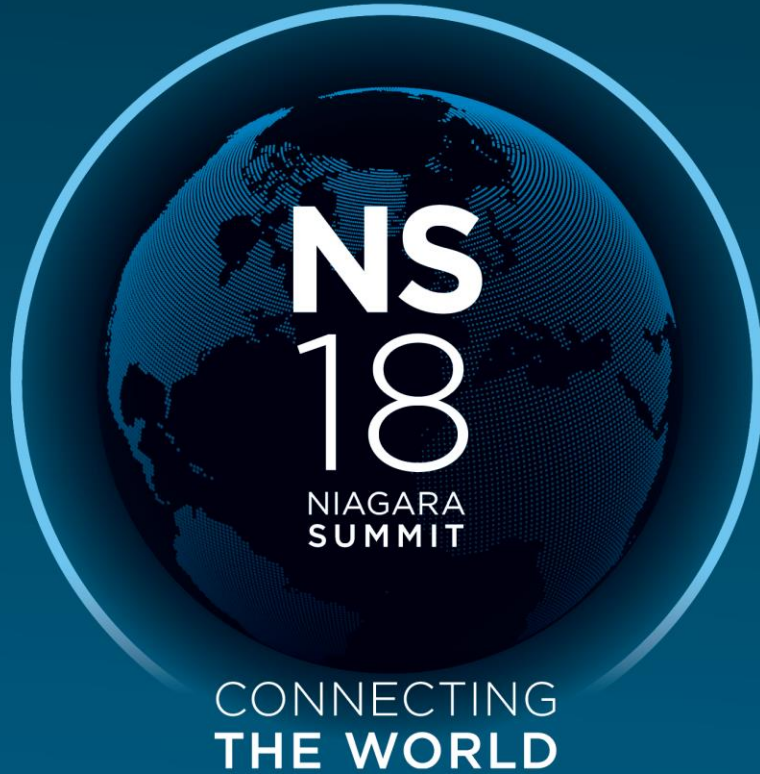




CONNECTING
THE WORLD



Building Niagara

Tim Urenda and Bill Smith
Software Engineering -- Tridium

Agenda

- New Developer Environment Features
- Slot-o-matic Improvements
- Code Signing

Development Environment

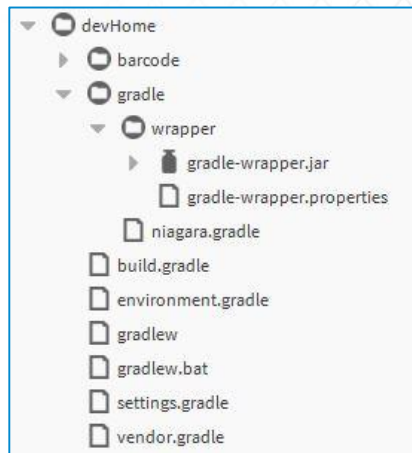
Gradle 4

- New Gradle Plugins
- Daemon enabled by default
- Requires some manual steps to upgrade existing environments
- <https://gradle.org>



IDE Integration

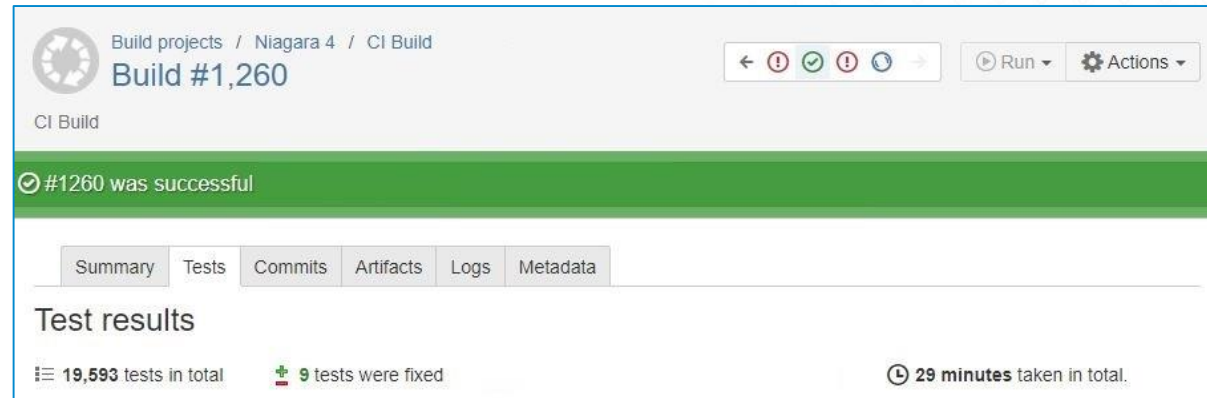
- Install a separate Java 8 JDK
- New Module Wizard copies all files needed to import into IntelliJ or Eclipse
- environment.gradle defines paths for Niagara installation



```
gradle.ext.niagara_dev_home =  
"C:/Users/Bubba/Niagara4.6/tridium/devHome"  
gradle.ext.niagara_home =  
"C:/niagara/niagara-4.6.84.1.288"  
gradle.ext.niagara_user_home =  
"C:/Users/Bubba/Niagara4.6/tridium"
```

