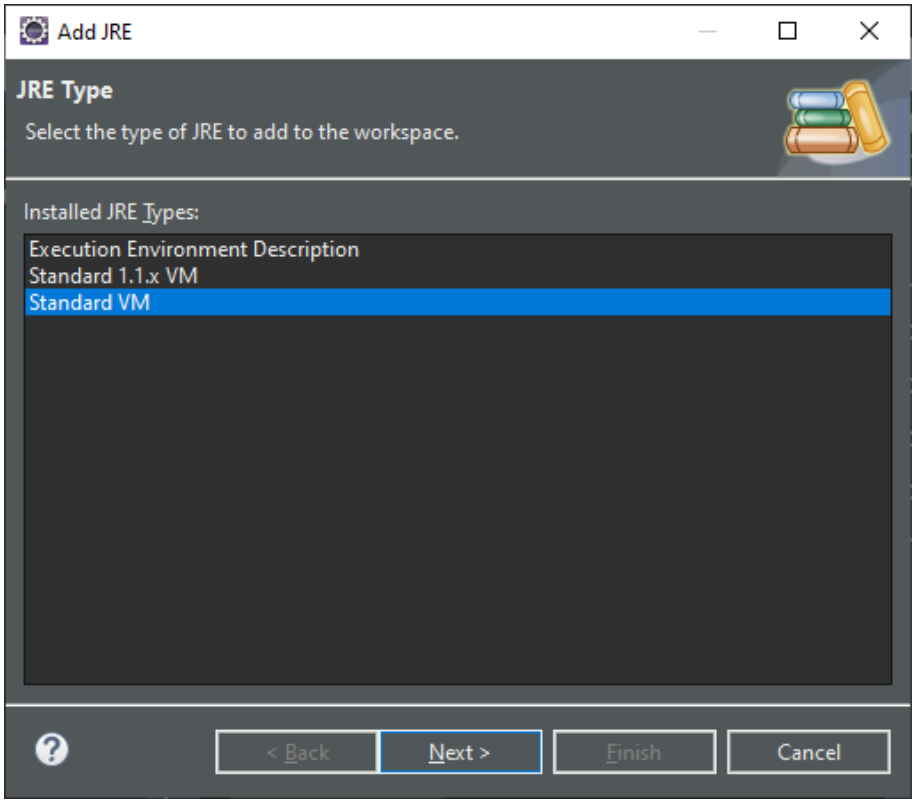
5. ที่หน้าต่าง Edit library เลือกที่ Installed JREs...



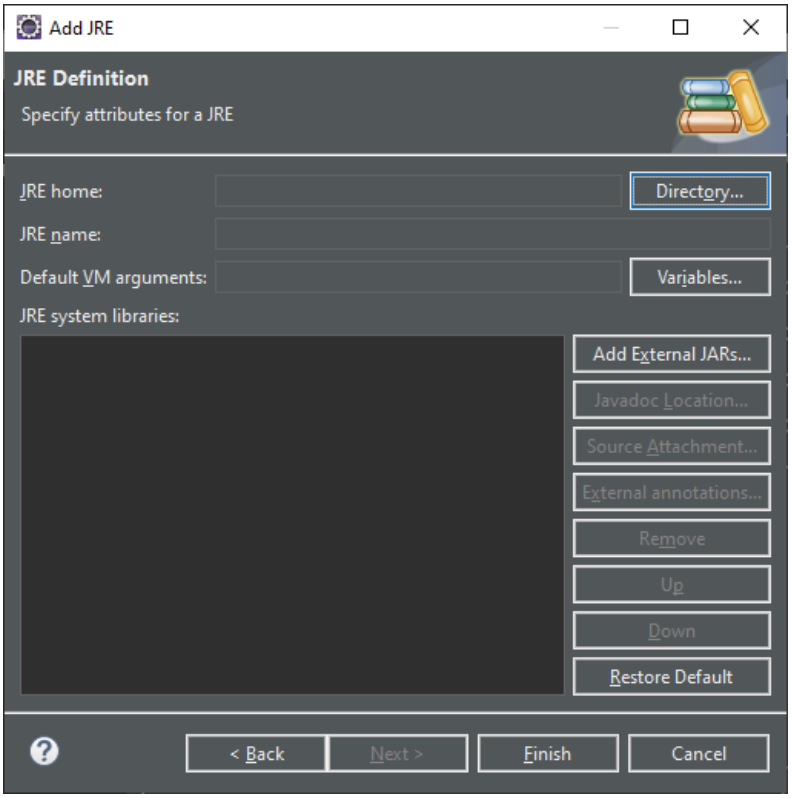
6. ที่หน้าต่าง Preferences (Filtered) เลือกที่ Add...



