TRIDIUM

NS2022

ACCELERATING INNOVATION

Niagara Development

Garrett L. Ward @ Tridium

TRIDIUM

Overview

TRIDIUM

Developer Environment Setup

- Niagara 4 Developer
- Java 8 JDK
- node.js LTS
- IDE/Text Editor

Java 8

- ‘tools.jar + javac’ is sufficient for simply building
- Full JDK needed for (most) IDE integrations
- Available from various vendors:
  - ~~AdoptOpenJDK~~ ~~Adoptium~~ “Eclipse Temurin by Adoptium”
  - Azul Zulu Community™
  - Amazon Coretto
  - …

Gradle

- https://gradle.org
- Niagara Gradle plugins
- Gradle wrapper
- Significant update in Niagara 4.13
  - Gradle 4.0.2 -> Gradle 7.x

Niagara Module

- Standard jar
  - Niagara module manifest ‘module.xml’
  - Jar signature
- Module Parts
  - ‘module’ is logical grouping
  - ‘runtime profile’ indicates (roughly) purpose
  - ‘module part’ is actual .jar

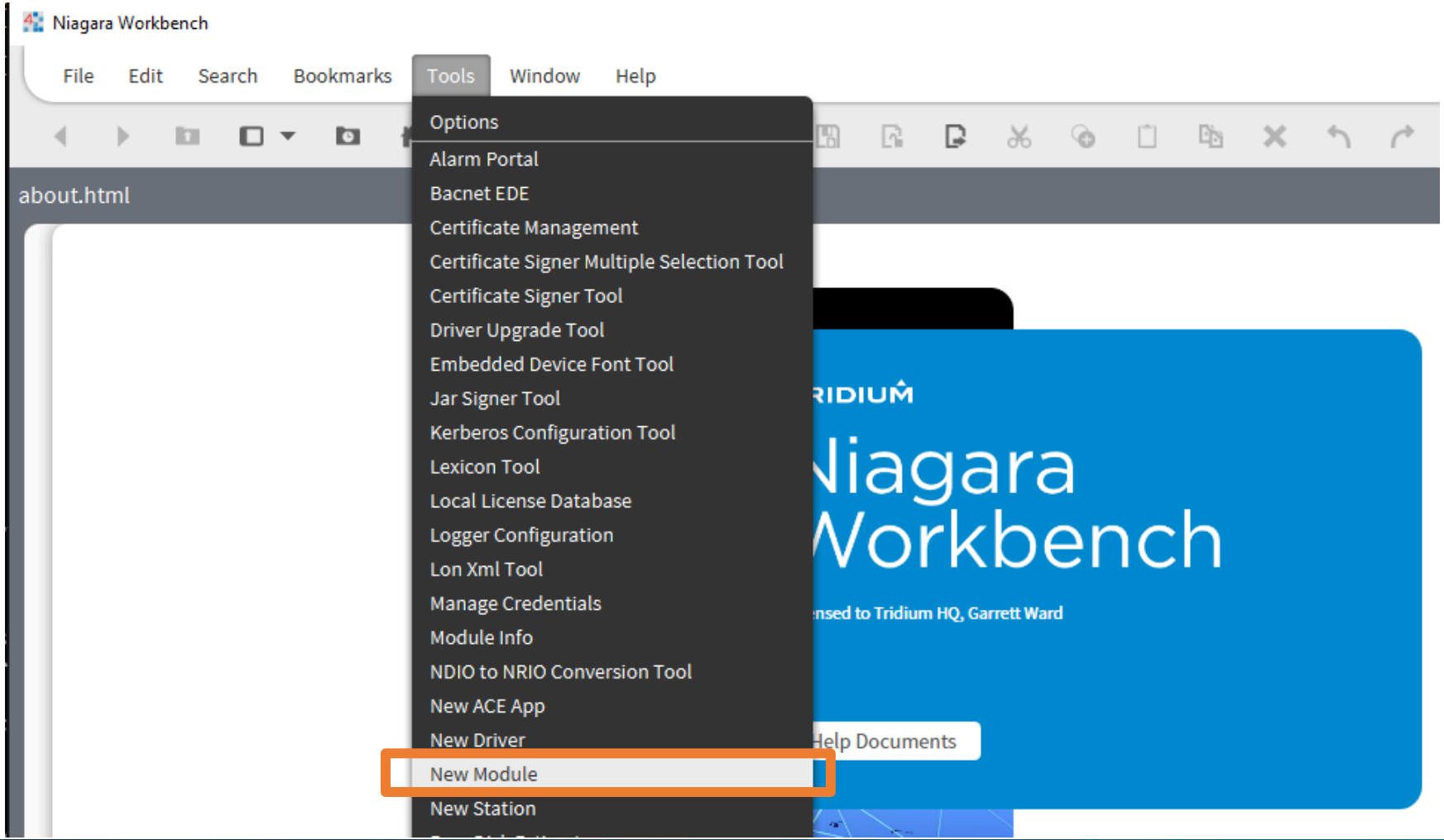
Example: Basic Module Build

TRIDIUM

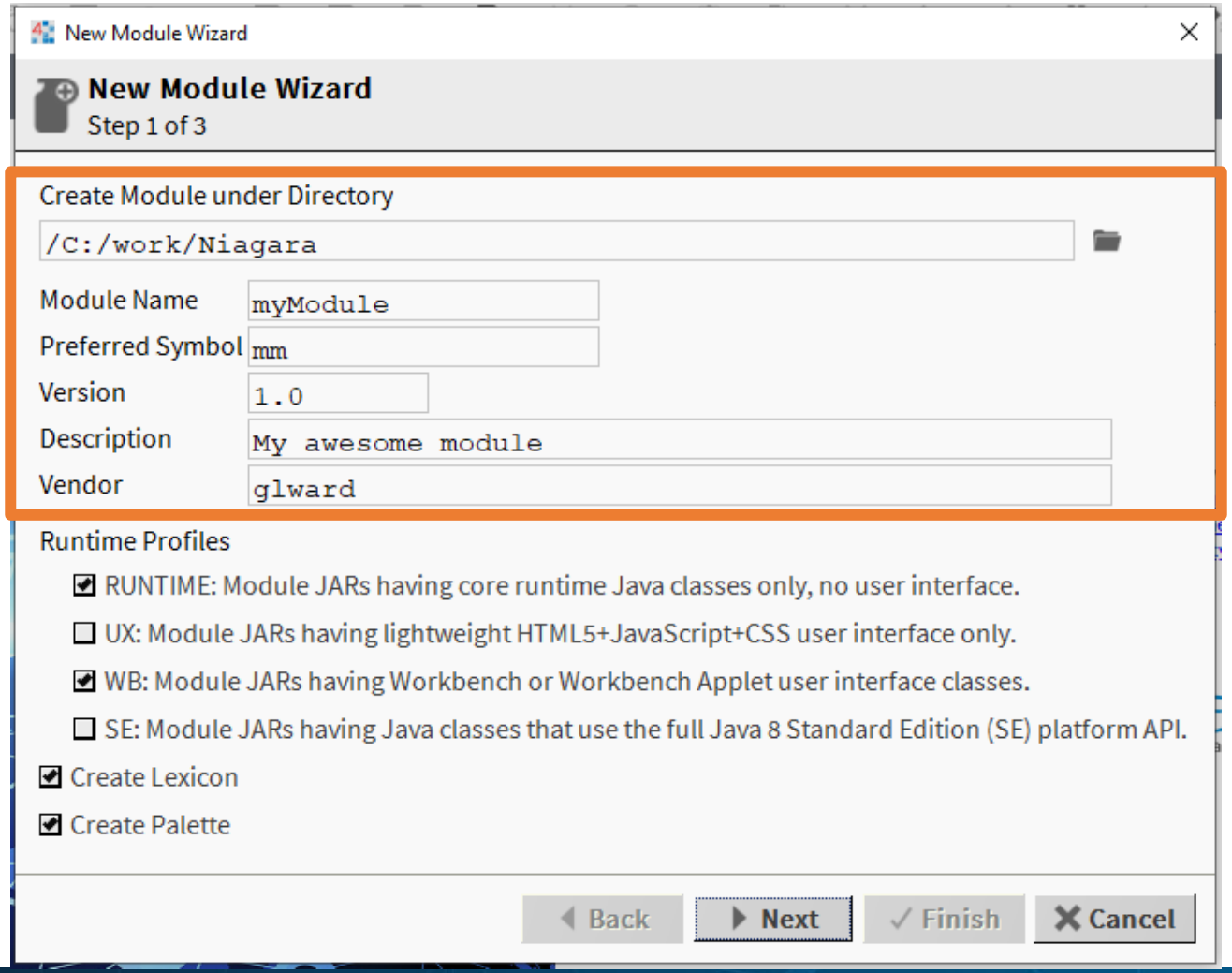
Building Niagara Modules

- Create the module with the New Module Wizard
- Add a sample BComponent
- Add a sample BTestNg test
- Run Slot-o-matic on it
- Build it
- Test it