Continuous Integration Features

- `gradlew niagaraTest`
- `gradlew gruntCi`



The screenshot displays a CI build interface for a project named "Niagara 4". The build is labeled "Build #1,260" and is identified as a "CI Build". A green banner at the top indicates that "#1260 was successful". Below this, there are tabs for "Summary", "Tests", "Commits", "Artifacts", "Logs", and "Metadata". The "Tests" tab is selected, showing "Test results". The results indicate that "19,593 tests in total" were run, with "9 tests were fixed" (represented by a green plus icon). The total time taken for the build is "29 minutes". Navigation icons (back, info, success, error, refresh) and buttons for "Run" and "Actions" are visible in the top right corner.

Demo

- New Module Wizard
- Copy files from old barcode project
- Load into IntelliJ
- Compile modules
- Run tests and display results

Slot-o-matic

Annotations

- Niagara 4 uses annotations instead of custom comments

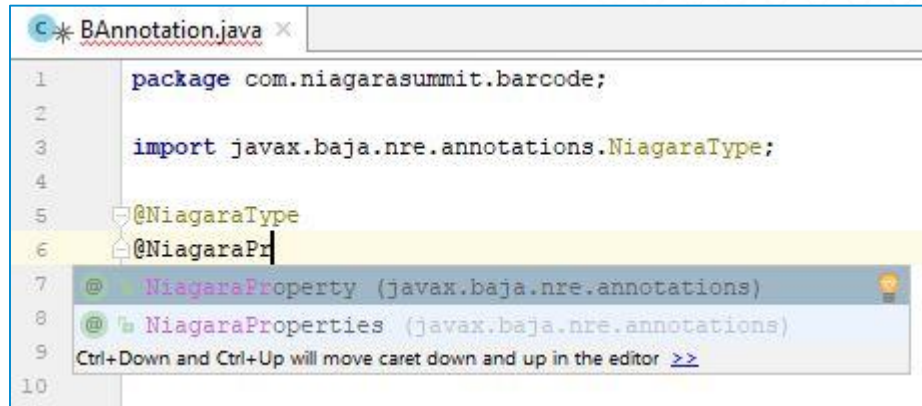
```
gradlew :module-rt:slotomatic
```

- Migration tool available:

```
gradlew :module-rt:migrateSlotomatic
```

Advantage of Annotations

- Automatic update of module(Test)-include.xml
- IDE Support



```
1 package com.niagarasummit.barcode;
2
3 import javax.baja.nre.annotations.NiagaraType;
4
5 @NiagaraType
6 @NiagaraProperty
7 @NiagaraProperty (javax.baja.nre.annotations)
8 @NiagaraProperties (javax.baja.nre.annotations)
9 Ctrl+Down and Ctrl+Up will move caret down and up in the editor >>
10
```

Slot Annotations

```
@NiagaraType(agent=@AgentOn(types={"baja:Folder"},
                                requiredPermissions="w"),
              defaultAgent=Preference.PREFERRED)
@NiagaraProperty(name = "year", type = "int",
                 defaultValue = "2018",
                 flags = Flags.READONLY)

public class BCustomView
    extends BWbView
{
    ...
}
```

module-include.xml

```
<type class="com.niagarasummit.module.ui.BCustomView"  
      name="CustomView">  
  <agent requiredPermissions="w" default="true">  
    <on type="baja:Folder" />  
  </agent>  
</type>
```