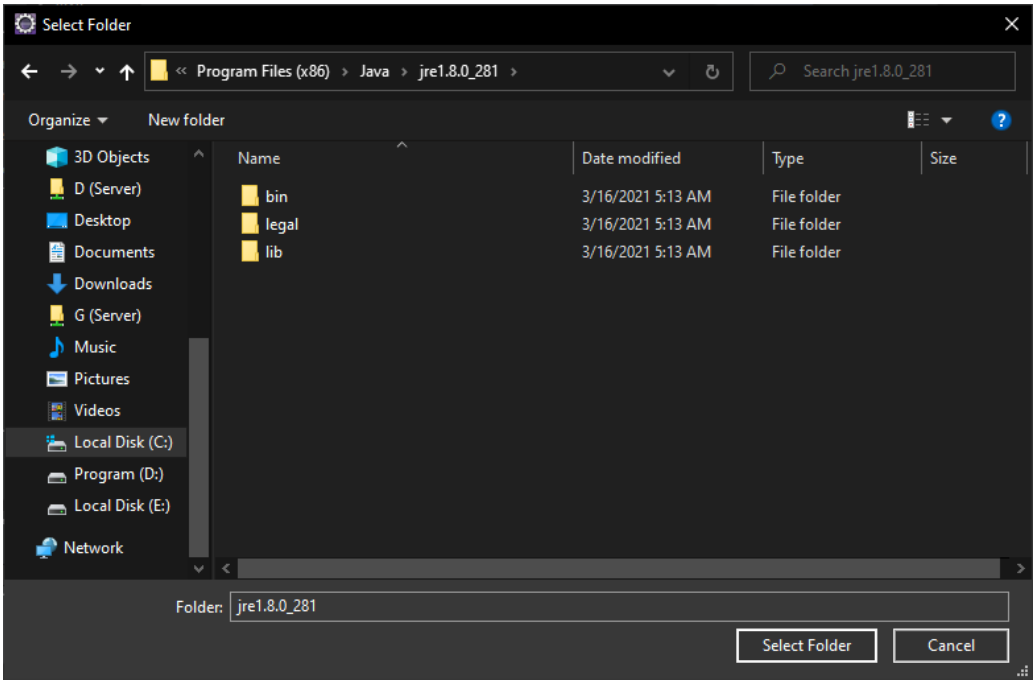
7. เลือก Standard VM จากนั้นกด Next



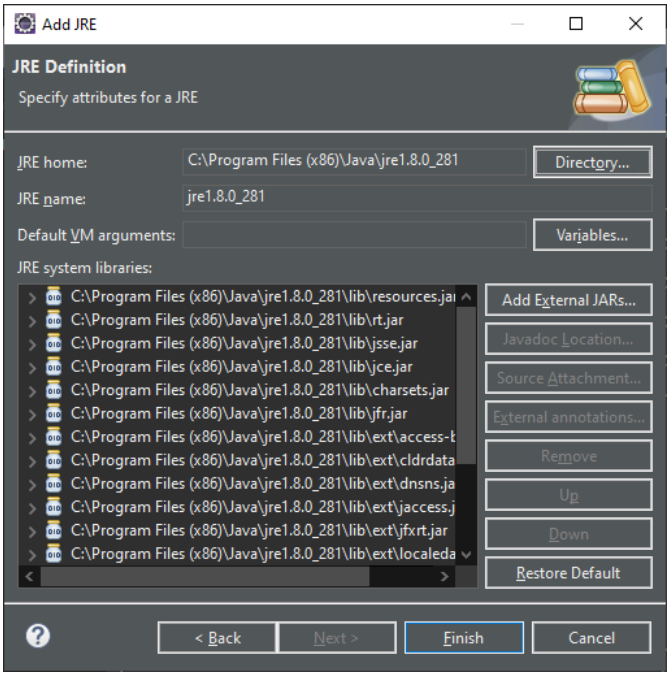
8. กดเลือก Directory...