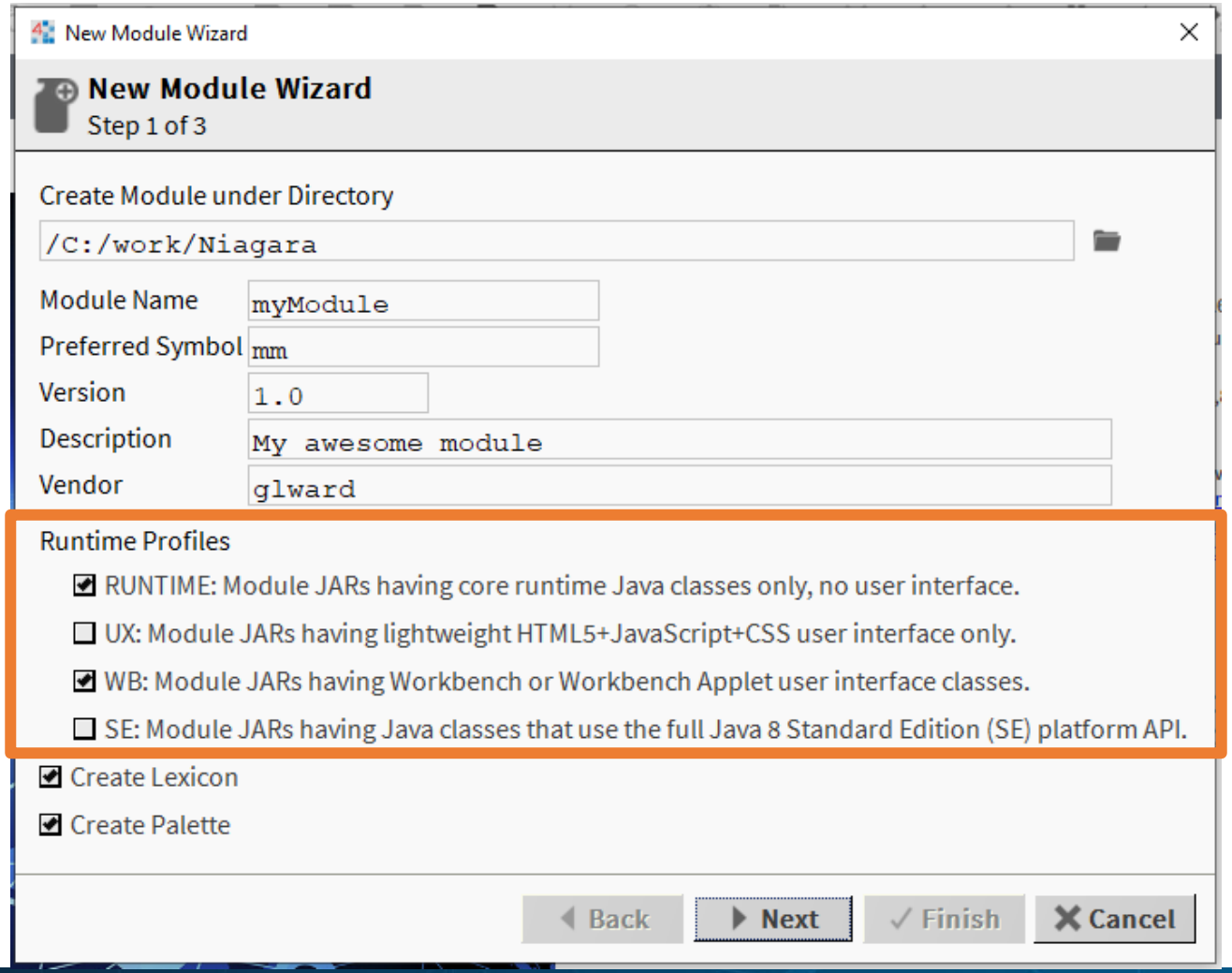
New Module Wizard



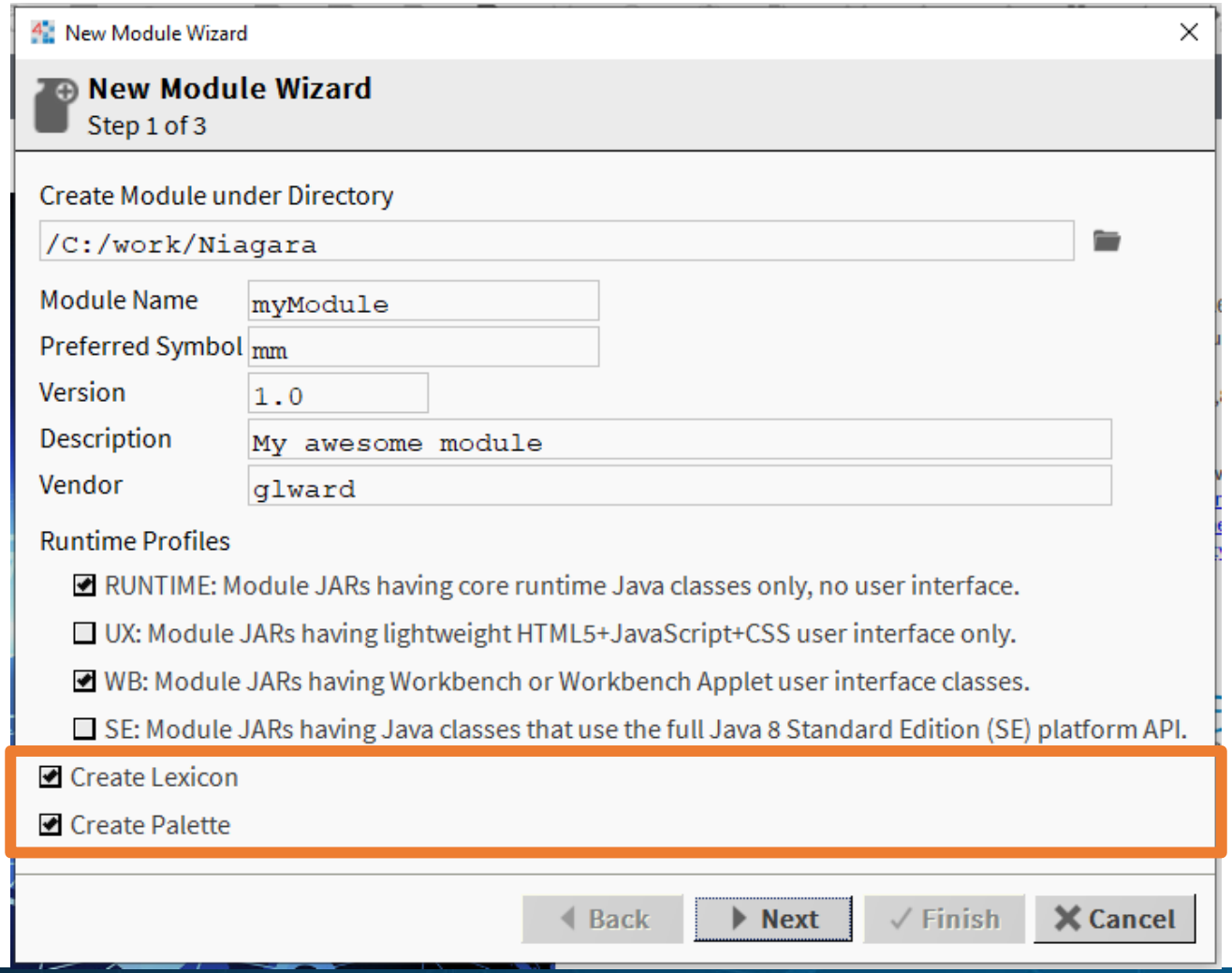
New Module Wizard



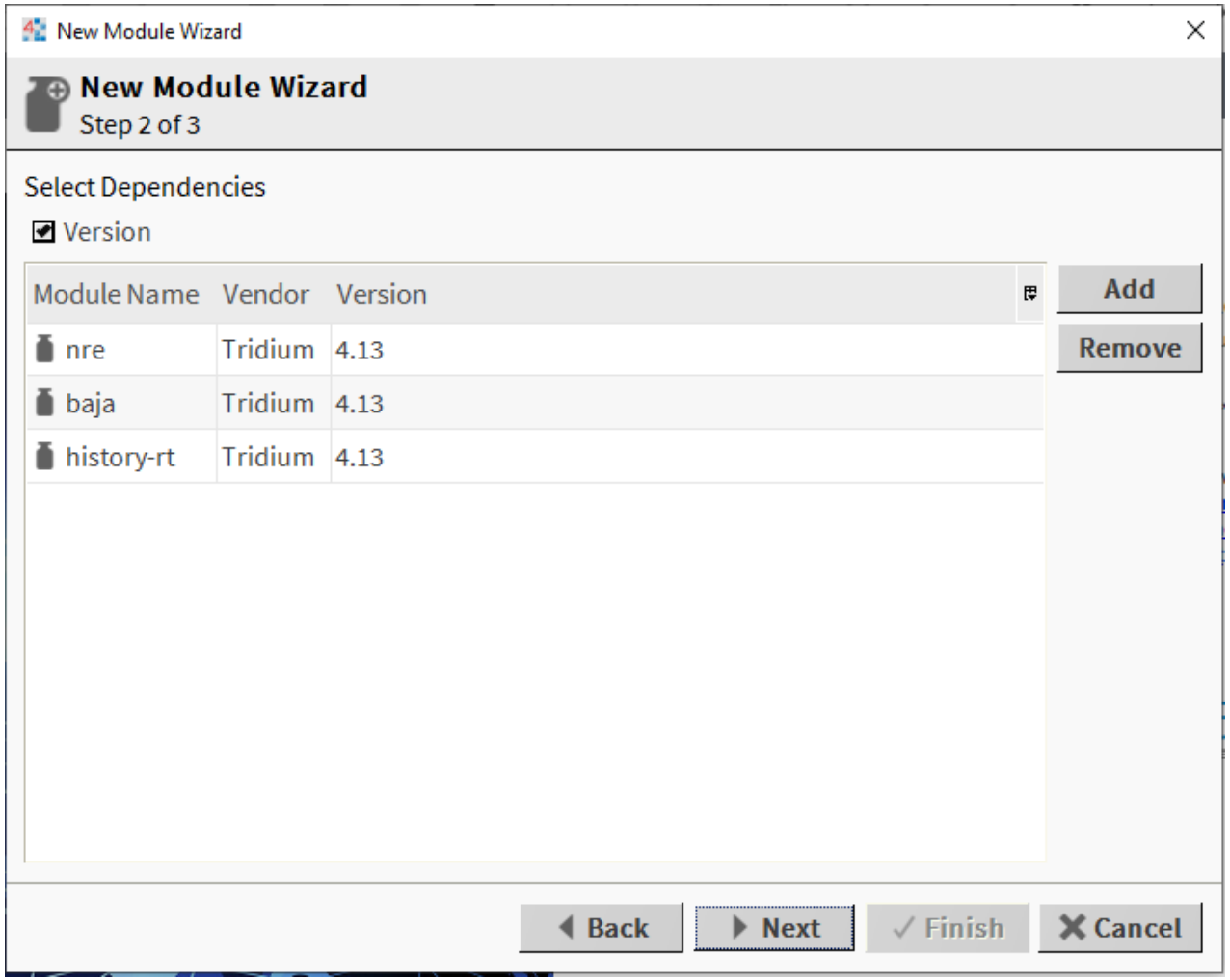
New Module Wizard



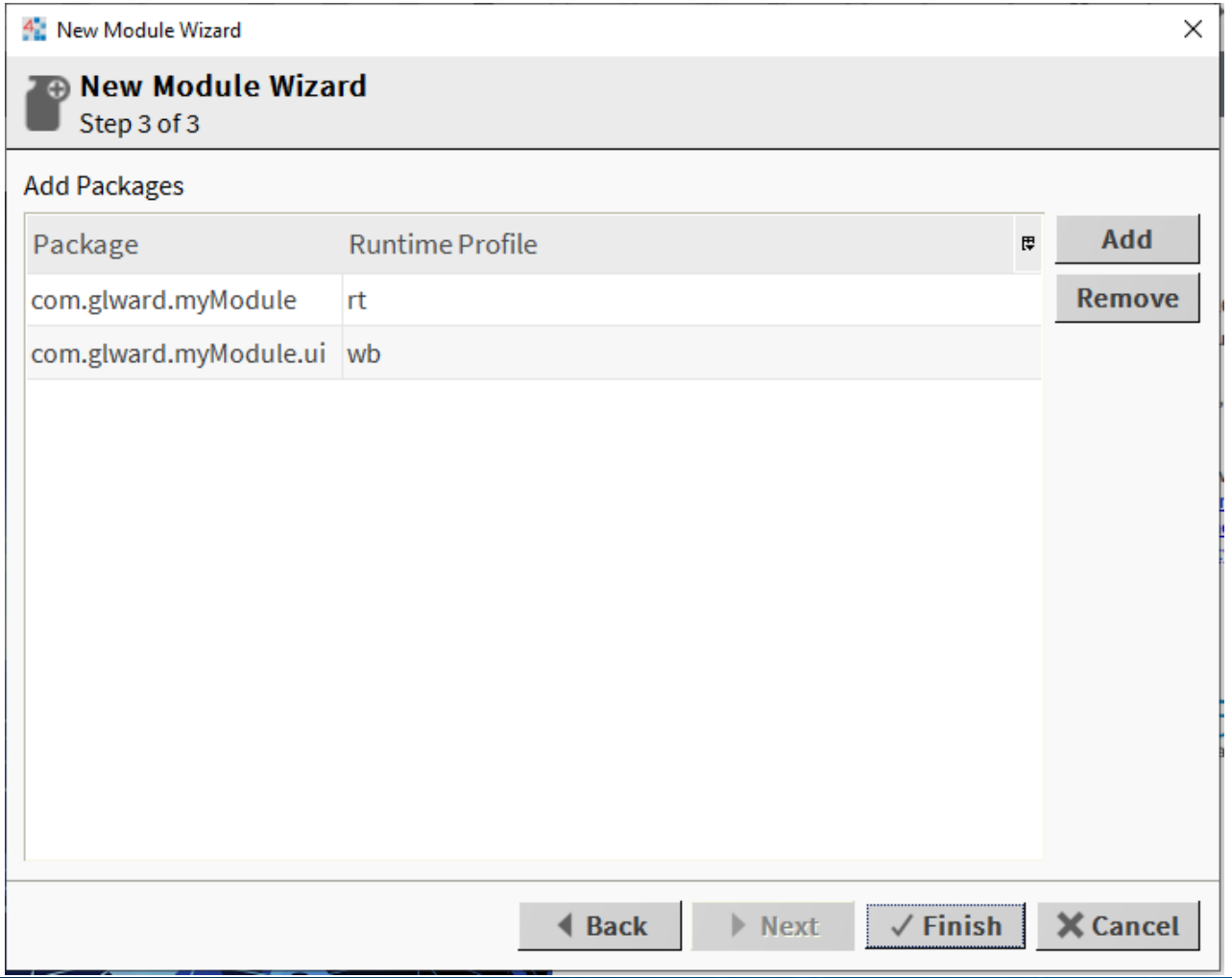
New Module Wizard



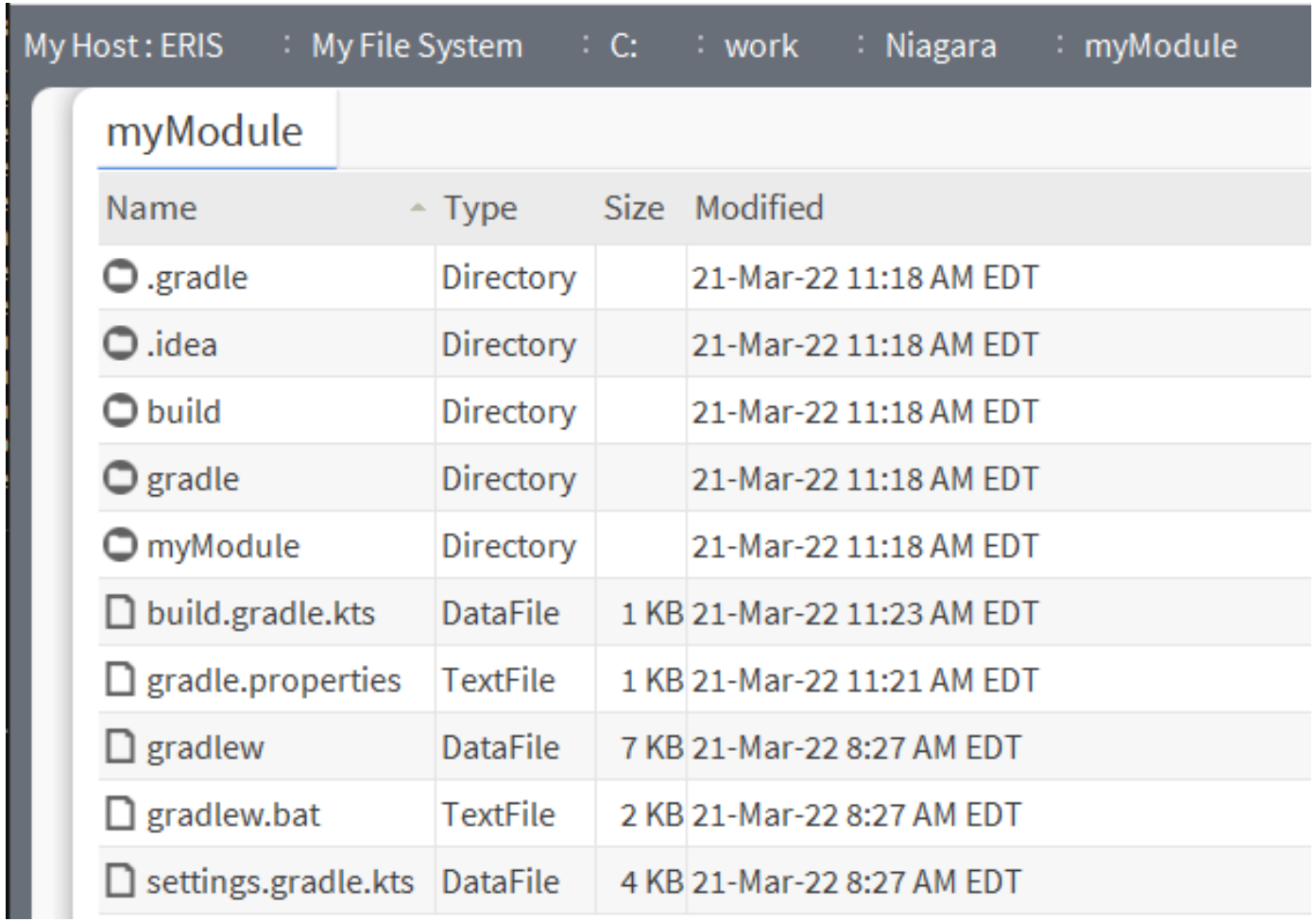
New Module Wizard



New Module Wizard



Build Environment Files – Root



settings.gradle.kts

```
/*
 * Copyright 2022 glward, All Rights Reserved.
 */

import com.tridium.gradle.plugins.settings.MultiProjectExtension

pluginManagement {
  // lots of magic

  val gradlePluginHome: String = "magic"
  val gradlePluginRepoUrl = "file:///${gradlePluginHome.replace('\\', '/')}"

  val gradlePluginVersion: String = "7.3.5-SNAPSHOT"
  val settingsPluginVersion: String = "7.3.0"
```

settings.gradle.kts

```
  repositories {
    maven(url = "$gradlePluginRepoUrl")
    gradlePluginPortal()
  }

  plugins {
    id("com.tridium.settings.multi-project") version (settingsPluginVersion)
    id("com.tridium.settings.local-settings-convention") version (settingsPluginVersion)

    id("com.tridium.niagara") version (gradlePluginVersion)
    id("com.tridium.vendor") version (gradlePluginVersion)
    id("com.tridium.niagara-module") version (gradlePluginVersion)
    id("com.tridium.niagara-signing") version (gradlePluginVersion)

    id("com.tridium.convention.niagara-home-repositories") version (gradlePluginVersion)
  }
}
```

settings.gradle.kts

```
plugins {
  // Discover all subprojects in this build
  id("com.tridium.settings.multi-project")
}

configure<MultiProjectExtension> {
  findProjects()
}
```

build.gradle.kts

```
/* Copyright 2022… */

plugins {
  // Base Niagara plugin
  id("com.tridium.niagara")

  // The vendor plugin provides the vendor {} extension …
  id("com.tridium.vendor")

  // The signing plugin configures signing …
  id("com.tridium.niagara-signing")

  // The niagara_home repositories convention plugin configures !bin/ext and !modules…
  id("com.tridium.convention.niagara-home-repositories")
}
```