Slot Annotations - Facets

```
@NiagaraProperty(  
    name = "useFoo",  
    type = "boolean",  
    defaultValue = "true",  
    facets = {  
        @Facet(name = "BFacets.TRUE_TEXT",  
            value = "BString.make(\"%lexicon(bajau:dialog.yes)%\")"),  
        @Facet(name = "BFacets.FALSE_TEXT",  
            value = "BString.make(\"%lexicon(bajau:dialog.no)%\")")  
    }  
)
```

Non-slot Annotations

```
@NiagaraEnum(  
    range= {  
        @Range(ordinal=10, value="red"),  
        @Range(ordinal=20, value="blue")  
    },  
    defaultValue = "red")  
public final class BRedBlueEnum extends BFrozenEnum  
{...}
```

@NiagaraSingleton

@NoSlotomatic

Slot Override

```
@NiagaraType
@NiagaraProperty(name = "foo", ...)
public class BFoo extends BComponent {}

...
```

```
@NiagaraType
@NiagaraProperty(name = "foo", flags =
Flags.HIDDEN, override = true)
@NiagaraProperty(name = "fnord", ...)
public class BBar extends BFoo {}
```

```
public class BBar extends BFoo
{
    public static final Property fnord = ...
    public int getFnord();
    public void setFnord(int v);

    ...
    public static final Property foo = ...
    ...
    public static final Property bar = ...
    public int getBar();
    public void setBar(int v);
}
```

Selective Slot-o-matic

- Slot-o-matic can selectively run on only one class or package:

```
gradlew :module-rt:slotomatic -Pslotomatic.include=com.acme.*
```

- Can also exclude instead of include:

```
gradlew :module-rt:slotomatic -Pslotomatic.exclude=com.acme.BFoo
```

Demo

- Annotations
- Slotomatic task
- Automatic generation of module-include.xml

Code Signing

What Signing Isn't!



So... What is Code Signing?

- Non-repuda... what....
 - Ah! Non-repudiation!!!
- **IS** proof that code has not been modified
- **IS NOT** a guarantee that code is safe from defects
- If you want to read more:
 - <https://www.digicert.com/code-signing/best-practice-guide.htm>
 - <https://casecurity.org/wp-content/uploads/2016/12/CASC-Code-Signing.pdf>

Code Signing Example

```
5754564d205141454741554135204c59415956354741204c335948424c323920
345338343246474d20395a33324d585038203732554d4157564e204e55423536
51394b205132354c57384c33204e454e544154515320414558594e5637572056
3633565143394a20563555473338364120534b5857505839572055334a335351
5455204b34444d50485555204b534c4a4147415420454b484a5543334a203950
3753574d4b45204238563251365346204236333838444e3220345248584e3845
4b204e3239425941353820435632333255524d20504e364a5a4648422047514e
594e4437502059545a5a59365850203551513541555938205855594536465858
203558394b43345a50205441474c37365038205141483846544e5a204a4e5536
4343444720594d464a52535a4b205a4a36384356554a2045413347504c334420
4a323845524b344e20464450575848394a2046465050444e4c46204738575437
474e54204335394145564648204e564d4d3244524620553732334e5332532051
58474b4459585320474d524232565146204e4d41505846574620554a42574335
```

Hash: 3541414c5552353620434d5a5142473658204b4c444239474b4b2045564b3437

Code Signing Example

```
5754564d205141454741554135204c59415956354741204c335948424c323920
345338343246474d20395a33324d585038203732554d4157564e204e55423536
51394b205132354c57384c33204e454e544154515320414558594e5637572056
3633565143394a20563555473338364120534b5857505839572055334a335351
5455204b34444d50485555204b534c4a4147415420454b484a5543334a203950
3753574d4b4520423856325133838444e3220345248584e3845
4b204e3239425941353820433332d20504e364a5a4648422047514e
594e4437502059545a5a5936541555938205855594536465858
203558394b43345a50205441474c37365038205141483846544e5a204a4e5536
4343444720594d464a52535a4b205a4a36384356554a2045413347504c334420
4a323845524b344e20464450575848394a2046465050444e4c46204738575437
474e54204335394145564648204e564d4d3244524620553732334e5332532051
58474b4459585320474d524232565146204e4d41505846574620554a42574335
```