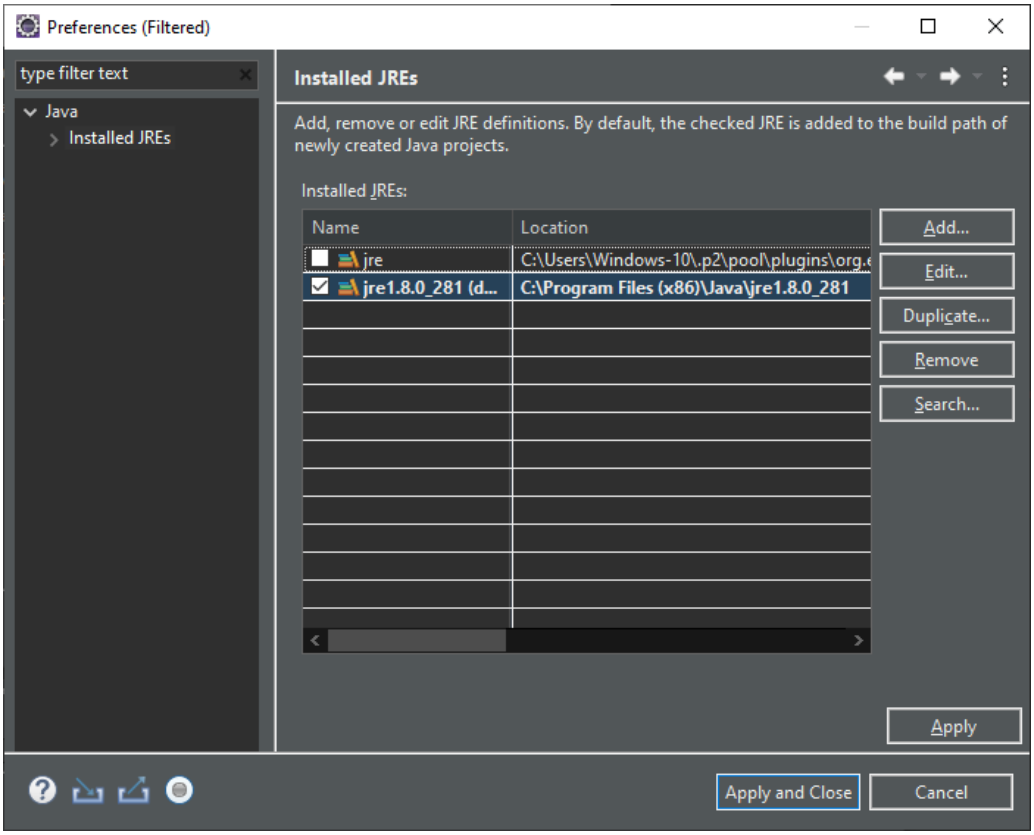
9. เลือกไปยังสถานที่ที่ติดตั้ง JRE 8 32-Bit ไว้ จากนั้นเลือก Select folder



10. ที่หน้าต่าง Add JRE จะปรากฏ System library ขึ้น กดปุ่ม Finish เพื่อยืนยันการเลือก