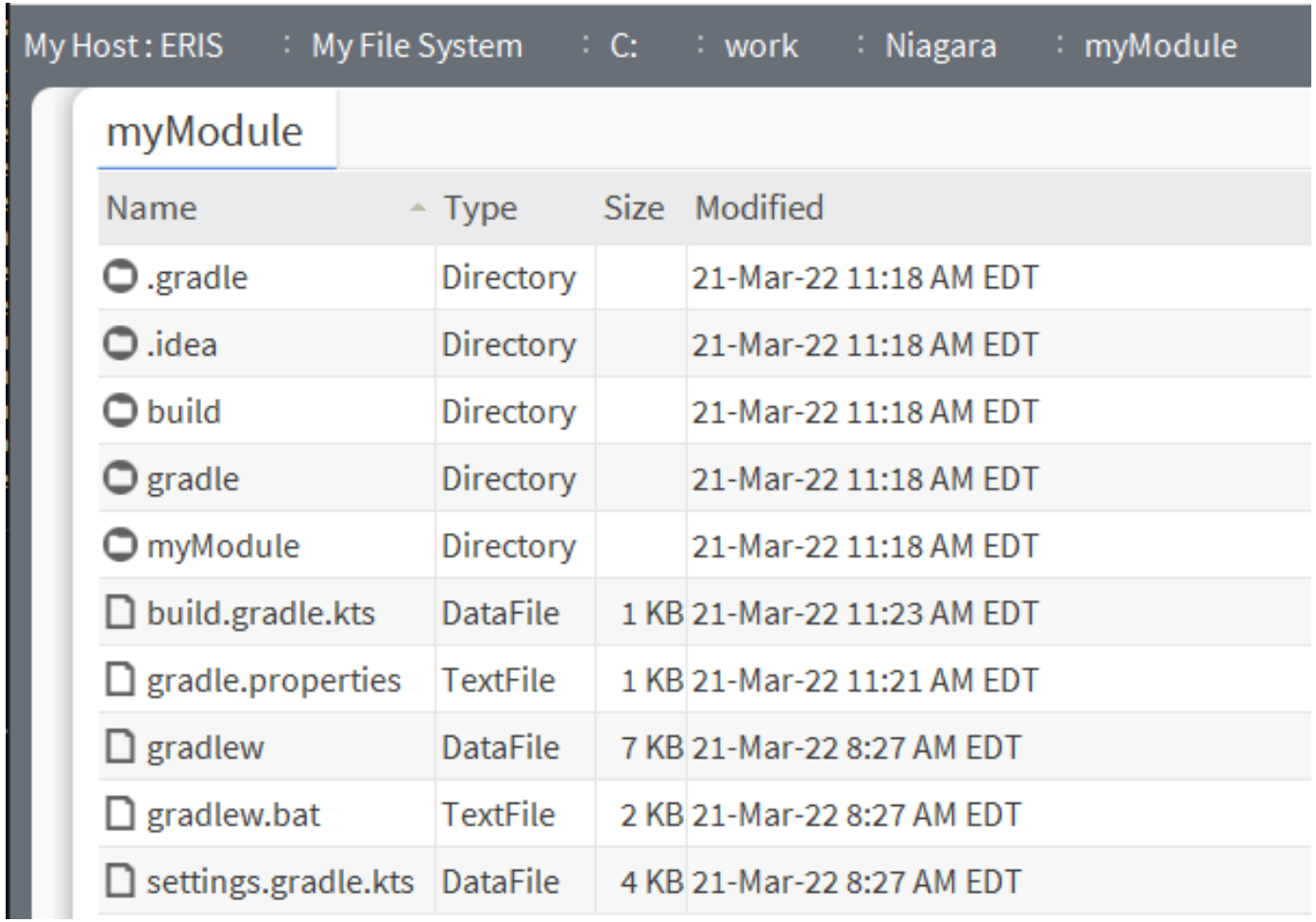
build.gradle.kts

```
vendor {
  // defaultVendor sets the "vendor" attribute on module and dist files; it's
  // what's shown in Niagara when viewing a module or dist.
  defaultVendor("glward")

  // defaultModuleVersion sets the "vendorVersion attribute on a module.
  defaultModuleVersion("1.0")
}

subprojects {
  repositories {
    mavenCentral()
  }
}
```

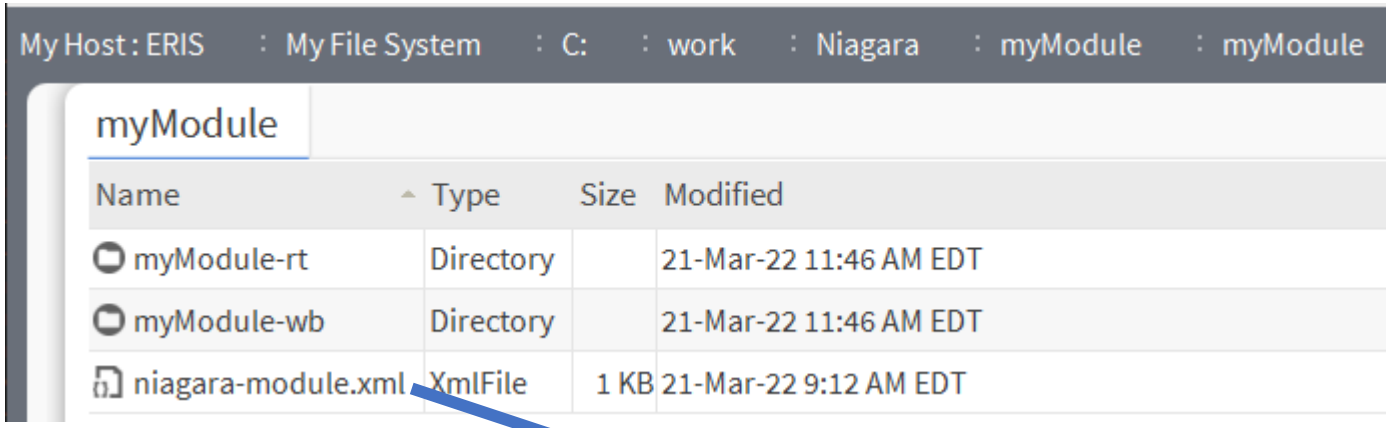
gradle.properties

```
gradlePluginHome=C:\\work\\developer-4.13.0.0
niagara_home=C:\\niagara\\r4dev\\niagara\\niagara_home
niagara_user_home=C:\\niagara\\r4dev\\niagara\\niagara_user_home
nodeHome=C:\\Program Files\\nodejs
#AUTOMATICALLY GENERATED BY GRADLE -- DO NOT MODIFY
org.gradle.java.installations.auto-detect=false
#AUTOMATICALLY GENERATED BY GRADLE -- DO NOT MODIFY
org.gradle.java.installations.auto-download=false
#AUTOMATICALLY GENERATED BY GRADLE -- DO NOT MODIFY
org.gradle.java.installations.paths=C:\\niagara\\r4dev\\niagara\\niagara_home\\jre
```