3332

Hash: 3541414c5552353620434d5a5142473658204b4c444239474b4b2045564b3437

Code Signing Example

5754564d205141454741554135204c59415956354741204c335948424c323920
345338343246474d20395a33324d585038203732554d4157564e204e55423536
51394b205132354c57384c33204e454e544154515320414558594e5637572056
3633565143394a20563555473338364120534b5857505839572055334a335351
5455204b34444d50485555204b534c4a4147415420454b484a5543334a203950
3753574d4b45204238563251333838444e3220345248584e3845
4b204e3239425941353820435d20504e364a5a4648422047514e
594e4437502059545a5a5936541555938205855594536465858
203558394b43345a50205441474c37365038205141483846544e5a204a4e5536
4343444720594d464a52535a4b205a4a36384356554a2045413347504c334420
4a323845524b344e20464450575848394a2046465050444e4c46204738575437
474e54204335394145564648204e564d4d3244524620553732334e5332532051
58474b4459585320474d524232565146204e4d41505846574620554a42574335

Hash: 3541414c5552353620434d5a5142473658204b4c444239474b4b2045564b3437

NOT EQUAL

Hash: 45574820393950424147334a20474344383655434a20534544393433424a205a

Code Signing Example

- Hashes are signed with a private key
- The signature of the hashes are verified with the public key
- Any changes to the hashes or private key will fail verification

Signing

5754564d20 <data> 4a42574335

+

d84394ac32 <private key> a765e8f201

=

87ae32c190 <signature> 17bd6518d1

Verification

5754564d20 <data> 4a42574335

+

87ae32c190 <signature> 17bd6518d1

+

76c78a3edd <public key> 1122ae3d6f

Why Sign Code?

It's all about trust!

Code Signing in Niagara 4

- Tridium code is already signed (since Niagara 4 release)
 - QNX OS image (JACE-8000 and Edge 10)
 - Distribution files
 - Java modules
- With N4.6 build tools, third party modules are now signed (dev signature)
- Some new permissions require trusted signing
 - Reflect
 - Ecc508 signing
- System integrity can be validated with signed code

Sample Signing Profile

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>Code Signing Properties</comment>
  <entry
    key="niagara.signing.profileType">com.tridium.signing.RestrictedSigningProfile
  </entry>
  <entry key="niagara.signing.keypass.Modules">password</entry>
  <entry key="niagara.signing.keypass.OtherModules">password1</entry>
  <entry key="niagara.signing.storepass">Password!</entry>
  <entry key="niagara.signing.standardtsa">
    http://sha256timestamp.ws.symantec.com/sha256/timestamp
  </entry>
</properties>
```

Sample Signing Profile

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>Code Signing Properties</comment>
  <entry
    key="niagara.signing.profileType">com.tridium.signing.RestrictedSigningProfile
  </entry>
  <entry key="niagara.signing.keypass.Modules">password</entry>
  <entry key="niagara.signing.keypass.OtherModules">password1</entry>
  <entry key="niagara.signing.storepass">Password!</entry>
  <entry key="niagara.signing.standardtsa">
    http://sha256timestamp.ws.symantec.com/sha256/timestamp
  </entry>
</properties>
```

Sample Signing Profile

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>Code Signing Properties</comment>
  <entry
    key="niagara.signing.profileType">com.tridium.signing.RestrictedSigningProfile
  </entry>
  <entry key="niagara.signing.keypass.Modules">password</entry>
  <entry key="niagara.signing.keypass.OtherModules">password1</entry>
  <entry key="niagara.signing.storepass">Password!</entry>
  <entry key="niagara.signing.standardtsa">
    http://sha256timestamp.ws.symantec.com/sha256/timestamp
  </entry>
</properties>
```

Sample Signing Profile

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>Code Signing Properties</comment>
  <entry
    key="niagara.signing.profileType">com.tridium.signing.RestrictedSigningProfile
  </entry>
  <entry key="niagara.signing.keypass.Modules">password</entry>
  <entry key="niagara.signing.keypass.OtherModules">password1</entry>
  <entry key="niagara.signing.storepass">Password!</entry>
  <entry key="niagara.signing.standardtsa">
    http://sha256timestamp.ws.symantec.com/sha256/timestamp
  </entry>
</properties>
```