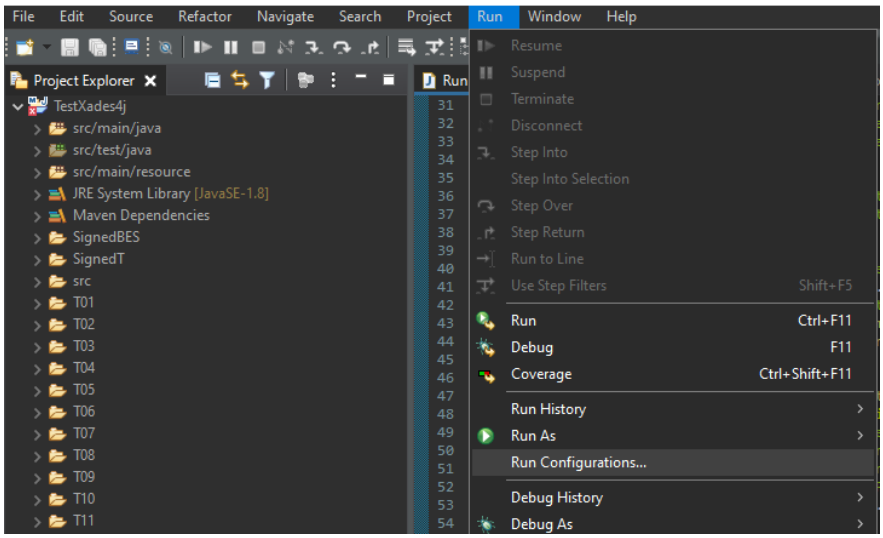


11. เมื่อกลับมาที่หน่าต่าง Preferences (Filtered) ให้เลือก JRE ที่เพิ่มเข้ามาใหม่ จากนั้นกด Apply and Close

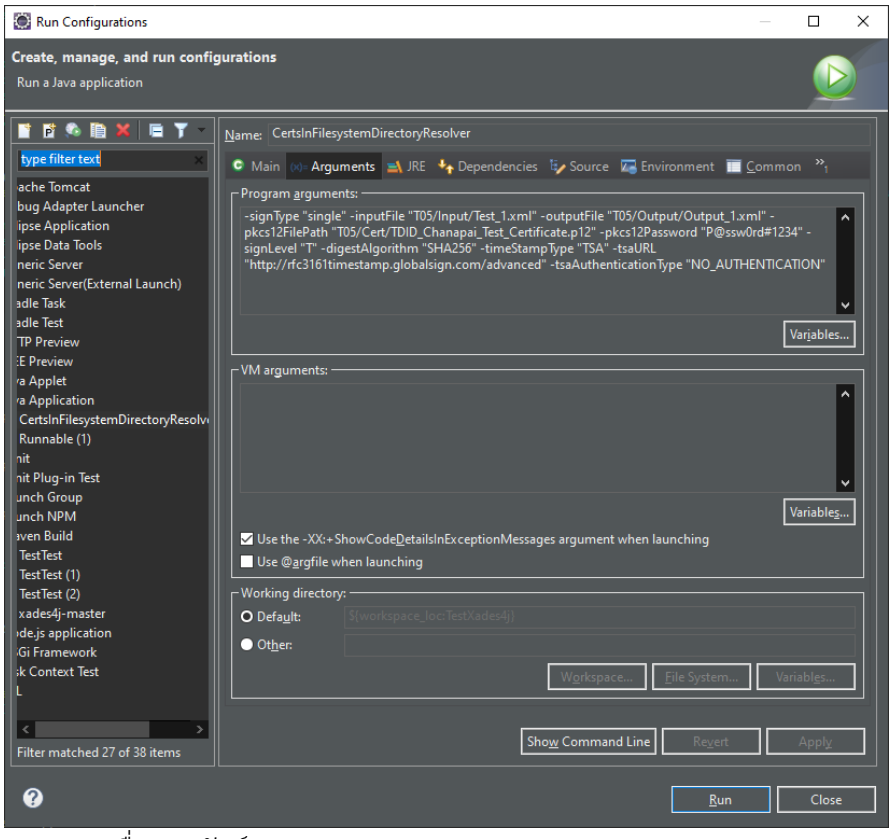


2. การเรียกใช้งานสำหรับการทดสอบ (Debug)

1. ที่ menu bar เลือกไปที่ Run > Run configurations...