Build Environment Files – Root

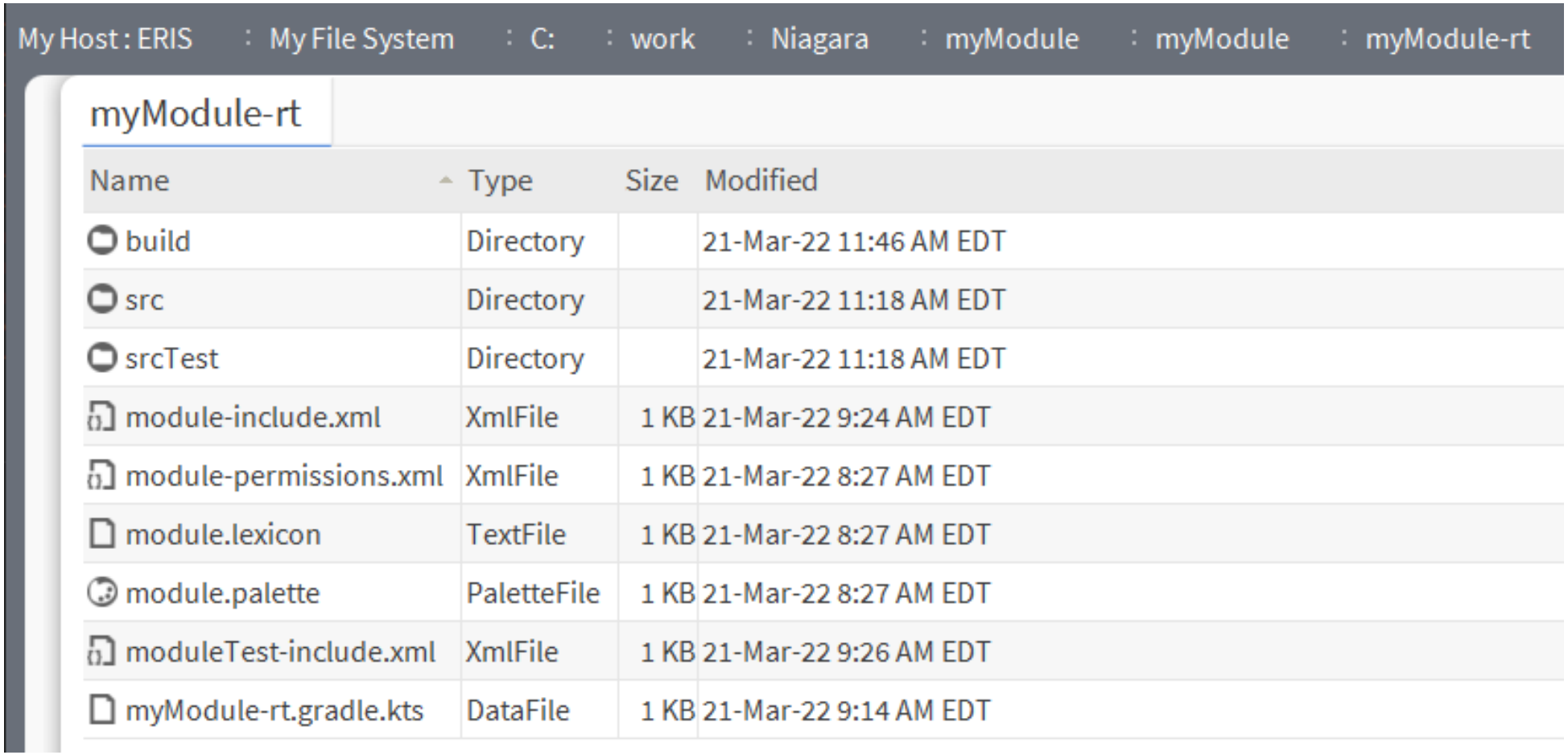


Build Environment Files - Module




```
<?xml version="1.0" encoding="UTF-8"?>
<niagara-module
  moduleName="myModule"
  preferredSymbol="mm"
  runtimeProfiles="rt,wb"
/>
```

Build Environment Files – Module Part



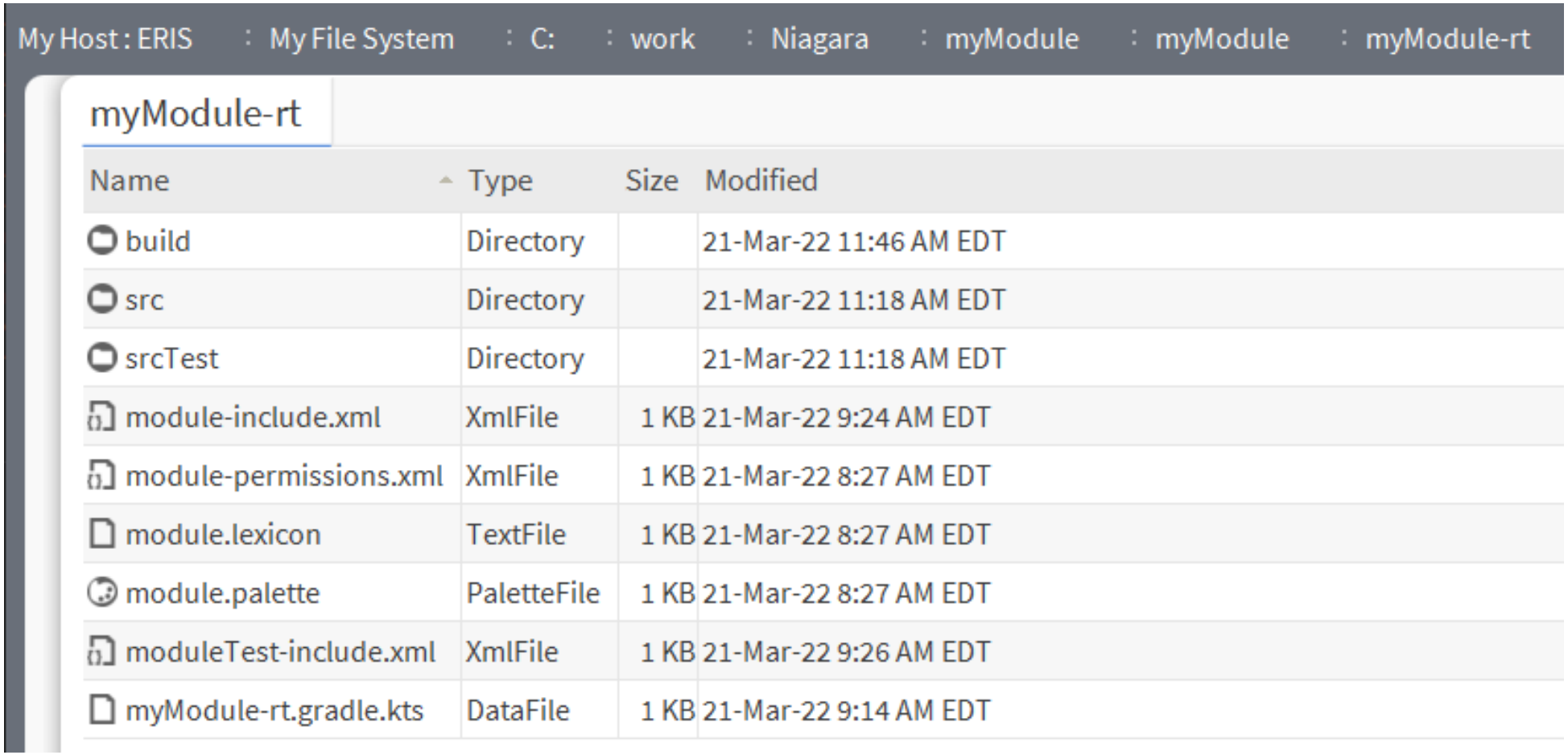
myModule-rt.gradle.kts

```
/* Copyright 2022 … */
import com.tridium.gradle.plugins.module.util.ModulePart.RuntimeProfile.*

plugins {
  id("com.tridium.niagara-module")
  id("com.tridium.niagara-signing")
  id("com.tridium.bajadoc")
  id("com.tridium.niagara-jacoco")
  id("com.tridium.niagara-annotation-processors")
  id("com.tridium.convention.niagara-home-repositories")
}

description = "My awesome module"

moduleManifest {
  moduleName.set("myModule")
  runtimeProfile.set(rt)
}
```

myModule.gradle.kts

```
// See documentation in docDeveloper for the supported
// dependency types
dependencies {
  // NRE dependencies
  nre("Tridium:nre")

  // Niagara module dependencies
  api("Tridium:history-rt")
  api("Tridium:baja")

  // Test Niagara module dependencies
  moduleTestImplementation("Tridium:test-wb")
}
```

Build Environment Files – Module Part



Example BComponent

```
package com.glward.myModule;

import javax.baja.nre.annotations.NiagaraProperty;
import javax.baja.nre.annotations.NiagaraType;
import javax.baja.sys.*;

@NiagaraType
@NiagaraProperty(
  name = "foo",
  type = "baja:String",
  defaultValue = "BString.make(\"bar\")"
)
public class BMyFoo
  extends BComponent
{
}
```

```
@NiagaraProperty
@NiagaraAction
@NiagaraTopic
```

Example BTestNg Test

```java
package test.com.glward.myModule;

import com.glward.myModule.BMyFoo;
import org.testng.Assert;
import org.testng.annotations.Test;

import javax.baja.nre.annotations.NiagaraType;
import javax.baja.test.BTestNg;
import javax.baja.sys.Sys;
import javax.baja.sys.Type;

@NiagaraType
@Test(groups = "ci")
public class BMyFooTest extends BTestNg {
// test code
}
```

```java
// test 'foo'
public void testFoo()
{
  // given
  BMyFoo myFoo = new BMyFoo();

  // when
  myFoo.setFoo("foo");

  // then
  Assert.assertEquals(myFoo.getFoo(), "foo");
}
```

Slot-o-matic

Annotations

```
@NiagaraType
@NiagaraProperty(
  name = "foo",
  type = "baja:String",
  defaultValue = "BString.make(\"bar\")"
)
```

Generated Slot Code

```
//region Property "foo"
public static final Property foo =
  newProperty(0, BString.make("bar"), null);
public String getFoo() { return getString(foo); }
public void setFoo(String v) { setString(foo, v, null); }

//region Type
@Override
public Type getType() { return TYPE; }
public static final Type TYPE = Sys.loadType(BMyFoo.class);
```

Slot-o-matic

Input

```
@NiagaraType
@Test(groups = "ci")
public class BMyFooTest
  extends BTestNg
{
//region...
}
```

Generated Slot Code

```
//region /*+ ------------ BEGIN BAJA AUTO GENERATED CODE ------------ +*/
//@formatter:off
/*@ $test.com.glward.myModule.BMyFooTest(2979906276)1.0$ @*/
/* Generated Mon Mar 21 09:23:42 EDT 2022 by Slot-o-Matic (c) Tridium, Inc.
2012-2022 */

  //region Type

  @Override
  public Type getType() { return TYPE; }
  public static final Type TYPE = Sys.loadType(BMyFooTest.class);

  //endregion Type

//@formatter:on
//endregion /*+ ------------ END BAJA AUTO GENERATED CODE -------------- +*/
```

Running Slot-o-matic

```
C:\work\Niagara\myModule>gradlew :myModule-rt:slotomatic
> Task :myModule-rt:slotomatic

BUILD SUCCESSFUL in 1s
1 actionable task: 1 executed
```