Sample Signing Profile

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>Code Signing Properties</comment>
  <entry
    key="niagara.signing.profileType">com.tridium.signing.RestrictedSigningProfile
  </entry>
  <entry key="niagara.signing.keypass.Modules">password</entry>
  <entry key="niagara.signing.keypass.OtherModules">password1</entry>
  <entry key="niagara.signing.storepass">Password!</entry>
  <entry key="niagara.signing.standardtsa">
    http://sha256timestamp.ws.symantec.com/sha256/timestamp
  </entry>
</properties>
```

Specifying Global Alias

- In “build.gradle”:

```
...
subprojects { Project p ->
    p.pluginManager.withPlugin('com.tridium.niagara-module') {
        p.niagaraModule {
            certAlias = "Modules"
        }
    }
}
...
```

Specifying Global Alias

- In “build.gradle”:

```
...
subprojects { Project p ->
    p.pluginManager.withPlugin('com.tridium.niagara-module') {
        p.niagaraModule {
            certAlias = "Modules"
        }
    }
}
...
```

Specifying Module Alias

- In “module-rt.gradle”:

```
...
niagaraModule {
    moduleName = "module"
    preferredSymbol = "mod"
    runtimeProfile = "rt"
    certAlias = "Modules"
    ...
}
...
```

Specifying Module Alias

- In “module-rt.gradle”:

```
...
niagaraModule {
    moduleName = "module"
    preferredSymbol = "mod"
    runtimeProfile = "rt"
    certAlias = "Modules"
    ...
}
...
```

Signing Modules

- Default alias "Niagara4Modules"
- Can specify multiple aliases

```
certAlias = "Alias1, Alias2"
```

Signing Modules

- Local signing: automatic

- Specify signing profile:

```
gradlew jar -Pniagara.signing.profile=c:\workspace\profile.xml
```

- Store the keystore and signing properties for release in a secure location!

Signing Modules

Example JAR contents

- META-INF

- MANIFEST.MF
- module.xml
- NIAGARA1.RSA
- NIAGARA1.SF



Manifest-Version: 1.0
Implementation-Version: 4.6.80.1
Sealed: true
Implementation-Vendor: Tridium

Name: com/tridium/sys/engine/LeaseManager\$SummaryPage.class
SHA-256-Digest: 0HV/mQ7jxZf9wJsPM+lo+ewNxQGfJQMkM15JvXuK8r4=

Name: javax/baja/util/BTypeConfig\$IConfigurable.class
SHA-256-Digest: AlK1dKs3sL+mMiic61w0zHfUIAq4cYIFziq5NIW3aMQ=

Name: javax/baja/naming/UnresolvedException.class
SHA-256-Digest: 2JkRAtsuV/NpJGhmEu+ErELW2fHq1KUPLFTDeCcXzc8=

Verifying Signed Modules

```
$ jarsigner -verify -verbose %NIAGARA_HOME%\modules\mod-rt.jar
```

```
...
```

```
- Signed by "CN=user@hostname(Niagara4Modules), OU=For Development Purposes Only Do Not Distribute, O=Tridium, L=Richmond, ST=Virginia, C=US"
```

```
  Digest algorithm: SHA-256
```

```
  Signature algorithm: SHA256withRSA, 2048-bit key
```

```
jar verified.
```

Warning:

This jar contains entries whose certificate chain is not validated.

This jar contains signatures that does not include a timestamp. Without a timestamp, users may not be able to validate this jar after the signer certificate's expiration date (2018-12-31) or after any future revocation date

Verifying Signed Modules

```
$ jarsigner -verify -verbose %NIAGARA_HOME%\modules\baja.jar
...
- Signed by "CN=Niagara, OU=Niagara Modules, O=Niagara, L=Richmond, ST=VA, C=US"
  Digest algorithm: SHA-256
  Signature algorithm: SHA256withRSA, 2048-bit key
  Timestamped by "CN=Symantec SHA256 TimeStamping Signer - G2, OU=Symantec Trust
Network, O=Symantec Corporation, C=US" on Tue Feb 13 20:14:31 UTC 2018
  Timestamp digest algorithm: SHA-256
  Timestamp signature algorithm: SHA256withRSA, 2048-bit key
...

jar verified.
```

How does this impact you?

- Optional but recommended in 4.6
 - Signatures are verified
 - Certificates are not validated
- Signing will be mandatory in a future version of Niagara
- Integrating this now will prepare you for the future!

Questions?