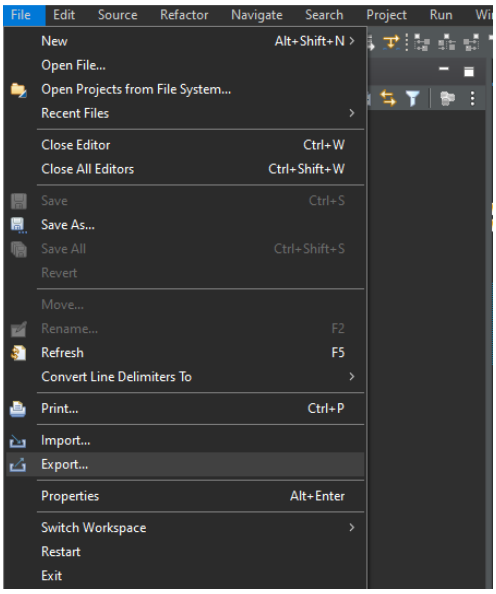
2. เลือกไปที่ Tab Arguments จากนั้นที่หัวข้อ Program arguments สามารถเปลี่ยนเป็น Argument ที่ต้องการทดสอบได้ (โดยรายละเอียดดูได้ที่หัวข้อการเรียกใช้งานผ่าน Command-line interface)



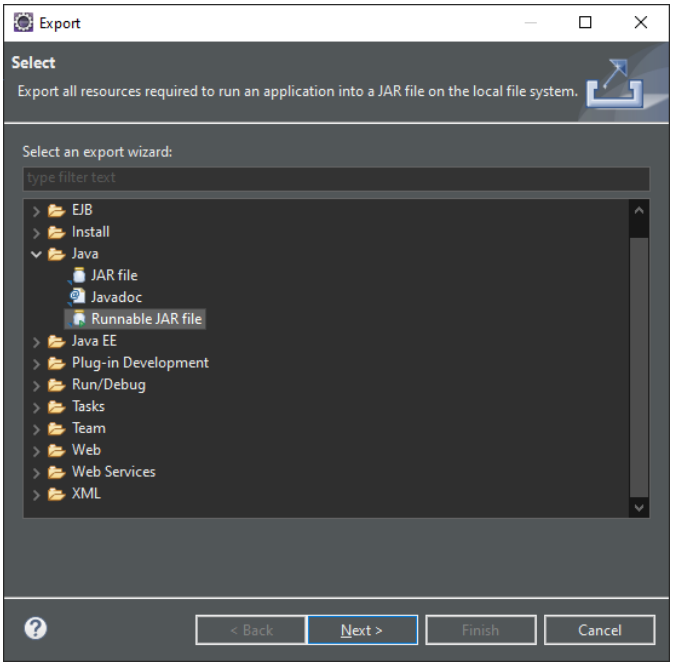
3. กด Run เพื่อดูผลลัพธ์

3. การ Deploy library (Executable jar)