…

```
C:\work\Niagara\myModule>gradlew :myModule-rt:slotomatic --include com.glward.myModule.BFoo
C:\work\Niagara\myModule>gradlew :myModule-rt:slotomatic --include com.glward.myModule.*
C:\work\Niagara\myModule>gradlew :myModule-rt:slotomatic --include com/glward/myModule/**.*
```

Building

```
C:\work\Niagara\myModule>gradlew moduleTestJar

BUILD SUCCESSFUL in 3s
9 actionable tasks: 9 executed
```

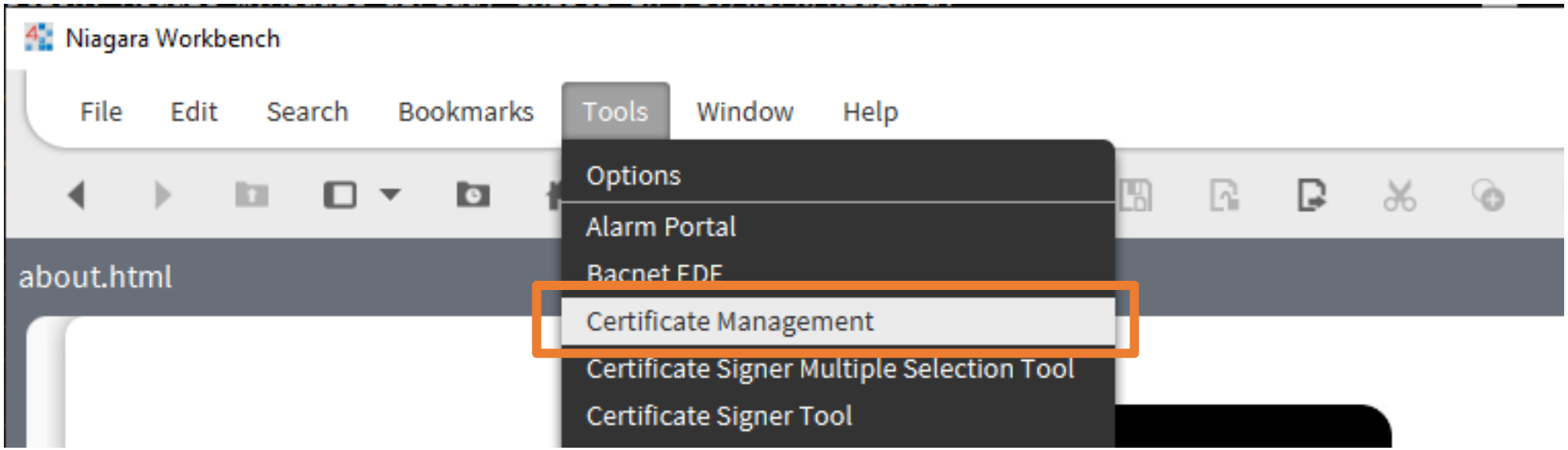
Testing

```
C:\work\Niagara\myModule>gradlew :myModule-rt:niagaraTest
> Task :myModule-rt:niagaraTest
INFO [nre] Launching Niagara Runtime Environment
...
SEVERE [11:25:01 21-Mar-22 EDT][sys.module] Could not verify module signature
javax.baja.sys.ModuleException: Module myModule-rtTest failed signature validation: Could
not validate cert chain
...

SEVERE [testrunner] invalid test target(s): myModule
javax.baja.sys.ModuleException: Module myModule-rtTest failed signature validation: Could
not validate cert chain

BUILD SUCCESSFUL in 34s
7 actionable tasks: 4 executed, 3 up-to-date
C:\work\Niagara\myModule>
```

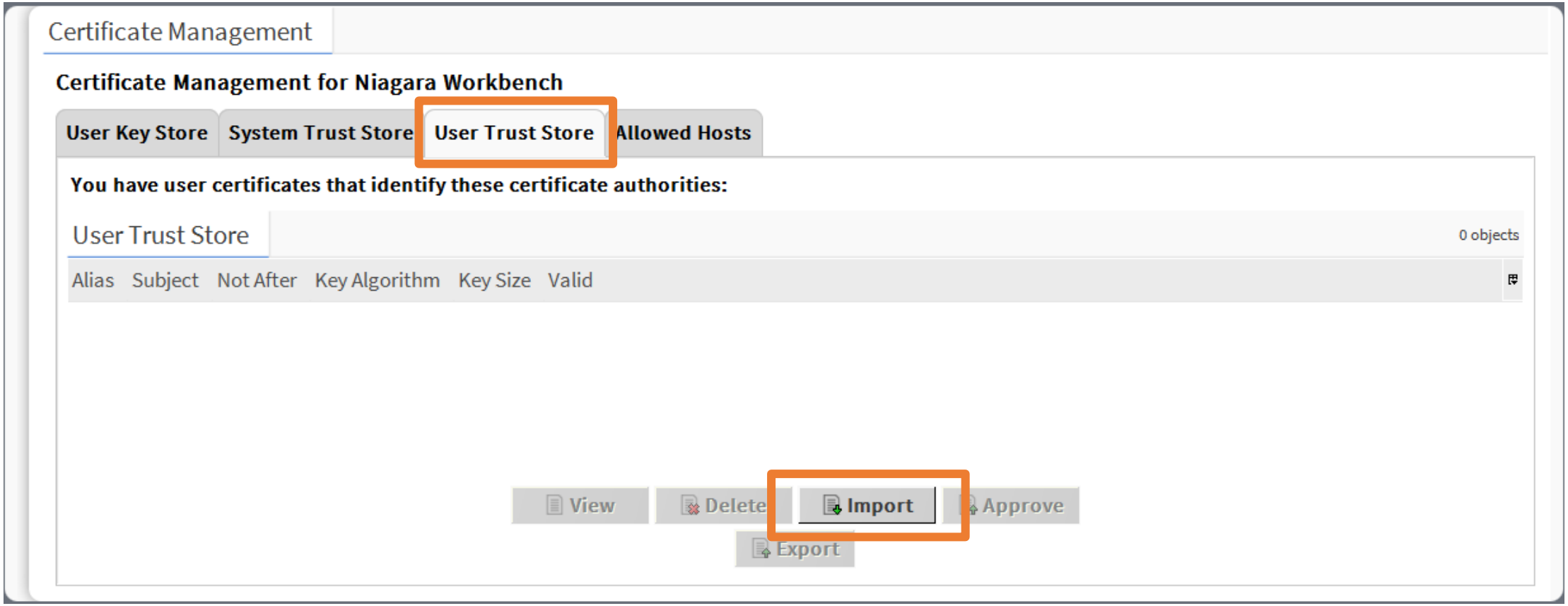
Module Signing

```
C:\work\Niagara\myModule>gradlew exportCertificate --profile-path
%USERPROFILE%\.tridium\security\niagara.signing.xml --alias Niagara4Modules --pem-
file Niagara4Modules.pem

> Task :exportCertificate


BUILD SUCCESSFUL in 1s

1 actionable task: 1 executed
```

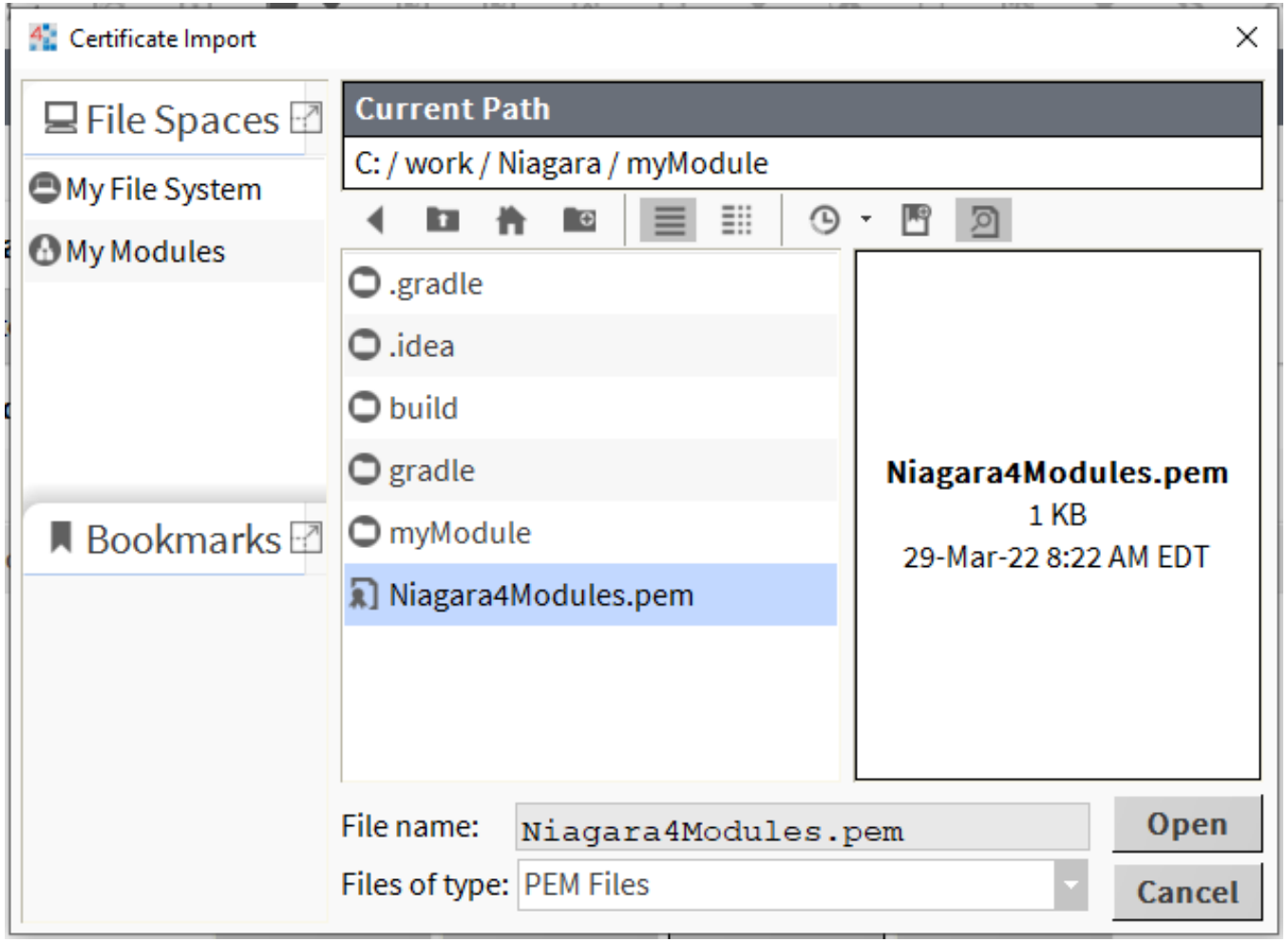
Module Signing



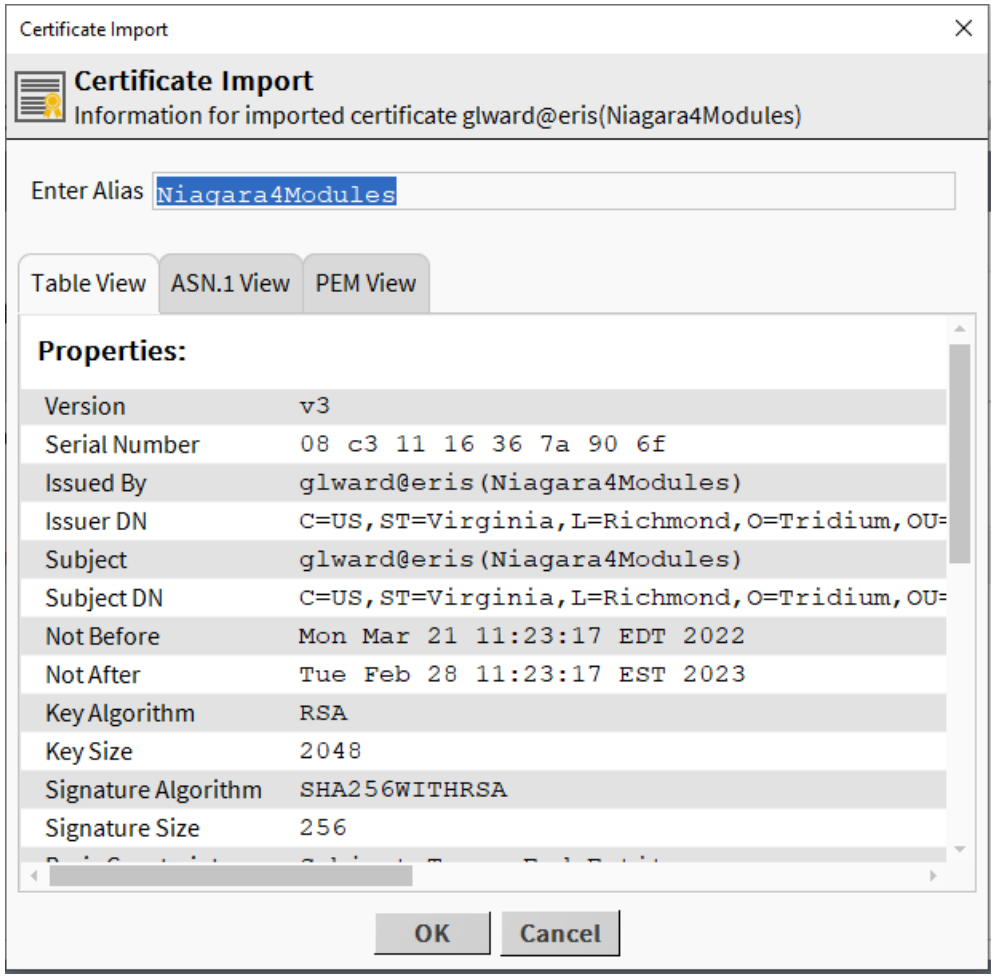
Module Signing



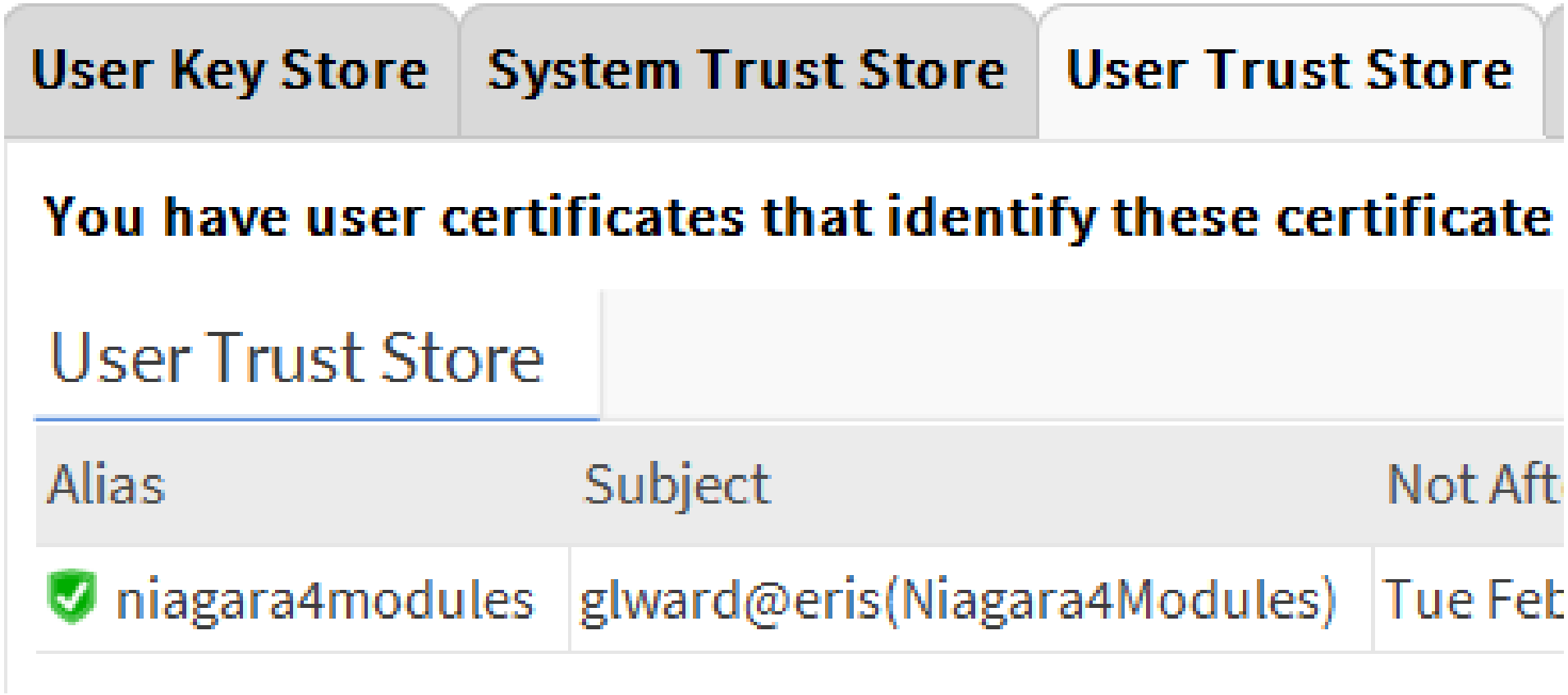
Module Signing



Module Signing



Module Signing



Testing

```
C:\work\Niagara\myModule>gradlew :myModule-rt:clean :myModule-rt:niagaraTest
➢ Task :myModule-rt:niagaraTest
➢ INFO [nre] Launching Niagara Runtime Environment
…

================================================
myModuleTest_MyFooTest
Total tests run: 1, Failures: 0, Skips: 0
================================================

================================================
Results of all run tests
Total tests run: 1, Failures: 0, Skips: 0
================================================


BUILD SUCCESSFUL in 17s
7 actionable tasks: 7 executed
```

Updates

TRIDIUM

Gradle Breaking Changes

- Configuration name changes:
  - ‘compile’ -> ‘api’
  - ‘niagaraModuleTestCompile’ -> ‘moduleTestImplementation’

Gradle Breaking Changes

- Configuration name changes:
  - ‘compile’ -> ‘api’
  - ‘niagaraModuleTestCompile’ -> ‘moduleTestImplementation’
- ‘niagaraModule’ -> ‘moduleManifest’
  - certAlias replaced with niagaraSigning extension

Gradle Breaking Changes

- Configuration name changes:
  - `'compile' -> 'api'`
  - `'niagaraModuleTestCompile' -> 'moduleTestImplementation'`
- `'niagaraModule' -> 'moduleManifest'`
  - `certAlias` replaced with `niagaraSigning` extension
- environment.gradle -> gradle.properties

Gradle Breaking Changes

- Configuration name changes:
  - ‘compile’ -> ‘api’
  - ‘niagaraModuleTestCompile’ -> ‘moduleTestImplementation’
- ‘niagaraModule’ -> ‘moduleManifest’
  - certAlias replaced with niagaraSigning extension
- environment.gradle -> gradle.properties
- vendor.gradle -> vendor extension in build.gradle.kts