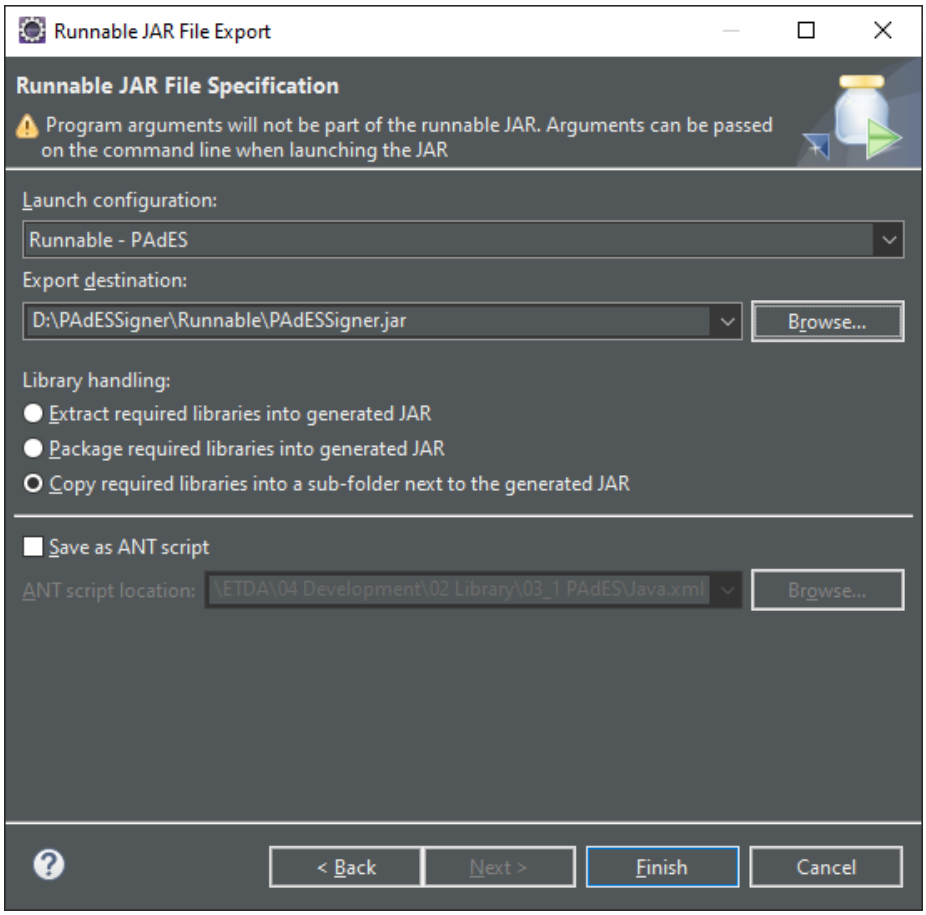
1. ที่ Menu bar เลือกไปที่ File > Export



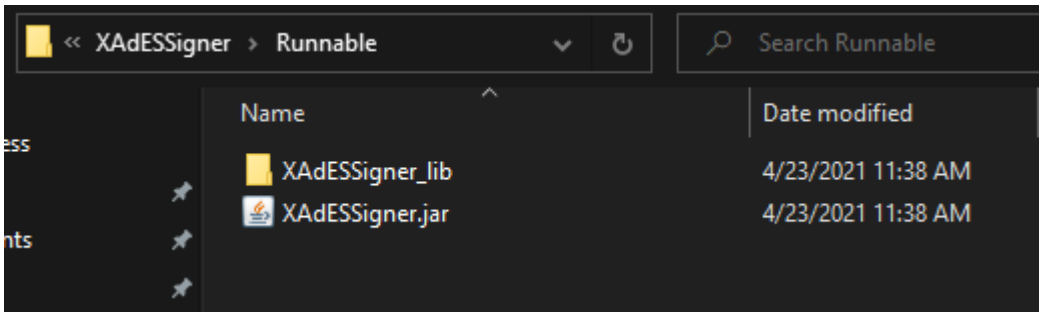
2. เลือกไปที่ Java > Runnable JAR file จากนั้นกด Next



3. ใส่ค่าต่าง ๆ ดังนี้
    - Launch configuration: เลือก Class ที่มี Method main()
    - Export destination: สถานที่สำหรับจัดเก็บ JAR ไฟล์ที่สร้างเสร็จแล้ว
    - Library handling: กำหนดรูปแบบการ Package JAR โดยแนะนำให้เลือกเป็น Copy required libraries into a sub-folder next to the generated JAR



4. กด Finish
5. โดยไฟล์ JAR และ Dependency ที่เกี่ยวข้อง จะถูกจัดเก็บใน folder ที่กำหนด



การเรียกใช้งานผ่าน Command-line interface

1. รายละเอียด Argument

ชุดโปรแกรมรองรับการเรียกใช้งานผ่าน Command line สำหรับ Executable program โดยมี Parameter ที่สามารถ input ค่าได้ ดังนี้