Gradle Breaking Changes

- Configuration name changes:
  - ‘compile’ -> ‘api’
  - ‘niagaraModuleTestCompile’ -> ‘moduleTestImplementation’
- ‘niagaraModule’ -> ‘moduleManifest’
  - certAlias replaced with niagaraSigning extension
- environment.gradle -> gradle.properties
- vendor.gradle -> vendor extension in build.gradle.kts

Updating existing builds

Updating existing builds

- Cheat – use the New Module Wizard
  - Create shell with new Gradle scripts
  - Copy src, srcTest, supporting files for each module
  - Fix dependencies
  - Fix module resources
  - Fix signing

Updating existing builds

module-rt.gradle

```
dependencies {
  compile 'Tridium:history-wb:4.10'
  compile 'Acme:fnord:1.0'

  uberjar 'com.example:example:3.1415'

  niagaraModuleTestCompile 'Tridium:test-se:4.10'
}
```

module-rt.gradle.kts

```
dependencies {
  api("Tridium:history-wb")
  api("Acme:fnord")

  uberjar("com.example:example:3.1415")

  moduleTestImplementation("Tridium:test-se")
}
```

Updating existing builds

module-rt.gradle

```
jar {
  from('src/rc') {
    include 'foo.png'
    include '**/*.px'
    exclude 'test.jpg'
  }
}
```

module-rt.gradle.kts

```
tasks.named<Jar>("jar") {
  from("src/rc") {
    include("foo.png")
    include("**/*.px")
    exclude("test.jpg")
  }
}
```

Updating existing builds

module-rt.gradle

```
niagaraModule {
  certAlias = ["myCert"]
}
```

module-rt.gradle.kts

```
niagaraSigning {
  aliases.set(listOf("myCert"))
}
```

Build signing configuration

build.gradle.kts

```
signingServices {
  signingProfileFactory {
    allowDefaultProfile.set(false)
  }
}

niagaraSigning {
  aliases.set(listOf("myCert"))
  signingProfileFile.set(project.layout.projectDirectory.file("security/niagara.signing.xml"))
}
```

Build signing configuration

build.gradle.kts

```
signingServices {
  signingProfileFactory {
    allowDefaultProfile.set(false)
  }
}

niagaraSigning {
  aliases.set(listOf("myCert"))
  signingProfileFile.set(project.layout.projectDirectory.file("security/niagara.signing.xml"))
}
```

Build signing configuration

build.gradle.kts

```
signingServices {
  signingProfileFactory {
    allowDefaultProfile.set(false)
  }
}

niagaraSigning {
  aliases.set(listOf("myCert"))
  signingProfileFile.set(project.layout.projectDirectory.file("security/niagara.signing.xml"))
}
```

Updating existing builds

- Gradle upgrade guide
- docDeveloper Update Guide
- Developer support
- Niagara Community

Gradle Signing Profile Tools

- No more ‘keytool’ commands!

Gradle Signing Profile Tools

```
C:\work\Niagara\myModule>gradlew :createProfile --profile-path
security\niagara.signing.xml
> Task :createProfile

BUILD SUCCESSFUL in 1s
1 actionable task: 1 executed
```

Gradle Signing Profile Tools

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>Code Signing Properties</comment>
<entry key="niagara.signing.storetype">JCEKS</entry>
<entry key="niagara.signing.validity">365</entry>
<entry key="niagara.signing.refresh">183</entry>
<entry key="niagara.signing.profileType">
  com.tridium.gradle.plugins.signing.profile.RestrictedSigningProfile
</entry>
<entry key="niagara.signing.dname">
  C=US,ST=Virginia,L=Richmond,O=Tridium,OU=For Development …,CN=${user.name}@${host.name}${alias}
</entry>
<entry key="niagara.signing.storepass">…</entry>
<entry key="niagara.signing.keyalg">RSA</entry>
<entry key="niagara.signing.keysize">2048</entry>
</properties>
```

Gradle Signing Profile Tools

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>Code Signing Properties</comment>
<entry key="niagara.signing.storetype">JCEKS</entry>
<entry key="niagara.signing.validity">365</entry>
<entry key="niagara.signing.refresh">183</entry>
<entry key="niagara.signing.profileType">
  com.tridium.gradle.plugins.signing.profile.RestrictedSigningProfile
</entry>
<entry key="niagara.signing.dname">
  C=US,ST=Virginia,L=Richmond,O=Tridium,OU=For Development …,CN=${user.name}@${host.name}${alias}
</entry>
<entry key="niagara.signing.storepass">…</entry>
<entry key="niagara.signing.keyalg">RSA</entry>
<entry key="niagara.signing.keysize">2048</entry>
</properties>
```

Gradle Signing Profile Tools

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>Code Signing Properties</comment>
<entry key="niagara.signing.storetype">JCEKS</entry>
<entry key="niagara.signing.validity">365</entry>
<entry key="niagara.signing.refresh">183</entry>
<entry key="niagara.signing.profileType">
  com.tridium.gradle.plugins.signing.profile.RestrictedSigningProfile
</entry>
<entry key="niagara.signing.dname">
  C=US,ST=Any State,L=Any City,O=Acme\\, Inc,OU=Engineering,CN=${alias}
</entry>
<entry key="niagara.signing.storepass">…</entry>
<entry key="niagara.signing.keyalg">RSA</entry>
<entry key="niagara.signing.keysize">2048</entry>
</properties>
```

Gradle Signing Profile Tools

```
C:\work\Niagara\myModule>gradlew :generateCertificate --profile-path
security\niagara.signing.xml --alias NiagaraSigning --csr-file NiagaraSigning.csr
> Task :generateCertificate
Generating new private key for NiagaraSigning

BUILD SUCCESSFUL in 2s
1 actionable task: 1 executed
C:\work\Niagara\myModule>
```

Gradle Signing Profile Tools

```
C:\work\Niagara\myModule>gradlew printCertificateInfo --profile-path security\niagara.signing.xml
--alias NiagaraSigning

> Task :printCertificateInfo
cert[0]:
  subject:      CN=(NiagaraSigning), OU=Engineering, O="Acme, Inc", L=Any City, ST=Any State, C=US
  issuer:       CN=(NiagaraSigning), OU=Engineering, O="Acme, Inc", L=Any City, ST=Any State, C=US
  serialNumber: 16771420705559655382

  keyAlg:       RSA (2048 bits)
  keyUsage:     digitalSignature
  eku:          codeSigning

  notBefore:    Tue Mar 22 07:32:51 EDT 2022
  notAfter:     Tue Feb 28 07:32:51 EST 2023

  fp(sha1):     77:46:63:a2:2f:65:f7:85:2e:d1:da:29:20:e4:ad:a3:26:de:2d:b0
  fp(sha-256):
ec:fc:fa:2f:79:5a:9e:71:68:75:a8:39:c2:46:77:f2:12:cb:05:0f:41:ef:98:55:14:1f:85:55:12:5f:6a:dd

BUILD SUCCESSFUL in 1s
1 actionable task: 1 executed
C:\work\Niagara\myModule>
```

Gradle Signing Profile Tools

```
C:\work\Niagara\myModule>gradlew :importCertificate --profile-path
security\niagara.signing.xml --alias NiagaraSigning --pem-file NiagaraSigning.pem
> Task :importCertificate

BUILD SUCCESSFUL in 1s
1 actionable task: 1 executed
```

Gradle Signing Profile Tools

```
C:\work\Niagara\myModule>gradlew printCertificateInfo --profile-path security\niagara.signing.xml -
-alias NiagaraSigning

> Task :printCertificateInfo
cert[0]:
  subject:      C=US, ST=Any State, L=Any City, O="Acme, Inc", OU=Engineering, CN=(NiagaraSigning)
  issuer:       C=US, ST=Virginia, L=Richmond, O=Tridium, OU=Niagara Summit, CN=Summit CA
  serialNumber: 220762207778738914793226979 66

  keyAlg:       RSA (2048 bits)
  keyUsage:     digitalSignature
  eku:          codeSigning

  …
cert[1]:
  subject:      C=US, ST=Virginia, L=Richmond, O=Tridium, OU=Niagara Summit, CN=Summit CA
  issuer:       C=US, ST=Virginia, L=Richmond, O=Tridium, OU=Niagara Summit, CN=Summit CA
  serialNumber: 220762207768403593739041689 86

  …
```

Build signing configuration

build.gradle.kts

```
signingServices {
  signingProfileFactory {
    allowDefaultProfile.set(false)
  }
}

niagaraSigning {

  aliases.set(listOf("NiagaraSigning"))

  signingProfileFile.set(project.layout.projectDirectory.file("security/niagara.signing.xml"))
}
```

All The Small Things

All The Small Things

- Gradle parallel builds

All The Small Things

- Gradle parallel builds
- JaCoCo support for test modules

All The Small Things

- Gradle parallel builds
- JaCoCo support for test modules
- Brand dependency for modules

All The Small Things

- Gradle parallel builds
- JaCoCo support for test modules
- Brand dependency for modules
- New Module/New Driver in Gradle

All The Small Things

- Gradle parallel builds
- JaCoCo support for test modules
- Brand dependency for modules
- New Module/New Driver in Gradle
- Slot-o-matic migration improvements
  - AX-style comment block slot code deprecated for removal

Beta Access

TRIDIUM

Beta Access

- Available in Niagara 4.13

- Early Access/Beta for build tools
  - Gradle/Plugins only – not New Module Wizard

Beta Access

- Available in Niagara 4.13

- Early Access/Beta for build tools
  - Gradle/Plugins only – not New Module Wizard
  - **Works against 4.10U2, 4.11U1, 4.12**
  - Contact tridiumbetatest@tridium.com
  - Niagara Community Group

Q & A

TRIDIUM