- **-signType "single | multiple"** กำหนดการ sign แบบไฟล์เดียวหรือหลายไฟล์
- **-signLevel “BES | T”** กำหนดระดับของ XAdES
- **-inputFile "<PATH_TO_FILE>"** กำหนดไฟล์นำเข้า
- **-outputFile "<PATH_TO_FILE>"** กำหนดไฟล์ผลลัพธ์
- **-inputFolder "<PATH_TO_FOLDER>"** กำหนดโฟลเดอร์นำเข้า
- **-outputFolder ""<PATH_TO_FOLDER>"** กำหนดโฟลเดอร์ผลลัพธ์
- **-pkcs11TokenName "<NAME>"** กำหนดชื่อ PKCS11 Token
- **-pkcs11LibraryPath "<PATH_TO_FILE>"** กำหนดชื่อ PKCS11 Token
- **-pkcs11Pin "<PASSWORD>"** Password ของ Token
- **-pkcs11KeyStorePassword "<PASSWORD>"** password ของ Ketstore
- **-pkcs11SeachKeyword "<ANY_TEXT>"** คำค้นหา Certificate ใน Token
- **-pkcs12FilePath "<PATH_TO_FILE>"** ตำแหน่งของไฟล์ P12, PFX
- **-pkcs12Password "<PASSWORD>"** Password ของไฟล์ P12, PFX
- **-timeStampType "TSA | COMPUTER_CLOCK"** รูปแบบ Timestamp
- **-tsaURL "<URL>"** URL ของ TSA
- **-tsaAuthenticationType "<NO_AUTHENTICATION | USERNAME_PASSWORD | CERTIFICATE>"** รูปแบบการ Authentication ของ TSA
- **-tsaUsername "<USERNAME>"** Username ของ TSA
- **-tsaPassword "<PASSWORD>"** Password ของ TSA
- **-digestAlgorithm "<SHA256 | SHA384 | SHA512>"** กำหนด hash function

2. ตัวอย่างการเรียกใช้งาน

- กรณี Sign XML แบบ XAdES Baseline-B ไฟล์เดียว

```
java -jar XAdESSigner.jar -signType "single" -inputFile "Input.xml" -outputFile "
Output.xml" -pkcs12FilePath "User_Certificate.p12" -pkcs12Password "password" -
signLevel "BES" -timeStampType "COMPUTER_CLOCK" -digestAlgorithm "SHA256"
```

- กรณี Sign XML แบบ XAdES Baseline-B แบบ Bulk (Sign ทั้ง Folder)

```
java -jar XAdESSigner.jar -signType "single" -inputFolder "Input/
-outputFolder "Output/" -outputSuffix "_BES" -pkcs12FilePath
"User_Certificate.p12"
-pkcs12Password "password" -signLevel "BES" -timeStampType
"COMPUTER_CLOCK"
-digestAlgorithm "SHA256"
```

- กรณี Sign XML แบบ XAdES Baseline-T ไฟล์เดียว

```
java -jar XAdESSigner.jar -signType "single" -inputFile "Input.xml" -outputFile "
Output.xml" -pkcs12FilePath "User_Certificate.p12" -pkcs12Password "password"
-signLevel "T" -timeStampType "TSA" -tsaURL "https://TSA_URL"
-tsaAuthenticationType "CERTIFICATE" -tsaPKCS12File "TSA_Certification.p12"
-tsaPKCS12Password "password" -digestAlgorithm "SHA256"
```

- กรณี Sign XML แบบ XAdES Baseline-T แบบ Bulk (Sign ทั้ง Folder)

```
java -jar XAdESSigner.jar -signType "single" -inputFolder "Input/
-outputFolder "Output/" -outputSuffix "_BES" -pkcs12FilePath
"User_Certificate.p12"
-pkcs12Password "password" -signLevel "T" -timeStampType "TSA"
-tsaURL https://TSA_URL -tsaAuthenticationType "CERTIFICATE" -tsaPKCS12File
"TSA_Certification.p12" -tsaPKCS12Password "password" -digestAlgorithm
"SHA256"
```